Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station

Guoliang Xing; Tian Wang; Weijia Jia; Minming Li
Deptment of Computer Science, City University of Hong Kong
83 Tat Chee Ave., Hong Kong
{glxing, tianwang, wei.jia, mli000}@cityu.edu.hk

ABSTRACT

Recent research shows that significant energy saving can be achieved in wireless sensor networks with a mobile base station that collects data from sensor nodes via short-range communications. However, a major performance bottleneck of such WSNs is the significantly increased latency in data collection due to the low movement speed of mobile base stations. To address this issue, we propose a rendezvous-based data collection approach in which a subset of nodes serve as the *rendezvous points* that buffer and aggregate data originated from sources and transfer to the base station when it arrives. This approach combines the advantages of controlled mobility and in-network data caching and can achieve a desirable balance between network energy saving and data collection delay. We propose two efficient rendezvous design algorithms with provable performance bounds for mobile base stations with variable and fixed tracks, respectively. The effectiveness of our approach is validated through both theoretical analysis and extensive simulations.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Architecture and Design—*wireless communication*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Routing and layout*

General Terms

Algorithms, Performance, Theory

Keywords

Sensor Networks, Controlled Mobility, Energy Efficiency, Real-time Systems

1. INTRODUCTION

Energy is a paramount concern to wireless sensor networks (WSNs) that must operate for an extended period of time on limited power supplies such as batteries. A major portion of energy expenditure of WSNs is attributed to multi-hop wireless communications. Recent research has exploited controlled mobility as a promising approach to reduce communication energy consumption of WSNs. For instance, a mobile base station (BS) may roam about a sensing field and collect data from sensor nodes through short-range communications. The energy consumption of static nodes is thus reduced because fewer number of wireless relays are needed in the network.

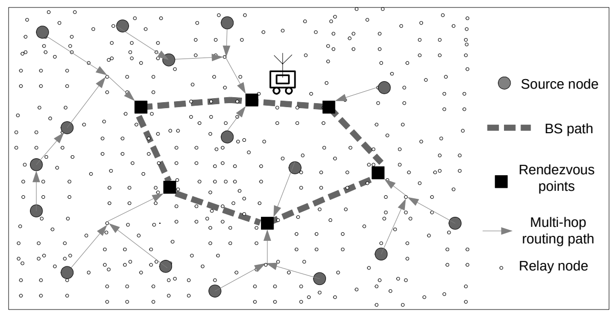

Figure 1: An example of data collection in a $500 \times 500\ m^2$ sensing field. The BS moves at $0.5\ m/s$. It takes the BS about 20 minutes to visit all rendezvous points located within $100\ m$ from the center of field. It takes more than 2 hours to visit 100 source nodes randomly distributed in the field.

The major performance bottleneck of WSNs with a mobile BS is the increased latency in data collection. The typical speed of practical mobile sensor systems (e.g., NIMs [21], Packbot [24] and Robomote [6]) is about $0.1 - 2$ m/s. As a result, it takes a mobile BS hours to tour a large sensing field, which cannot meet the delay requirements of many sensing applications. The low movement speed is a fundamental design constraint for mobile BSs because increasing the speed will lead to significantly higher manufacturing cost and power consumption. For instance, the power consumption of the Packbot node [24] is about 60 W when moving at 1 m/s, and increases quadratically with speed [4].

In this paper, we propose a rendezvous-based data collection approach that explores the controlled mobility of BS and the capability of in-network data caching. Specifically, a subset of static nodes in the network will serve as the *rendezvous points (RPs)* and aggregate data originated from sources. The BS periodically visits each RP and picks up the cached data. An example of rendezvous-based data collection is illustrated in Fig. 1. This approach has several key advantages. First, a broad range of desirable trade-offs between energy consumption and communication delay can be achieved by jointly optimizing the choices of RPs, motion path of BS and data transmission routes. Second, the use of RPs enables the BS to collect a large volume of data at a time without traveling a long distance, which mitigates the negative impact of slow speed

MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.

of BS on overall network throughput. Third, mobile nodes communicate with the rest of the network through RPs at scheduled times, which minimizes the disruption to the network topology caused by mobility.

This paper makes the following contributions. 1) We formulate the rendezvous design problem for WSNs with a mobile BS, which aims to find a set of RPs that can be visited by the BS within a required delay while the network cost incurred in transmitting data from sources to RPs is minimized. 2) We develop two efficient rendezvous design algorithms with constant approximation ratios. The first algorithm places RPs on an approximate Steiner Minimum Tree (SMT) of source nodes, which allows the data to be efficiently aggregated at RPs while shortening the data collection tour of BS. The second algorithm is designed for mobile BSs that must move along fixed tracks. Based on the analysis on the optimal structure of connection between sources and a fixed track, we can find efficient RPs within bounded BS tour on the track. 3) Simulation results show that both algorithms can achieve satisfactory performance under a range of settings. The theoretical performance bounds of the algorithms are also validated through simulations.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the basic model and assumptions of this work. The rendezvous design problems with a variable and fixed BS track are studied in Section 4 and 5, respectively. Section 6 presents the simulation results and Section 7 concludes the paper.

2. RELATED WORK

Recent work has exploited controlled mobility to enhance the connectivity of sparse ad hoc networks [8, 25], and reduce the energy consumption of WSNs. We review three different approaches [3] of utilizing controlled mobility in data collecting WSNs.

Motivated by the observation that the nodes in the vicinity of the base station deplete energy first as they forward more data, several projects [16, 7, 27] propose to use mobile base stations to achieve balanced energy usage. It is showed in [16, 7] that the optimal path of BS is the perimeter of the sensing field. However, the average network energy consumption in this approach is high as nodes must communicate with the mobile base stations through multi-hop routes. Moreover, as base stations often change their paths dynamically, additional overhead is incurred in maintaining efficient network topology. In this paper, we explore the delay-tolerant nature of many WSN applications by caching data inside the network and transferring to the BS when it arrives. Furthermore, we assume a data aggregation model in which nodes close to the BS may not consume more energy than other nodes as data traffic can be aggregated before being relayed. The results of [16, 7] are derived without accounting for data aggregation and hence are not applicable to our problem.

In the second approach, the BS visits source nodes and gather data from them via one-hop communications. Shah et al. [22] model the performance of BS based on the random mobility model. Several heuristics are proposed in [9, 24] to schedule the movement of BS such that the source nodes can be visited before buffer overflow. While this approach minimizes the network energy consumption by completely avoiding multi-hop wireless transmissions, it incurs high latency when collecting data from large sensing fields due to the slow speed of BS.

The third is a hybrid approach that jointly considers multi-hop network transmissions and the movement of BS in data collection. The rendezvous approach studied in this paper falls into this category. In [13, 12], the data are sent from other nodes to the nodes close to the path of BS. The BS then picks up the cached data when it passes by. Wang et al. [26] show that constraining the BS in the vicinity of the base station can maximize the network lifetime. These projects are not concerned with collecting data within bounded delay. In [10], urgent messages are sent to the source nodes that are visited by the BS more frequently in order to achieve early delivery. As the BS picks up most data (except the urgent messages) from data sources, such a scheme results in high latency in large networks. Moreover, different from our objective of minimizing network energy consumption in data collection within bounded delays, the urgent messages are assumed to be infrequent in [10] and hence have limited impact on network energy consumption. Xing et al. [29] proposed two algorithms for planning the data collection tours of mobile nodes. However, the mobile nodes must travel along network routing trees in [29]. In this work, we aim to jointly optimize data routing paths and the BS tour. In addition, our algorithms are based on a data aggregation model that is not considered in [29].

Our problem formulation is related to the Traveling Salesman Problem (TSP) [1]. However, new techniques are needed for our problem as the tour of BS and network routes of data should be jointly considered in order to determine the optimal locations of rendezvous points while only the tour of visiting a fixed set of sites needs to be found in TSP.

3. BASIC APPROACH AND ASSUMPTIONS

In this section, we first provide a brief overview of the problem, and then introduce the network model used in this paper.

3.1 Problem Description

In our problem, a set of source nodes generate data samples that must be delivered to the base station (BS) within time interval D. Our objective is to find a tour of the BS that visits a set of nodes referred to as *rendezvous points (RPs)*. The RPs cache the data originated from sources and send to the BS via short-range transmissions when it arrives. The total energy consumption incurred by the network to transmit the data from sources to the RPs should be minimized under the constraint that all data must be delivered to the BS before the deadline D. An important characteristic of this problem is that the BS tour and data transmission routes must be jointly designed in order to find the optimal RP locations. We refer to this problem as *rendezvous design* in data collection.

The delay bound may be imposed for two different reasons. First, applications often require data to be delivered within certain deadline. For instance, a user may issue the following sliding-window query: "sample seismic data every 10s and archive at the base station every 10 minutes", where the deadline is 10 minutes. Second, the delay bound may also be imposed due to the recharging cycle of the BS. For instance, the battery of Robomote node lasts for about 30 minutes [6] during movement. Although a mobile BS can periodically replenish its energy (e.g., by moving to a fixed docking station), frequent battery recharging should be avoided to reduce the disruptions to normal operation of the network.

3.2 Network Model

According to several empirical studies [5], the speed that data packets are relayed in a WSN is about several hundred meters per second, which is much higher than the speed that a mobile device moves. Therefore, the data collection deadline can be mapped to the maximum allowable length of the BS tour that visits all RPs. We denote the maximum length of the BS tour, $L = v_m D$, where D is a data collection deadline and v_m is the average movement speed of BS.

We assume that data from different sources can be aggregated at a node before being relayed. Data aggregation [18] has been widely adopted by data collection applications to reduce network traffic. Specifically, we assume the *N-to-one* aggregation model in which a node can aggregate multiple data packets it received into one packet before relaying it. Such a model is applicable to a number of scenarios such as collecting the maximum or average value of samples from different sensors.

We assume that nodes are densely deployed in a region and all nodes use the same transmission power. Accordingly, the total energy consumed by transmitting a data packet along a multi-hop path is proportional to the Euclidean distance between sender and receiver. This assumption is justified by the fact that the Euclidean distance between two nodes in a dense wireless network is approximately proportional to the hop count between the same nodes [20]. We note that such an energy model is also adopted by several existing power-efficient data communication protocols in WSNs [14]. This assumption also allows the BS to estimate the network energy consumption without knowing the global network topology.

We assume that the storage capacity of a node is large enough to buffer the total volume of data generated by the sources within delivery deadline D. Several recent sensor network platforms [19] can integrate $10 \sim 100$ Mb NAND flash memory with ultra-low power consumption. Finally, nodes and the BS are assumed to know their own physical locations through the GPS units on them or a location service in the network.

3.3 Overview of the Approach

We investigate two rendezvous design problems in this paper. In the first problem, the BS may freely move within the network deployment region. In the second problem, the motion of the BS is constrained on a fixed track. Although such limited mobility reduces the contacts with fixed nodes in a network, it significantly simplifies the motion control of BS and improves the system reliability. For instance, several mobile sensor systems (e.g., XYZ [17] and NIMs [2, 21]) are designed to move along fixed cables.

For each rendezvous design problem, we develop an approximation algorithm that is executed by the BS to find a data collection tour, a set of RPs on the tour, and a set of routing trees that are rooted at the RPs and connect all sources. As we assume that the BS does not have the global information about the network except the locations of sources, the RPs found are physical locations at which there may not exist real nodes. This issue can be addressed in the following two ways. First, the BS may find a real node near each RP through the network. For instance, it may send an area anycast [11] message addressed to the physical location of an RP. The message will be delivered to a node in the vicinity of the intended location, which may serve as the RP. Alternatively, the BS may travel along the calculated tour and recruit nodes to serve as RPs.

4. RENDEZVOUS DESIGN WITH A VARIABLE BS TRACK

In this section, we study the rendezvous design problem when the BS can freely move in the network deployment region along any track. Our objective is to find a BS tour no longer than L and a set of routing trees that are rooted on the tour and connect all sources, such that the total Euclidean length of the trees is minimized. The problem is formally defined as follows.

DEFINITION 1. *(Rendezvous Design with a Variable BS Track) Given a set of sources S, find 1) a tour U no longer than L and 2) a set of geometric trees $\{T_i(V_i, E_i)\}$ that are rooted on U and $S \subseteq \cup_i V_i$; such that $\sum_i \sum_{(u,v)\in E_i} |uv|$ is minimized, where (u, v) is an edge of tree T_i and $|uv|$ is its Euclidean length.*

An example of the solution is illustrated in Fig. 3(b). The BS tour visits three RPs: RP_1, RP_2 (which is also a source node) and RP_3. Two trees[1] are rooted at RP_1 and RP_3 and connect all sources. The objective is to minimize the total edge length of the two trees.

This problem can be shown to be NP-hard by a reduction from the Euclidean Traveling Salesman Problem (TSP). Specifically, a special case of the decision version of the problem is to ask if there exists a set of RPs such that the network energy consumption is zero. In order to incur zero network energy consumption, all the sources must be RPs as well. In other words, the BS must visit all the RPs on a tour no longer than L. This is exactly the decision version of the GTSP problem in which a salesman needs to visit a set of sites on a tour no longer than a given bound.

4.1 A SMT-based Approximation Algorithm

In order to find the optimal RP locations, the BS tour and the data routing paths need to be jointly designed. When the BS is fixed, the optimal routing tree under the N to one aggregation model is the Steiner Minimum Tree (SMT). For a given set of nodes V on a plane, finding the shortest tour that visits all the nodes is a TSP problem. Interestingly, the SMT of the nodes in V is a lower bound of the optimal TSP tour because the SMT connects all nodes using the shortest length of edges and does not contain any cycle. This fact suggests that positioning RPs on the SMT of source nodes may lead to short BS tour while maintaining good data aggregation performance. Motivated by this observation, we develop an SMT-based approximation algorithm referred to as *Rendezvous Design for Variable Tracks (RD-VT)*. The basic idea is to find a subtree of an approximate SMT of sources such that all the RPs on the subtree can be visited by a BS tour no longer than L while the total edge length of the rest of the SMT is minimized. The pseudo code of the algorithm is shown in Fig. 2.

/*S is the set of source node locations, L is the maximum BS tour length, σ is a positive constant smaller than L*/

Input: $S = \{s_i\}$, L, σ
Output: RP list R

1. Find an approximate Steiner minimum tree T that connects all points in S. Randomly choose a source B as the root of the tree.
2. Y=L/2;
3. Traverse T from B in the preorder until the total length of edges visited is Y. Denote the subtree traversed as T'.
4. $R = \{r_i | \, r_i$ is the intersection between T' and path $s_i \rightarrow B$ on $T\}$.
5. **if** $X = L - TSP(R) > \sigma$
6. Y=Y+X/2; **goto** 3;
7. **else exit**;

Figure 2: The pseudo code of the RD-VT algorithm.

The algorithm first constructs an approximate SMT, T, which is rooted at source B and connects all other sources. Then T is traversed in the preorder until a length of $L/2$ edges is covered. The preorder (also referred to as depth-first) traversal of tree T is the recursive process of visiting all the nodes on T, starting from the root, and then traversing in the preorder each of the subtrees of the root. Denote the subtree traversed as T'. In the example shown in Fig. 3(b), T' is composed of the highlighted edges. An RP may lie on an edge of T because each edge approximates a multi-hop

[1]Edge (RP_2,S_2) can be viewed as a single-edge tree.

network path. The white circles in Fig. 3(b) represent the RPs on T'. Each RP is the intersection between T' and a source-to-root path on T. For instance, RP_3 is the intersection between T' and the path from s_2 to BS.

After the set of RPs, R, is found on subtree T', the algorithm finds a tour that visits all RPs by executing a TSP solver. Suppose $TSP(R)$ is the length of the TSP tour visiting all the RPs. If $X = L - TSP(R) > \sigma$ where σ is a small constant in the input, T' is then expanded to include a length of $X/2$ more edges of T. This process repeats until the difference between the length of the TSP tour and L is smaller than σ. σ can be set according to the desirable trade-off between solution quality and time complexity. The rationale for the iterative process of RD-VT is that each iteration explores more edges of the approximate SMT while ensuring that the current RPs can always be visited by a BS tour no longer than L.

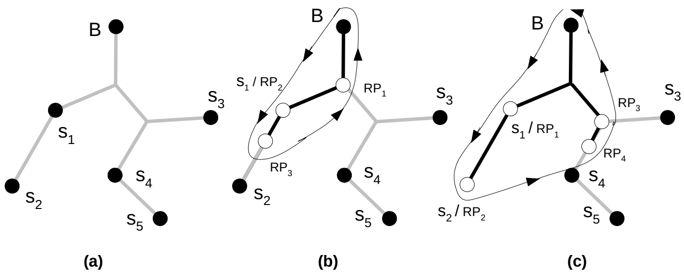

Figure 3: An example of the RD-VT algorithm's execution. Source nodes and RPs are denoted by white and black circles, respectively. s_i/RP_j represents a source node that serves as a RP as well. (a) The initial approximate SMT. (b) L/2 length of tree edges are traversed in preorder (c) The final TSP tour is no longer than L.

4.2 Performance Analysis

In this section, we first show the correctness of RD-VT and then derive its approximation ratio. We define the following notation. For a given graph G, $c(G)$ represents the total edge length of G. The correctness of RD-VT can be proved by showing that all the found RPs can be visited by a BS tour no longer than L. We have the following theorem.

THEOREM 1. *Suppose R_i represents the set of RPs found at step 4 in iteration i of RD-VT. Then there always exists a BS tour no longer than L that visits all the RPs in R_i. That is, $L - TSP(R_i) \geq 0$ always holds before the algorithm exits.*

PROOF. Suppose T_i is the subtree traversed at step 3 in iteration i of the algorithm. We first show that there exists a tour which visits all vertices of T_i and has a length no greater than $2c(T_i)$. Walking along the edges of T_i in preorder from the root and returning to the root on the same route passes each edge exactly twice. Hence, the total length of such a tour is $2c(T_i)$. We refer to such a tour as the *preorder walk* of T_i hereafter. For instance, in Fig. 3(b), the preorder walk of the highlighted subtree is $B \rightarrow RP_1 \rightarrow s_1 \rightarrow RP_3 \rightarrow s_1 \rightarrow RP_1 \rightarrow B$.

According to the definition of RPs (step 4 in Fig. 3), R_i is a subset of the vertices of T_i. If the tour found by the TSP solver is longer than $2c(T_i)$, we replace it with the preorder walk of T_i. We now prove the theorem by induction. In the first iteration, $TSP(R_1) \leq 2c(T_1) = L$. Suppose $TSP(R_i) \leq L$ holds in iteration i. According to step 6 in iteration $i+1$ of the algorithm, we have:

$$c(T_{i+1}) - c(T_i) = \frac{L - TSP(R_i)}{2} \tag{1}$$

As tree T is always traversed in preorder (step 3), the edges in $T_{i+1} \setminus T_i$ are connected. The length of the preorder walk of these edges is equal to $2(c(T_{i+1}) - c(T_i))$. A tour that visits all RPs in R_{i+1} can be constructed by appending this preorder walk to the TSP tour in iteration i. We have:

$$\begin{aligned} TSP(R_{i+1}) &\leq TSP(R_i) + 2(c(T_{i+1}) - c(T_i)) \\ &= TSP(R_i) + 2 \cdot \frac{L - TSP(R_i)}{2} \\ &= L \end{aligned}$$

□

We now analyze the performance of RD-VT. We define the following notation. SMT_S denotes the SMT connecting all sources $S = \{s_i\}$. Let β be the ratio of L to the total edge length of SMT_S, i.e., $\beta = \frac{L}{c(SMT_S)}$. We assume $\beta \ll 1$. When this condition does not hold, the BS tour is close to or directly includes many source nodes. As a result, data aggregation does not play a significant role in cost saving because the opportunity for a node to relay the data originated from more than one source is low. Let α be the best known approximation ratio for the SMT problem. We have the following theorem regarding the performance of the RD-VT algorithm.

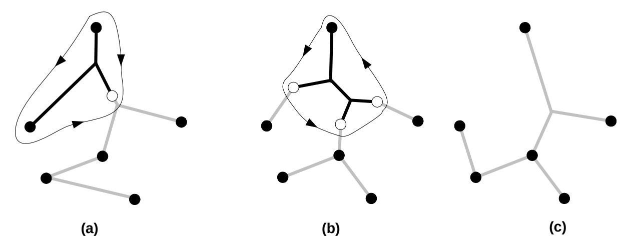

Figure 4: (a) The approximate SMT used as input and the BS tour found by RD-VT. (b) The optimal RPs and BS tour; the black tree inside the BS tour is SMT_{Q^*}, the SMT connecting the BS and RPs; A^* includes the gray trees rooted at RPs. (c) SMT_S - the SMT connecting all sources.

THEOREM 2. *The approximation ratio of the RD-VT algorithm is no greater than $\alpha + \frac{\beta(2\alpha-1)}{2(1-\beta)}$.*

PROOF. We first derive a lower bound of the optimal solution. Suppose Q^* represents the set of the BS and the RPs in the optimal solution. A^* represents the set of the trees rooted at the optimal RPs and connect all sources. A^* includes all the edges in gray and the end nodes on them. The cost of optimal solution, i.e., total distance that data travels before being picked up at the RPs, is therefore $c(A^*)$. SMT_{Q^*} and SMT_S represent the SMT with the terminal nodes as RPs in Q^* and the sources in S, respectively. Fig. 3 illustrated the graph structures defined above. As the union of SMT_{Q^*} and A^* is a Steiner tree with terminal nodes S, the total edge length of them must be no smaller than that of SMT_S:

$$c(A^*) + c(SMT_{Q^*}) \geq c(SMT_S) \tag{2}$$

Suppose T^* represents a tree connecting the BS and all the optimal RPs, which is created by removing an arbitrary line segment between two RPs on the optimal BS tour. $c(T^*)$ is smaller than the length of the BS tour and hence also smaller than L. Moreover, T^* is a Steiner tree with terminal nodes Q^*. Therefore,

$$c(SMT_{Q^*}) \leq c(T^*) < L \tag{3}$$

From (2) and (3), we have:

$$c(A^*) \geq c(SMT_S) - c(SMT_{Q^*}) > c(SMT_S) - L \tag{4}$$

Suppose T is an α-approximation SMT connecting the BS and the source nodes, and T' is the subtree of T covered by the BS tour found by the RD-VT algorithm. We have $c(T) \leq \alpha SMT_S$. The cost of the solution found by the RD-VT algorithm, i.e., the total distance that data travels on T before being picked up by the BS is equal to $c(T) - c(T')$. According to Theorem 1, at least a length of $L/2$ tree edges are covered by the BS tour[2]. That is, $c(T') \geq L/2$. We have:

$$\begin{array}{rcl} c(A^*) & \geq & c(SMT_S) - L \\ & \geq & c(T)/\alpha - L \\ & = & \dfrac{c(T) - L/2}{\alpha} - \dfrac{(2\alpha - 1)L}{2\alpha} \\ & \geq & \dfrac{c(T) - c(T')}{\alpha} - \dfrac{(2\alpha - 1)L}{2\alpha} \end{array} \tag{5}$$

Since $c(A^*) \geq c(SMT_S) - L$ and $L = \beta \cdot c(SMT_S)$, we have $L \leq \frac{\beta}{1-\beta} c(A^*)$. Combining with (5), we have:

$$\begin{array}{rcl} c(T) - c(T') & \leq & \alpha \left(c(A^*) + \dfrac{2\alpha - 1}{2\alpha} L \right) \\ & = & \alpha \left(c(A^*) + \dfrac{2\alpha - 1}{2\alpha} \cdot \dfrac{\beta}{1 - \beta} c(A^*) \right) \\ & \leq & c(A^*) \left(\alpha + \dfrac{\beta(2\alpha - 1)}{2(1 - \beta)} \right) \end{array}$$

□

We now analyze the complexity of RD-VT. According to Theorem 1, the length of BS tour increases in each iteration as the number of RPs to be visited grows. Therefore, the total number of iterations of RD-VT before termination depends on L and how fast the length of the TSP tour increases. In iteration $i + 1$, if the number of RPs remain unchanged, the increase of the TSP tour is likely greater than $c(T_{i+1}) - c(T_i)$. This is because removing an arbitrary line segment between two consecutive RPs on the TSP tour results in a tree connecting all RPs. Hence, $c(T_{i+1}) - c(T_i)$ is likely smaller than the partial TSP tour connecting the nodes in $T_{i+1} \setminus T_i$. According to (1), $c(T_{i+1}) - c(T_i) = \frac{L - TSP(R_i)}{2}$. Therefore, the length that the current TSP tour can be expanded (initially L) is reduced by at least half in each iteration. Therefore, the number of iterations of RD-VT is approximately $\mathcal{O}(\log L)$. In each iteration, a TSP tour is computed for all RPs. As each source is on one aggregation tree, the number of RPs is no more than $|S|$. There exist many efficient TSP algorithms, which have different solution quality and complexity trade-offs. The algorithm in [1] has an approximation ratio of $(1 + 1/b)$ and a complexity of $O(|S|(\log |S|)^{O(b)})$ for any fixed $b > 1$. We note that RD-VT is only run by the BS, which has more computational power than sensor nodes.

[2] $L/2$ is a tight lower bound on the length of the edges of T'. When the first edge of T visited in preorder is longer than $L/2$, T' only includes the partial edge of length $L/2$.

5. RENDEZVOUS DESIGN WITH A FIXED BS TRACK

In this section, we study the rendezvous design problem when the BS moves on a fixed track. Although a fixed track reduces the contacts between the BS and the fixed nodes in the network, it significantly simplifies the motion control of the BS and is hence adopted by several mobile sensor systems in practice. For example, in the NIMS system deployed at the James Reserve, data collecting sensors can only move along fixed cables between trees [2, 21]. Moreover, a fixed track improves the system reliability. For instance, the BS can be recharged any time during the movement.

We assume that the track consists of non-intersecting contiguous line segments, which is consistent to several practical mobile sensor systems [2, 21]. Specifically, a track is specified by $P = \{\overline{p_i p_{i+1}} \mid 1 \leq i \leq n-1\}$ where $\overline{p_i p_{i+1}}$ does not intersect with $\overline{p_j p_{j+1}}$ if $|i - j| > 1$.

Our objective is to find a continuous tour no longer than L along the track and a set of routing trees rooted on the tour that connect all sources, such that the total Euclidean length of the trees is minimized. The problem can be formally defined as follows.

DEFINITION 2. *(Rendezvous Design with a Fixed BS Track) Given a set of sources S and a track specified by $P = \{\overline{p_i p_{i+1}} \mid 1 \leq i \leq n-1\}$, find 1) a tour U on P that is no longer than L and 2) a set of geometric trees $\{T_i(V_i, E_i)\}$ that are rooted on U and $S \subseteq \cup_i V_i$; such that $\sum_i \sum_{(u,v) \in E_i} |uv|$ is minimized, where (u, v) is an edge of tree T_i and $|uv|$ is its Euclidean length.*

5.1 An Approximation Algorithm

As this problem can be easily shown to be NP-Hard, we focus on the design of approximation algorithms. When the BS moves along the track, a tour of length L corresponds to a partial track of length $L/2$. If the track is shorter than $L/2$, the problem is trivial because the BS can simply move back and forth along the whole track. Therefore, we focus on the case where the track is longer than $L/2$.

When the BS tour is given, the problem becomes minimizing the total length of edges that connect sources to the tour. Fig. 7 (a) shows an example in which sources connect to the BS tour at RPs $\{r_i\}$ on the track. It can be seen that the optimal solution is composed of a set of Steiner Minimum Trees (SMTs) that connect sources to $\{r_i\}$. However, our problem cannot be mapped to the SMT problem because $\{r_i\}$ are not known. On the other hand, it is known that MST is a good approximation of SMT in the Euclidean space. For a given BS tour ρ, denote MST_ρ as the *optimal* set of MSTs that connect sources to ρ. That is, MST_ρ has the minimum total edge length among all sets of MSTs connecting sources to ρ. In this section, we develop an approximation algorithm based on the analysis on the properties of MST_ρ.

By definition of MST, any edge of MST_ρ lies either between two sources or between a source and a point on ρ. MST_ρ has the following important property. If an edge lies between a source s_i and a point r_i on ρ, r_i is closest to s_i among all points on ρ. This is because, if there was another point r_i' on ρ closer to s_i than r_i, replacing edge (s_i, r_i) with (s_i, r_i') leads to a lighter tree, which contradicts the fact that (s_i, r_i) is an edge of MST. Note that r_i is either a start/end/turning point of ρ or the projection of a source on ρ. Fig. 7 (c) illustrates an example of MST_ρ. The sources connect to ρ at r_1' (projection of s_2 on ρ), r_2' (projection of s_2 on ρ) and r_3 (end point of ρ).

Based on the above discussion, MST_ρ can be found by extending the Kruskal MST algorithm [15] as follows. First, for each source s_i, find the closest point on ρ, r_i. Then, sort all the edges of the form (s_i, s_j) and (s_i, r_j) in the increasing order of their lengths. Edges are then tested one by one for insertion. If an edge does not generate a cycle, it will be added, otherwise, it is discarded. An important distinction with the original Kruskal algorithm is that all the edges (r_i, r_j) should be inserted before other edges as they lie on the BS tour and hence do not incur any cost. The above procedure requires that the BS tour ρ is known. However, the position of ρ on the track plays an important role in the quality of the found MSTs. Several heuristics can be used to find good BS tours. First, the closest points of sources on the track can be considered as possible start points of BS tours. Second, when sources are sparsely distributed, a set of equally spaced points can be added to the track as possible start points of BS tours.

The pseudo code of our algorithm, referred to as *Rendezvous Design for Fixed Tracks (RD-FT)*, is shown in Fig. 5. At step 1, a set of points are added on the track such that the segment between any two adjacent points (except the last segment) has a length of ΔL. At step 2, the sources' closest points on the track are identified. Then all the added points are tested in sequence as start points of BS tours at step 3. Denote X_k as the subset of the added points that lie on tour ρ_k. For each tour ρ_k, find MST_{ρ_k} using the extended Kruskal algorithm described above. At step 4, the final solution is the tour ρ that minimizes the length of MST_ρ among all tours tested.

/*S is the set of source node locations, L is the maximum BS tour length, ΔL is a positive constant, P is a fixed track composed of n line segments.*/

Input: $S = \{s_i\}$, L, ΔL, $P = \{\overline{p_i p_{i+1}} \mid 1 \le i \le n\}$
Output: RP list R

1. Find a set of points $W = \{w_i \mid 1 \le i \le m, w_1 = p_1, w_m = p_n\}$ on P such that $\forall i \ge 1\ |w_i w_{i+1}| = \Delta L$ and $|w_{m-1} w_m| = |P| - \lfloor |P|/\Delta L \rfloor \cdot \Delta L$ where $|P|$ is the total length of the track.
2. Find a set of points on the track that are closest to sources: $U = \{u_i \mid (u_i\ is\ on\ P) \wedge (|u_i s_i|\ is\ minimum)\}$. Renumber the points in $W \cup U$ along the track as $X = \{x_1, x_2, \cdots, x_l\}$.
3. **for** $1 \le k \le l-1$
   (a) Starting from x_k, find the partial track ρ_k which either has a total length of $L/2$ or includes x_l. Denote X_k the subset of points in X that lie on ρ_k.
   (b) Find a set of MSTs, MST_{ρ_k}, which connect sources to the points in X_k and has the minimum total length.
4. $MST_\rho = argmin\ c(MST_{\rho_k})$ is the set of MSTs with the minimum total length among all found MST sets. The BS tour is ρ and the RP list R is composed of the endpoints of edges of MST_ρ that lie on ρ.

Figure 5: The pseudo code of the RD-FT algorithm.

5.2 Performance Analysis

We now analyze the performance of the RD-FT algorithm. As discussed above, for a given BS tour ρ_k, the sources connect to ρ_k at their closest points on ρ_k. We prove in the following that at most three sources connect directly to the start/end point of ρ_k on edges not perpendicular to the track.

LEMMA 1. *Suppose MST_{ρ_k} is a set of MSTs found at step 3.b of the RD-FT algorithm. MST_{ρ_k} contains at most three edges connected to the start/end point of tour ρ_k that are not perpendicular to the track.*

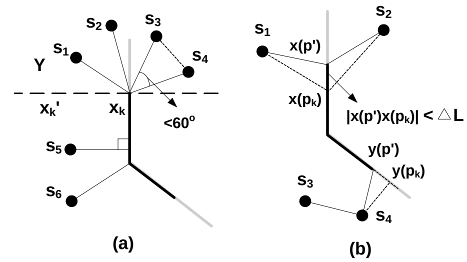

Figure 6: The track and BS tour are shown as grey and black lines, respectively. The sources are shown as solid circles. (a) $s_1 - s_4$ connect to x_k, the start point of tour ρ_k and the edges $(s_i, x_i) (1 \le i \le 4)$ are not perpendicular to the track. Angle $s_3 x_i s_4 < 60^o$ and replacing the longer of (s_3, x_k) and (s_4, x_k) with (s_3, s_4) reduces the total length of edges. (b) Tour ρ' contains the track between $x(\rho')$ and $y(\rho')$ and tour ρ_k contains the track between $x(\rho_k)$ and $y(\rho_k)$. $(s_1, x(\rho'))$ and $(s_2, x(\rho'))$ are edges of $MST_{\rho'}$. $(s_1, x(\rho_k))$ and $(s_2, x(\rho_k))$ are edges of MST_{ρ_k}. As $|x(\rho_k)x(\rho')| < \Delta L$, $|s_1 x(\rho_k)| + |s_2 x(\rho_k)| < |s_1 x(\rho')| + |s_2 x(\rho')| + 2\Delta L$. Edge $(s_4, y(\rho_k))$ connects to $y(\rho_k)$ and is shorter than edge $(s_4, y(\rho'))$ connected to $y(\rho')$.

PROOF. Without loss of generality, we only prove the lemma holds for the start point of ρ_k, x_k. Suppose $x_k x_k'$ is perpendicular to the track (shown in Fig. 6 (a)). Suppose there were more than three sources $s_1, s_2, s_3, s_4, \ldots$ connecting to x_k on edges not perpendicular to the track. It is easy to see that all these sources lie on a half plane (denoted by Y) bounded by $x_k x_k'$ because the sources on the other side of $x_k x_k'$ can find closer points on ρ_k than x_k. As a result, the half plane Y is divided by at least four radials connecting x_k and s_1, s_2, s_3, s_4. Hence at least one angle of form $\angle s_i x_k s_j$ $(1 \le i, j \le 4)$ is less than 60^o. Therefore, replacing the longer of the two edges (s_i, x_k) and (s_j, x_k) with (s_i, s_j) will reduce the total edge length, which contradicts the fact that MST_{ρ_k} has the minimum total edge length among all spanning trees that connect sources to ρ_k. $\square$

At step 1 of the RD-FT algorithm, a set of points are added on the track such that the segment between any two adjacent points (except the last segment) has a length of ΔL. At step 3, these points together with the closest points of all sources on the track are considered as possible start points of BS tours. This consideration ensures that the found BS tour has a good quality compared to the optimal tour.

LEMMA 2. *Suppose BS tour ρ' has the minimum $MST_{\rho'}$ among all tours no longer than L, and ρ is the tour found by RD-FT. $c(MST_\rho) - c(MST_{\rho'}) \le 3\Delta L$ must hold.*

PROOF. Denote $x(\rho)$ and $y(\rho)$ as the start and end points of BS tour ρ. We consider the following two cases. First, if there exists an edge between a source and $x(\rho')$ and it is perpendicular to the track, RD-FT considers $x(\rho')$ as the start point of a BS tour at step 2 and thus ρ' is examined at step 3.b. According to step 4 of RD-FT, ρ has the minimum total edge length among all the BS tours examined. Therefore, $c(MST_\rho) = c(MST_{\rho'})$ and the lemma holds.

We now discuss the case in which no edge between a source and $x(\rho')$ is perpendicular to the track. Suppose the sources in set S' directly connect to $x(\rho')$ on $MST_{\rho'}$ on edges E'. According to Lemma 1, there exist at most three such edges. As illustrated in Fig. 6 (b), E' includes $(s_1, x(\rho'))$ and $(s_2, x(\rho'))$. Suppose $x(\rho')$ is between $x(\rho_{k-1})$ and $x(\rho_k)$ that are the start points of two adjacent tours found by RD-FT at step 3, and $x(\rho_k)$ is closer to the end point of track. According to step 1 of RD-FT, $x(\rho_k)$ is at most ΔL away from $x(\rho')$ on the track. We can create a set of spanning trees $PMST_{\rho_k}$ (that connects sources to ρ_k) by connecting sources in S' directly to $x(\rho_k)$ instead of $x(\rho')$. The remaining structure is the same as $MST_{\rho'}$ because of two facts: 1) $MST(\rho')$ does not include any edge with an end point between $x(\rho')$ and $x(\rho_k)$ on the track (because if such an edge exists, it must be perpendicular to the track and hence be considered by RD-FT as a start point of a BS tour at step 3, which contradicts the fact that ρ_k is the BS tour next to ρ'). 2) $y(\rho')$ is no closer to the end point of the track than $y(\rho_k)$. In other words, these two facts imply that ρ_k gives the sources in $S \setminus S'$ the same number of or more choices on where to connect to the track. As illustrated in Fig. 6 (b), sources s_1 and s_2 (originally connected to $x(\rho')$) connect to $x(\rho_k)$ on ρ_k and their edges get at most ΔL longer.

As S' contains at most three sources, the total length of the edges in $PMST_{\rho_k}$ is at most $3\Delta L$ greater than that of the corresponding edges on $MST(\rho')$. Notice that we have $c(PMST_{\rho_k}) \geq c(MST_{\rho_k})$, which then implies that $c(MST_\rho) \leq c(MST_{\rho'}) + 3\Delta L$ holds.

$\square$

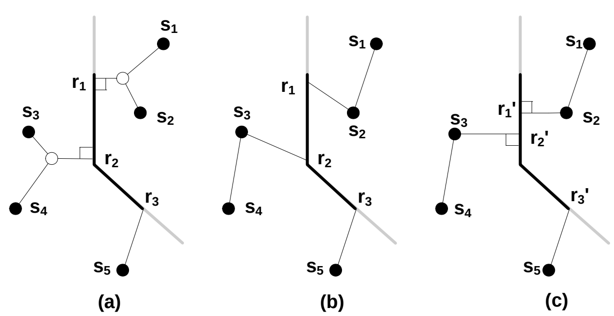

Figure 7: The track and BS tour are shown as grey and black lines, respectively. The sources are shown as solid circles. (a) SMT_{ρ^*}, the optimal solution. The sources are connected to tour ρ^* via a set of SMTs. The RPs are r_1, r_2 and r_3. (b) $MST(S \cup \{r_i\})$, the MSTs that connect sources to r_1, r_2 and r_3. (c) MST_{ρ^*}, the optimal set of MSTs that connect sources to ρ^*. The RPs are the points on ρ^* closest to the sources.

We have the following theorem regarding the approximation ratio of RD-FT.

THEOREM 3. *Denote MST_ρ and SMT_{ρ^*} as the solution found by RD-FT and the optimal solution, respectively. Define $\gamma = \frac{\Delta L}{c(MST_{\rho^*})}$. Then the approximation ratio of RD-FT is $(1+3\gamma)\cdot\frac{2}{\sqrt{3}}$. That is, $c(MST_\rho) \leq (1+3\gamma)\cdot\frac{2}{\sqrt{3}}\cdot c(SMT_{\rho^*})$.*

PROOF. Suppose sources in S connect to the track at $R = \{r_1, r_2, \ldots\}$ in SMT_{ρ^*}, which is illustrated in Fig. 7 (a). Construct a set of MSTs that connect sources in S to points in R, as illustrated in Fig. 7 (b). Denote the set of MSTs found as $MST(S \cup R)$. It is known that the MST connecting a set of points has an approximation ratio of $\frac{2}{\sqrt{3}}$ with respect to the SMT connecting the same set of points. Therefore, we have:

$$c(MST(S \cup R)) \leq \frac{2}{\sqrt{3}} \cdot c(SMT_{\rho^*}) \qquad (6)$$

Moreover, the optimal set of MSTs (denoted as MST_{ρ^*}) that connect sources to ρ^* can be found using the extended Kruskal algorithm described above. MST_{ρ^*} is illustrated in Fig. 7 (c). As MST_{ρ^*} has the minimum total length of edges among all sets of MSTs connecting sources to ρ^*, $c(MST_{\rho^*}) \leq c(MST(S \cup R))$. Suppose BS tour ρ' has the minimum $MST_{\rho'}$ among all tours no longer than L. We have $c(MST_{\rho'}) \leq c(MST_{\rho^*})$. Therefore,

$$\begin{aligned} c(MST_\rho) &\leq c(MST_{\rho'}) + 3\Delta L \qquad (Lemma\ 2) \\ &\leq c(MST_{\rho^*}) + 3\Delta L \\ &\leq (1+3\gamma)\cdot c(MST_{\rho^*}) \\ &\leq (1+3\gamma)\cdot c(MST(S \cup R)) \\ &\leq (1+3\gamma)\frac{2}{\sqrt{3}}\cdot c(SMT_{\rho^*}) \quad (Eqn.\ (6)) \end{aligned}$$

$\square$

Theorem 3 shows that the performance bound of RD-FT is dependent on $\gamma = \frac{\Delta L}{c(MST_{\rho^*})}$. As the optimal BS tour ρ^* is not known, ΔL can be set to be γ times of the cost of a lower bound of MST_{ρ^*}. For instance, the set of MSTs that connect sources to the whole track, e.g., without considering the length constraint of the BS tour, is a lower bound of MST_{ρ^*}. The extended Kruskal MST algorithm has a complexity of $|S| \log |S|$ where S is the set of sources. Hence the complexity of RD-FT is $O(\frac{|P|}{\Delta L} \cdot |S| \log |S|)$ where $|P|$ is the length of the track. We can see that a smaller ΔL will lead to a better performance bound at the price of higher overhead. Therefore, ΔL can be tuned to achieve the desirable trade-off between the solution quality and running time.

6. PERFORMANCE EVALUATION

This section evaluates the performance of proposed rendezvous design algorithms. The simulations are based on a geometric network model in which any physical point in the network region can be chosen as an RP. The performance metric is the total Euclidean length of routing trees that connect sources to the RPs. Such a geometric network model allows us to validate the design of RD-VT and RD-FT and the tightness of derived performance bounds.

In all simulations, sources are randomly distributed in a $300m \times 300m$ region. The simulation code is written in C++.

6.1 The Performance of RD-VT

In the implementation of RD-VT, the GeoSteiner software package [28] is used to compute the SMT of sources. Note that GeoSteiner is based on an exact SMT algorithm. All the exact SMTs of our problem instances were found within 5 seconds on a Pentium IV PC with a 2.4 GHz CPU. A TSP solver based on local search heuristics [23] is used at step 4 of RD-VT (see Fig. 2). We implemented the following baseline algorithms for performance comparison. *RD-SMT* is a simplified version of RD-VT in which the BS tour is a preorder walk on the SMT of sources. *RD-MST* computes an MST of source nodes first, then the BS walks for a distance of L/2 on the tree in the preorder. *NN* is a nearest-neighbor based heuristic in which the BS always travels to the source closest to the current source that has been visited. The sources that are not visited by the BS connect to the closest source on the BS tour. At the beginning of each simulation, the BS starts moving from the position of source node closest to the top left corner of region.

Fig. 8 shows the total length of routing trees found by different algorithms. Total 30 source nodes are distributed in the region. RD-VT significantly outperforms other baseline algorithms under all

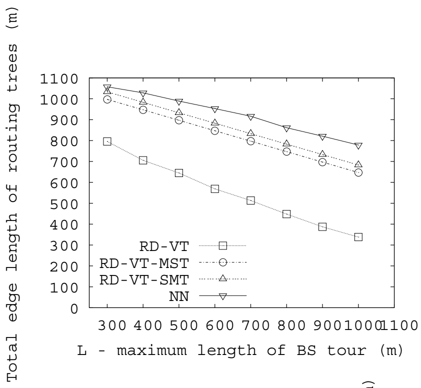

Figure 8: Total length of routing trees vs. L

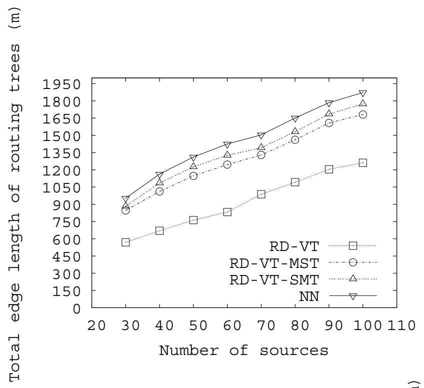

Figure 9: Total length of routing trees vs. the number of sources

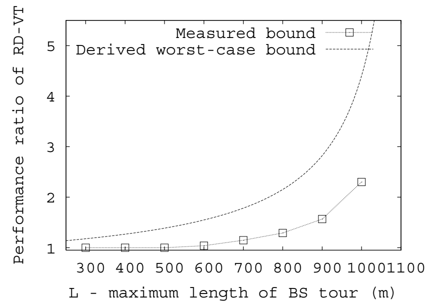

Figure 10: Performance ratio of RD-FT vs. L

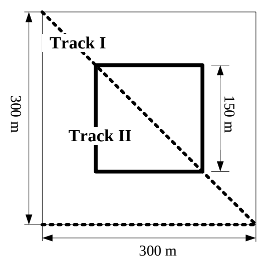

Figure 11: Two tracks used for performance evaluation.

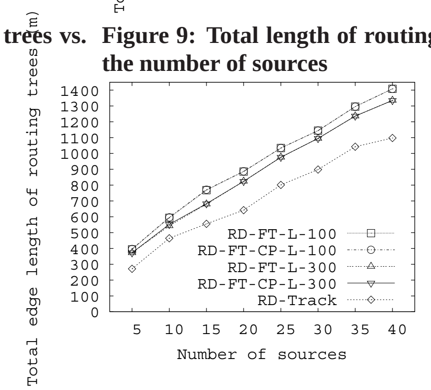

Figure 12: Performance of RD-FT with BS track I

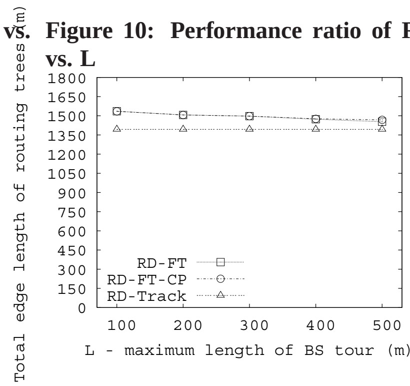

Figure 13: Performance of RD-FT with BS track II

settings. Moreover, the gap between RD-VT and other algorithms increases with L, which shows that RD-VT can effectively reduce network routing cost by taking advantage of a longer delay bound on data collection. Fig. 8 also shows that RD-VT-MST and RD-VT-SMT perform similarly as MST is a good approximation of SMT in the Euclidean space. Fig. 9 shows the performance with different number of sources. We can see that the total edge length of routing trees found by all algorithms increases with the number of sources. Consistent with the results in Fig. 8, RD-VT is superior to all other algorithms. NN performs the worst as it does not jointly consider the locations of all sources when computing the tour of BS.

Fig. 10 shows the measured and derived performance ratios of RD-VT compared with the optimal solution. The measured performance ratios are computed as the ratios of the results of RD-VT to $c(SMT_S) - L$ where SMT_S is the exact SMT of sources. According to (4), the value of $c(SMT_S) - L$ is a lower bound of the optimal solution. The derived ratio is plotted according to the equation given by Theorem 2. We can see that the derived worst-case ratio approaches the measured ratio when L is small. That is, the derived performance ratio is tighter for a smaller L. This result is due to the fact that the theoretical bound is derived under condition $L \ll c(SMT_S)$. We note that this condition holds in many practical scenarios. In this set of simulations, there are 30 sources in total and $c(SMT)$ is 1147 m. When L is 500 m and the average speed of BS [3] is $0.1 - 2$ m/s, the data collection delay is 4 to 80 minutes, which satisfy the requirements of many WSN applications.

6.2 The Performance of RD-FT

We evaluate the performance of RD-FT using two tracks shown in Fig. 11. We varied the value of γ in RD-FT from 0.01 to 0.1 and observed no obvious difference of performance. γ is set to be 0.01 in the following simulations. The following baseline algorithms are used for comparison. RD-FT-CP is a simplified version of RD-FT in which only the sources' closest points on the track are considered as possible start points of BS tours. RD-Track finds the optimal set of MSTs connecting sources to the track without the constraint of L.

Fig. 12 shows the performance of RD-FT with a triangle-shape BS track. Each algorithm is labeled by the name and the value of L. Interestingly, RD-FT and RD-FT-CP perform similarly under all settings. Therefore, it suffices to only consider the sources' closest points on the track as possible start points of BS tours, which significantly reduces the complexity of RD-FT. Moreover, RD-FT yields a good performance compared with RD-Track that does not consider the constraint of L. For instance, the performance of RD-FT falls within $28\% \sim 45\%$ of that of RD-Track when L is 100 m, which is only about 14% of the length of whole track.

Fig. 13 shows the performance of RD-FT with a square-shape BS track when the number of sources is 50. We can see that RD-FT and RD-FT-CP yield a satisfactory performance compared with RD-Track. Moreover, Fig. 12 and 13 show that the square-shape track gives a smaller gap between RD-FT/RD-FT-CP and RD-Track than the triangle-shape track. However, the triangle-shape yields shorter routing trees as it crosses the whole region and hence enables the BS to communicate with more nodes directly. These results provide insights into the design of fixed BS tracks.

7. CONCLUSION

In this paper, we study the rendezvous-based data collection in WSNs with a mobile base staion. We develop two efficient rendezvous design algorithms with constant approximation ratios. The first algorithm is based on SMT and allows the data to be efficiently aggregated at RPs while shortening the data collection tour of BS. The second algorithm is designed for mobile BSs that must move along fixed tracks. Based on the analysis on the optimal structure

[3]The speed is chosen according to several robotic units [21, 24, 6] used in mobile sensor systems.

of connection between sources and a fixed track, we can find efficient RPs within bounded BS tour on the track. Simulation results show that both algorithms can achieve satisfactory performance under a range of settings. The theoretical performance bounds of the algorithms are also validated through simulations.

8. ACKNOWLEDGEMENT

The work described in this paper was partially supported by the Research Grants Council of Hong Kong under grants RGC 9041266 and 9041129 and City University of Hong Kong under a grant ARD 9668009.

9. REFERENCES

[1] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5), 1998.

[2] M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin. Call and response: experiments in sampling the environment. In *SenSys*, 2004.

[3] E. E. Y. G. D. Bozdag. Mobility-based communication in wireless sensor networks. *IEEE Communications Magazine*, 44(7), 2006.

[4] D. J. Chang and E. K. Morlok. Vehicle speed profiles to minimize work and fuel consumption. *Journal of Transportation Engineering*, 131(3), 2005.

[5] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher. Real-time power-aware routing in sensor networks. In *IWQoS*, 2006.

[6] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme. Robomote: enabling mobility in sensor networks. In *IPSN*, 2005.

[7] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Globecom*, 2003.

[8] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang. Towards mobility as a network control primitive. In *MobiHoc*, 2004.

[9] Y. Gu, D. Bozdag;, R. W. Brewer, and E. Ekici. Data harvesting with mobile elements in wireless sensor networks. *Comput. Networks*, 50(17), 2006.

[10] Y. Gu, D. Bozdag, and E. Ekici. Mobile element based differentiated message delivery in wireless sensor networks. In *WoWMoM*, 2006.

[11] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. A spatiotemporal communication protocol for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(10), 2005.

[12] D. Jea, A. A. Somasundara, and M. B. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *DCOSS*, 2005.

[13] A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 5(8), 2006.

[14] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon. Dynamic delay-constrained minimum-energy dissemination in wireless sensor networks. *Trans. on Embedded Computing Sys.*, 4(3), 2005.

[15] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society.*, 7:48-50, 1956.

[16] J. Luo and J.-P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *INFOCOM*, 2005.

[17] D. Lymberopoulos and A. Savvides. Xyz: a motion-enabled, power aware sensor node platform for distributed sensor network applications. In *IPSN*, 2005.

[18] S. Madden, M. J. Franklin, , J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *OSDI*, 2002.

[19] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy. Ultra-low power data storage for sensor networks. In *IPSN*, 2006.

[20] D. Niculescu and B. Nath. Dv based positioning in ad hoc networks. *Journal of Telecommunication Systems*, 22(1-4), 2003.

[21] R. Pon, M. A. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. J. Kaiser, M. Srivastava, G. Sukhatme, and D. Estrin. Networked infomechanical systems: a mobile embedded networked sensor platform. In *IPSN*, 2005.

[22] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE SNPA Workshop*, 2003.

[23] M. Sofreavia. A local search based tsp solver. http://www.cs.sunysb.edu/ãlgorith/implement/tsp/distrib/maugis/README.

[24] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava. Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4), 2007.

[25] M. M. B. Tariq, M. Ammar, and E. Zegura. Message ferry route design for sparse ad hoc networks with mobile nodes. In *MobiHoc*, 2006.

[26] W. Wang, V. Srinivasan, and K.-C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom*, 2005.

[27] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *HICSS*, 2005.

[28] D. M. Warme, P. Winter, and M. Zachariasen. Exact algorithms for plane steiner tree problems: A computational study. *In D.Z. Du, J.M. Smith and J.H. Rubinstein (Eds.) Advances in Steiner Trees*, 2000. http://www.diku.dk/geosteiner/.

[29] G. Xing, T. Wang, Z. Xie, and W. Jia. Rendezvous planning in mobility-assisted wireless sensor networks. In *The 28th IEEE Real-Time Systems Symposium (RTSS)*, 2007.