Prolonging Network Lifetime Through the Use of Mobile Base Station in Wireless Sensor Networks

Oday Jerew
Department of Engineering
College of Engineering and Computer Science
The Australian National University
Canberra, ACT, 0200, Australia
oday.jerew@anu.edu.au

Weifa Liang
Department of Computer Science
College of Engineering and Computer Science
The Australian National University
Canberra, ACT, 0200, Australia
wliang@cs.anu.edu.au

ABSTRACT

Prolonging network lifetime is one of the most important design objectives in energy-constrained wireless sensor networks (WSNs). Using a mobile instead of a static base station (BS) to reduce or alleviate the non-uniform energy consumption among sensor nodes is an efficient mechanism to prolong the network lifetime. In this paper, we deal with the problem of prolonging network lifetime in data gathering by employing a mobile BS. To achieve that, we devise a novel clustering-based heuristic algorithm for finding a trajectory of the mobile BS that strikes the trade-off between the traffic load among sensor nodes and the tour time constraint of the mobile BS. We also conduct experiments by simulations to evaluate the performance of the proposed algorithm. The experimental results show that the use of clustering in conjunction with a mobile BS for data gathering can prolong network lifetime significantly.

Keywords

Wireless sensor networks, Mobile base station, Network lifetime, Clustering.

1. INTRODUCTION

A sensor network consists of a large number of small devices that have sensing, processing, and transmitting capabilities, that are powered by small batteries. Therefore energy efficiency in the design of routing protocols for wireless sensor networks (WSNs) is of paramount importance. Among different types of energy consumptions, a major portion of energy expenditure is contributed to wireless communication. To reduce the communication energy consumption, controlled mobility has been shown to be one promising approach [1][2][7]. For instance, a mobile base station (BS) can roam a sensing field and gather data from sensor nodes through short-range communications. The energy consumption of each sensor node is reduced, since fewer relays are needed for the sensor node to relay its message to

the BS [14]. In contrast, the increased latency on data gathering by employing mobile BSs represents a major performance bottleneck in WSNs, because it takes the mobile BS a while to tour a large sensing field, which may not meet the stringent delay requirement imposed in some mission-critical real-time applications. The slow speed of mobile BS thus is a fundamental design constraint, the faster the moving speed, the higher the manufacturing cost of the mobile BS, and the more energy the sensor nodes consume [14][10].

In a flat routing topology, sensor nodes near to the BS consume much more energy than others, since they relay packets for others. Due to the limitation imposed by the flat routing topology structure, the hierarchical organization of sensor nodes in the design of routing protocols is introduced [12][15], in which sensor nodes are organized into clusters and cluster heads relay aggregated results of sensing data within clusters via the other cluster heads to the BS. The essential operation in sensor clustering is to select a set of cluster heads from all sensor nodes, and to cluster the remaining sensor nodes within the cluster heads. Each cluster head is responsible for coordination of its sensor nodes. The sensor nodes within a cluster transmit their sensing data to the cluster head through multi-hop relays. A cluster head finally performs data gathering on the received data and forwards the aggregate result to the BS. Data gathering is one of most frequent and fundamental operations in sensor networks; the efficiency of implementing this operation in some degree determines the network lifetime. In a static BS environment, to perform data gathering using a routing tree rooted at the static BS, each sensor node can aggregate the received data from its children in order to transmit the same volume of data, regardless of how much data it has received from its child nodes. Though this scheme reduces the energy consumption of sensor nodes, one disadvantage of this scheme is that it generally takes more time to gather data, since the sensor nodes cannot transmit their results prior to receiving all data from their children, which in turn incurs a longer latency on data delivery.

In this paper, we consider data gathering under a mobile BS environment subject to a specified tour delay time constraint of the mobile BS, by adopting a clustering-based approach. To reduce the energy consumption of a cluster head to forward sensing data, the mobile BS roams the sensing field and visits only the cluster heads to gather sensing data. Therefore, the distribution of the cluster heads in the entire network affects the load balance among the sensor nodes and hence the network lifetime.

The main contribution of this paper is the use of a mobile BS for network lifetime improvement in data gathering. It proposes a schema of clustering sensor nodes into clusters. The heuristic algorithm for finding a trajectory of mobile BS consisting of cluster heads meets the following criteria: (i) the energy consumption among the sensor nodes within any cluster is balanced; and (ii) the total traversal time of the mobile BS on the trajectory is bounded by a given value. It also has been shown that the use of clustering in conjunction with the control mobility of the BS increases the network lifetime significantly.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the system model and problem definition. The process of cluster formation, and finding on cluster heads and the trajectory of the mobile BS are presented in Section 4. Section 5 and 6 present the simulation results and concludes the paper, respectively.

2. RELATED WORK

Several studies have been conducted to balance the energy consumption among sensor nodes by using mobile BSs [1, 2, 7, 9]. Zhao et al. [16] examined the problem of efficient data delivery in sparse networks by optimizing both the path and speed of a message ferry (a special mobile BS). They used a controllably mobile BS to improve the performance of data delivery, thereby reducing energy consumption; then they extended their work using multiple message ferries to minimize the data delivery delay by finding feasible ferry routes [17]. This work is further extended to find the ferry route where the sensor nodes are mobile rather than static [13]. These studies all assumed that there are special mobile BSs for facilitating the connectivity of sensor nodes.

Several other studies combine the mobile BS into a multi-hop forwarding approach. Kansal et al. [6] performed an experimental evaluation for a small size of sensor network, assuming that a mobile BS moves back and forth on a straight line (a fixed path). They employed a directed diffusion approach to gathering sensed data from the sensor nodes beyond the communication range of the mobile BS. This approach is further extended to the case where multiple mobile BSs move on a line, and an algorithm for load balancing is proposed in [5], assuming that the mobile sensor nodes fully cover the entire area of the network. Luo and Hubaux [7] proposed an analytical model to find a trajectory of the mobile BS. They showed that the optimal tour of the mobile BS is the perimeter of the sensing field, the average energy consumption by this approach however is quite high, as sensor nodes must communicate with the mobile BS through multi-hop relays. Moreover, additional overhead is incurred to maintain the routing topology since the BS changes its route dynamically.

Ma and Yang [8] and Sugihara and Gupta [11] proposed heuristics for finding routing paths for mobile BSs. In [8] it has been assumed that the moving path of mobile BS consists of a series of line segments. Sensor nodes close to each line segment are organized into clusters. A specified configuration is applied, where the mobile BS starts data gathering from the left side of the path, moves towards the right side, and comes back to the left side again. In [11] it has been assumed that the BS can select the path and change its speed under a predefined acceleration constraint to achieve the minimum data delivery latency and minimize

the energy consumption of sensor nodes, and that each sensor node transmits its data directly to the mobile BS only when the mobile BS is within the communication range of the sensor node. However, our research assumes that sensor nodes send their sensing data to their cluster heads and the cluster heads forward the sensing data to the mobile BS when the mobile BS visits the cluster head.

Xing et al. [14] considered the data gathering problem under the mobile BS environment by proposing a path selection algorithm for the mobile BS. They proposed a rendezvous-based data gathering approach, in which a subset of nodes are chosen as rendezvous points. The role of these points is to buffer and aggregate data originating from sensor nodes. When the mobile BS arrives within the transmission ranges of rendezvous points the data will be forwarded to the mobile BS, and the tour of the BS is selected such that all the sensing data to be gathered within a specified tour delay.

3. PRELIMINARIES

3.1 Network Model

We make the following assumptions about the network:

1. The transmission range of each sensor node r is fixed.

2. All sensor nodes have identical initial energy, and the mobile BS replenishes its energy periodically; thus, there is no energy concern with the mobile BS.

3. The speed of relaying a data packet is much faster than the moving speed of a mobile BS. Thus, the total delays in data gathering can be mapped into the maximum length of a BS tour. Assume that the length L of the mobile BS tour is fixed, and the average speed V_m of the mobile BS is constant. The maximum delay time of the mobile BS from starting data collection to return to the deposit point, is $D = L/V_m$. In addition, we assume that the relay time of a data packet originating from the farthest sensor node to the center of the sensing field area is less than the time $2r/V_m$ - the duration of the BS moving within the transmission range of a sensor node.

4. Sensor nodes are densely deployed in the monitored region. Accordingly, the total energy consumed by transmitting a data packet along a multi-hop path is proportional to the length of the path.

5. The storage of a sensor node is limited, so that it cannot buffer a large volume of data.

6. Sensor nodes and the mobile BS are assumed to be able to know their own physical locations through GPS or a location service in the network.

3.2 Problem Definition

This paper consider a data gathering application, such as environmental monitoring, in which all the sensing data must be delivered to the BS within a specified delay time. Our optimization objective is to prolong the network lifetime by minimizing the energy consumption of sensor nodes using a mobile BS. Given a network with a mobile BS, assume that the length L of BS tour is given, as is its speed V_m. The problem is to find a tour for the mobile BS such that the network lifetime is maximized.

We propose that sensor nodes be organized into clusters such that all the cluster heads can be visited by the mobile BS, where the length of BS tour is no longer than L. The location of the cluster head in its cluster is an essential factor to balance the energy consumption of the cluster sensor nodes. In addition, cluster head location affects the length of BS tour. The challenge of this problem is to find the optimal locations of cluster heads by jointly considering the BS tour and the network lifetime.

3.3 Clustering Based on Equal Area Partition

In order to achieve load balance among the cluster heads, it is required to balance the number of sensor nodes among the clusters, since each cluster head has to forward the data packets within it to the mobile BS. Suppose that the sensing field A is partitioned into k subareas $A_1, A_2, ..., A_k$ for a given $k \geq 1$. Assume that the number of sensor nodes n is randomly distributed in A. The sensor nodes in each A_i form a cluster, $1 \leq i \leq k$. Let $p_1, p_2, ..., p_k$ be the probabilities of sensor nodes located at $A_1, A_2, ..., A_k$ respectively. The multinomial probability distribution can be used to represent the number of sensor nodes in each area as follows:

$$P(n_1, n_2, ..., n_k) = \frac{n!}{n_1! n_2! ... n_k!} p_1^{n_1} p_2^{n_2} ... p_k^{n_k}, \qquad (1)$$

where $n = \sum_{i=1}^{k} n_i$ is the number of sensor nodes in the network.

If $N_1, N_2, ..., N_k$ have a multinomial distribution with parameters n and $p_1, p_2, ..., p_k$, then the expected number of sensor nodes within each cluster is $E(N_i) = np_i$. To obtain an equal number of sensor nodes in each cluster, we have $p_1 = p_2 = ... = p_k$, which cannot be achieved unless $A_1 = A_2 = ... = A_k$, assuming that the density function of sensor nodes in the sensing area is uniformly distributed.

4. ALGORITHM FOR FINDING THE ROUTE OF MOBILE BS

4.1 Algorithm Overview

To determine the best possible locations for cluster heads in order to maximize the network lifetime, two issues must be considered. The first issue is how to cluster the sensor nodes in the entire network such that (i) all the cluster heads can be visited by a mobile BS and (ii) the length of the BS tour is no greater than the given tour length L. The second issue is the selection of cluster heads to balance the energy consumption among sensor nodes within each cluster. The cluster heads are the bottleneck of energy consumption, since they have to forward the sensing data of sensor nodes within it to the mobile BS. Thus, to maximize the network lifetime, the energy consumption of cluster heads needs to be balanced, which can be achieved by partitioning the entire sensor field into equal subareas. To organize the sensor nodes into clusters, each sensor node is assigned to the subarea in which it is located. Thus, the energy consumption among the cluster heads will be balanced since the sensor nodes are uniformly deployed in the sensing field. Next, it is required to find the cluster heads by jointly considering energy consumption of sensor nodes in the entire cluster and the length of BS tour L. To balance the energy consumption among sensor nodes, it is important to select the cluster head such that any sensor node in a cluster is at most a certain
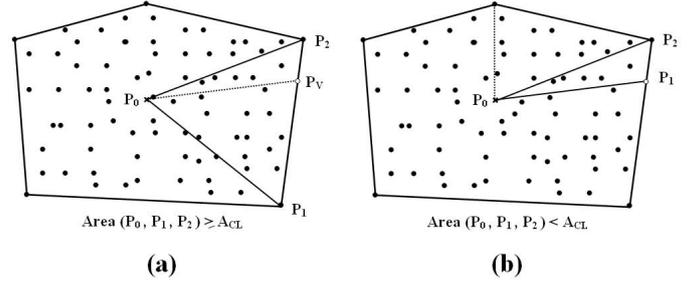
Figure 1: An example of clustering procedure, P_0 is the centroid location of sensing field, P_1 and P_2 are the locations of two boundary sensor nodes, $P_V \in \overline{P_1 P_2}$, and A_{CL} is the cluster area.

number of hops away from its cluster head. Accordingly, the sensor nodes near to the cluster center become the candidates for the cluster head, if the length of the BS tour is no greater than L. In subsections 4.2- 4.4 we propose detailed algorithm for the problem.

4.2 Clustering

The idea of clustering is to divide the sensing field area into equal subareas by radial lines from the center of the field area; therefore the sensor nodes on the boundary of the sensing field area need to be determined. Graham's scanning algorithm is applied to find a set of the boundary sensor nodes B for the convex polygon P of the sensing field. In this polygon, each sensor node is either on the boundary or inside of the polygon. The area of P can be calculated using the locations of boundary sensor nodes (X_i, Y_i), $1 \leq i \leq p_n$, where p_n is the number of boundary sensor nodes and (X_i, Y_i) is the location of a boundary sensor node. Assume the location of sensor node (X_{p_n+1}, Y_{p_n+1}) is (X_1, Y_1). The area A_P and the centroid location (X_0, Y_0) of the polygon can be found as follows [3]:

$$A_P = \frac{1}{2} \sum_{i=1}^{p_n} (X_i Y_{i+1} - X_{i+1} Y_i), \qquad (2)$$

$$X_0 = \frac{1}{6A_P} \sum_{i=1}^{p_n} (X_i + X_{i+1})(X_i Y_{i+1} - X_{i+1} Y_i), \qquad (3)$$

$$Y_0 = \frac{1}{6A_P} \sum_{i=1}^{p_n} (Y_i + Y_{i+1})(X_i Y_{i+1} - X_{i+1} Y_i). \qquad (4)$$

For a given number of clusters k, the sensing field area is divided into equal subareas (cluster area) and the area of each such subarea is $A_{CL} = A_p/k$. The clustering procedure selects an arbitrary sensor node on the boundary B as the starting point P_1. It then selects the second sensor node P_2 on the boundary in anti-clockwise order. The area bounded by P_1, P_2 and the centroid P_0 is calculated, using Equation (2). If this area is greater than A_{CL}, this means that the required area must be bounded by P_0, P_1 and an intermediate point (a virtual sensor node) P_V that lies on the $\overline{P_1 P_2}$ line, as shown in Figure 1(a). Making use of A_{CL} and the locations of P_0, P_1 and P_2, the location of P_V is calculated as follows:

$$Y_V = \frac{Y_1 - Y_0}{X_1 - X_0} X_0 + Y_0 - \frac{Y_1 - Y_0}{X_1 - X_0} X_0, \quad (5)$$

$$X_V = \frac{1}{Y_0}(X_0 Y_V - 2A_{CL}). \quad (6)$$

If the calculated subarea is less than the cluster area as shown in Figure 1(b), a new sensor node on the boundary of sensing field next to P_2 needs to be added, and the area has to be re-calculated. Ultimately, the sensor nodes of the boundary of the k clusters are stored in BCL. The details of clustering procedure are shown in Figure 2.

Input: k, A_P, B, P_0
Output: Set of boundary cluster sensor nodes BCL

Select an arbitrary sensor node P_1 from B;
$A_{CL} = A_P/k$;
$A = A_{CL}$ store cluster area;
foreach *cluster i, $1 \le i \le k$* **do**
    Flag=False;
    Put P_0 and P_1 as the boundary sensor nodes into BCL_i;
    while *not Flag* **do**
        Select sensor node P_2 next to P_1 in anti-clockwise order;
        if *area(P_0, P_1, P_2) $\ge A$* **then**
            Calculate a virtual sensor node P_V using Eqs. (5) and (6);
            Add P_V as a boundary sensor node into BCL_i;
            $P_1 = P_V$;
            $A = A_{CL}$;
            Flag=True;
        else
            $A = A - area(P_0, P_1, P_2)$;
            Add sensor node P_2 into BCL_i;
            $P_1 = P_2$;
        end
    end
end

Figure 2: The Clustering Procedure.

4.3 The Calculation of the Location of Virtual Cluster Heads

What remains is to determine the candidates for cluster heads. Recall that r is the transmission range of each sensor node, there is a strong relationship between the Euclidean distance d from the sender (a sensor node) to the receiver (its cluster head) and the number of hops within distance d. Each route from a sensor node to its cluster head must meet [4]:

$$\text{Number of hops in a shortest path} \ge \frac{d}{r}. \quad (7)$$

Most routing protocols use hop counting as one of the route selection criteria. These protocols aim to minimize the number of transmissions required to send a packet along the selected path. In addition, if the network is a dense

network, then the minimum number of hops in the shortest path approaches d/r.

To minimize the maximum number of hops between a cluster head and its sensor nodes, the maximum distance between them needs to be minimized. Accordingly, it is assumed that the center of a cluster area is the best location of the cluster head (referred to as a virtual cluster head VCH), which balances the energy consumption among the sensor nodes in the cluster. The locations of VCHs can be found using Eqs. (2), (3) and (4). The tour length L_C consisting of VCHs can then be obtained. If L_c is less than L, these VCHs will be considered, since the mobile BS can move back and forth along the route; otherwise, the tour length must be reduced through relocationing VCHs towards the center of the sensing field area. To achieve a load balance among cluster sensor nodes, the same amount of relocationing of VCHs is employed. The reduction rate is the ratio of L to the calculated tour L_C. Simple geometric equations are derived to find the new VCHs that satisfy the required tour length constraint. Figure 3 and Figure 4 illustrate the procedure for calculation of the location of VCHs and the concept of relocationing VCHs using an example, respectively.

Input: L, k, A_{CL}, BCL, $P_0(X_0, Y_0)$
Output: Set of virtual cluster heads $VCH(XV, YV)$

foreach *cluster i, $1 \le i \le k$* **do**
    Calculate cluster centroid point $PC_i(XC, YC)$ using Eqs (2), (3) and (4) with A_{CL} and BCL_i parameters;
end
L_C = Length of the tour connecting PC points;
if $L_C \le L$ **then**
    $VCH_i = PC_i$, $1 \le i \le k$;
else
    $R = L/L_C$;
    foreach *cluster i, $1 \le i \le k$* **do**
        $l_i = R\sqrt{(XC_i - X_0)^2 + (YC_i - Y_0)^2}$;
        $\theta_i = \arctan(YC_i/XC_i)$;
        $XV_i = l_i \cos(\theta_i)$;
        $YV_i = l_i \sin(\theta_i)$;
    end
end

Figure 3: The Procedure for Calculation of the Location of Virtual Cluster Heads.

4.4 Finding Real Cluster Head Sensor Nodes

So far the locations of VCHs have been calculated. To determine the real cluster heads and the tour consisting of real cluster heads such that the length of the tour is no greater than L, the nearest sensor nodes to each VCH are the candidates for the corresponding real cluster head.

The mobile BS gathers the sensed data while it visits cluster heads along the route. For each cluster, the term 'real segment' RS is used to refer to the length of the segment of the mobile BS tour that connects the sensor node within the current cluster with its previous and next VCHs. We will also refer to the segment as a 'virtual segment' VS, to the length of the segment of the BS tour that connects the VCH of the current cluster with the previous and next VCHs.
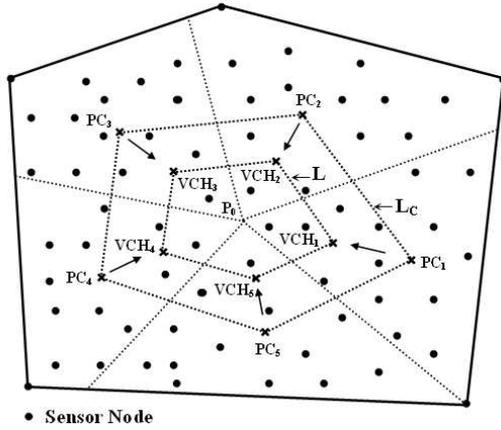
Figure 4: An illustrative example of virtual cluster heads calculation, for $k = 5$. Sensor nodes are denoted by black circles. PC_i, $1 \leq i \leq k$ are the locations of clusters area center points and VCH_i, PC_i, $1 \leq i \leq k$ are the virtual cluster heads. L_C and L are the BS tour length connecting PC_i and VCH_i, respectively. $L < L_C$.

The real and virtual segments are both used as references in order to decide whether the candidate cluster head will increase or decrease the total BS tour length. An example of real and virtual segments is shown in Figure 5.

To find a set of real cluster heads to form a tour such that the tour length is no greater than L, two sensor nodes close to each VCH need to be found. One sensor node increases the length of the tour while the other decreases it. To achieve that, the sensor nodes in each cluster are sorted in increasing order according to their distance from the corresponding VCH. Then, for the first sorted sensor node, the real segment is calculated and compared with the virtual segment. The sensor node is assumed to increase the length of BS tour if the real segment is greater than the corresponding virtual segment; otherwise, it decreases the length of the tour. The checking will continue until the two candidate cluster heads closest to the corresponding VCH are found.
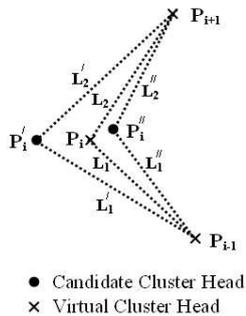
Figure 5: An example shows real and virtual segments. Candidate cluster heads and VCHs are denoted by black circles and crosses, respectively. The virtual segment for P_i is $L_i = L_1 + L_2$, while the real segments for P_i' and P_i'' are $L_i' = L_1' + L_2'$ and $L_i'' = L_1'' + L_2''$, respectively. $L_i < L_i'$ and $L_i > L_i''$.

If a candidate cluster head that decreases the length of the tour cannot be found, the candidate cluster head that increases the tour length is elected as a real cluster head. The details of finding two candidate cluster heads for each VCH are shown in Figure 6.

Input: VCH, k, L
Output: Set of real cluster heads R

Initiate status flag to unfinish for k clusters,
$FCL_i =$False, $1 \leq i \leq k$;
Initiate real cluster head $R_i = VCH_i$, $1 \leq i \leq k$;
foreach *cluster i in k* **do**
 Set a flag for each candidate cluster head, FI=False;
 FD=False;
 VS_i=length of virtual segment connecting VCH_i
 with VCH_{i-1} and VCH_{i+1};
 SN=Set of increasing sort of all cluster sensor nodes
 in i-th cluster according to its distance to VCH_i;
 while *not FI and not FD* **do**
  if *SN is not empty* **then**
   Pick a node n from SN;
   RS_i=length of real segment connecting n
   with VCH_{i-1} and VCH_{i+1};
   if $RS_i \geq VS_i$ **then**
    if *FI=False* **then**
     Add node n into NI_i set;
     Store $\Delta LI_i = RS_i - VS_i$;
     FI=True;
    end
   else
    if *FD=False* **then**
     Add node n into ND_i set;
     Store $\Delta LI_i = RS_i - VS_i$;
     FD=True;
    end
   end
  else
   $R_i = NI_i$;
   $FCL_i = True$;
  end
 end
end

Figure 6: Finding Two Candidate Cluster Heads for Each Virtual Cluster Head- Finding Real Cluster Head Procedure.

The election of real cluster heads proceeds in two phases. In Phase I, the closest candidate cluster head to the corresponding VCH is elected as a real cluster head, if the length of its real segment is less than the corresponding virtual segment. Thus, all not-chosen candidate cluster heads closest to their VCHs have a real segment greater than the corresponding virtual segment. What follows is to search for a real cluster head that has not yet been elected, so that the length of the connected segments is no greater than L. The connected segments are calculated using the candidate cluster head, the elected cluster heads which are obtained so far, and the VCHs of the remaining unelected cluster heads. To find such a cluster head, the candidate cluster heads closest to the VCH of the remaining clusters are sorted in increasing

```
foreach unfinished cluster i in FCL do
    if ΔLI_i > | ΔLD_i | then
        R_i = ND_i;
        FCL_i=True;
    end
end
SN=Set of increasing sort order of candidate cluster
heads in NI_i for the unfinished clusters according to
length ΔLI_i;
foreach unfinished cluster i in FCL do
    Pick a candidate cluster head C from SN;
    L_i=Length of the BS segments connecting real
    cluster heads in R by using C instead of R_i;
    if L_i < L then
        R_i = C;
        FCL_i=True;
    else
        exit loop;
    end
end
```

Figure 7: Phase I of Finding Real Cluster Heads Procedure.

```
foreach unfinished cluster i in FCL do
    R_i = ND_i;
end
foreach unfinished clusters i in FCL do
    SN=Set of increasing sort order of candidate cluster
    heads in NI_i according to its distance to the
    corresponding VCH_i;
    Pick a candidate cluster head C from SN;
    L_i=Length of BS tour segments connecting real
    cluster heads in R using C instead of R_i;
    if L_i < L then
        R_i = C;
        FCL_i=True;
    else
        exit loop;
    end
end
Set FCL=True for all unfinished clusters;
```

Figure 8: Phase II of Finding Real Cluster Heads Procedure.

order according to the difference in length between real and virtual segments ($VS - RS$). For the first candidate cluster head, the total length of the BS tour is calculated. If it is less than L, the candidate cluster head is elected as the real cluster head. For the next candidate cluster head, the total BS tour length is calculated, having taken into account the election of the previous one. Phase I will terminate once it finds a candidate cluster head with the corresponding length of BS tour greater than L. Ultimately, all the remaining candidate cluster heads that are closest to the VCH have RS greater than VS. The details of Phase I are shown in Figure 7. Phase II initially assumes that the remaining candidate cluster heads with RS less than VS are elected as the real cluster heads; therefore the length of BS tour is less than L. But it is possible to find a real cluster head closest

to the VCH, with RS greater than VS and the total BS tour is less than L. To find such a cluster head, the candidate cluster heads that have not been elected from Phase I are sorted in increasing order of distance from their corresponding VCHs. For the first candidate cluster head, the total length of BS tour is calculated, using the real cluster head elected from Phase I, in addition to the initial elected cluster head from Phase II after changing the initial elected cluster head with the corresponding candidate cluster head. If the length of the BS tour is less than L, the candidate cluster head is finalized and elected as the real cluster head. For the next candidate cluster head, the length of BS tour is calculated by taking into account the election of the previous one, and so on. Ultimately, all the initially elected cluster heads are finalized and elected as the real cluster heads. Figure 8 and Figure 9 show the details of Phase II and an example, respectively.
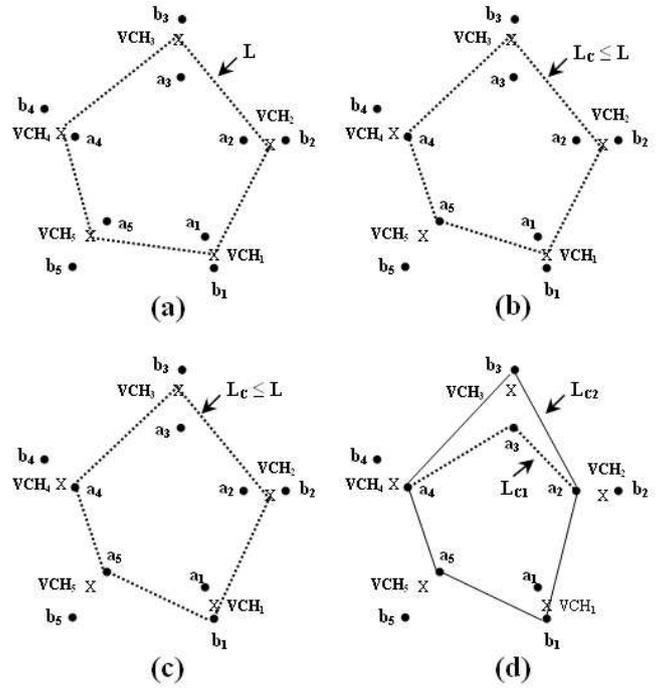
Figure 9: An execution example of finding real cluster heads, $k = 5$. (a) Finding two candidate cluster heads for each cluster, one with $RS_i \geq VS_i$ and the other with $RS_i < VS_i$, $1 \leq i \leq k$. (b) After the execution of Phase I, sensor nodes a_4 and a_5 are elected as real cluster heads since they are closest to VCH_4, VCH_5 and $RS_4 < VS_4$, $RS_5 < VS_5$, respectively. (c) Then, sensor node b_1 is elected as real cluster head since it is closest to VCH_1 and $L_c \leq L$. (d) After the execution of Phase II, sensor nodes a_2 and a_3 are initially elected as real cluster heads, then a_3 is changed with b_3 as real cluster head since it is closest to VCH_3 and $L_{C2} \leq L$, b_2 is not elected as real cluster head since the corresponding BS tour length is greater than L.

4.5 Maximizing Expected Network Lifetime

All non-cluster head sensor nodes send their data to their
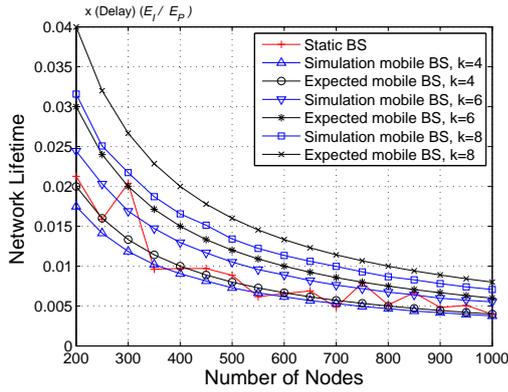
Figure 10: Network lifetime as it varies with the number of network sensor nodes, for static and mobile BSs with different k.

Figure 11: The minimum and maximum energy consumption differences among the cluster heads as the number of network sensor nodes are varied.

corresponding cluster heads. The cluster heads forward the data to the mobile BS when the BS visits the clusters; thus the cluster heads are the bottleneck of the network. Recall that E_I is the sensor node initial energy, if E_p is the average amount of energy required to transmit one packet, and D is the time required for the BS to gather data from sensor nodes in one round, then the expected network lifetime is

$$E(Lifetime) = \frac{kE_I D}{nE_p}, \tag{8}$$

where Equation (8) represents the maximum network lifetime that can be achieved when the n sensor nodes are evenly distributed in k clusters.

5. PERFORMANCE EVALUATION

In this section we evaluate the performance of the proposed algorithm through simulations with Matlab, assuming that the effect of the MAC layer is ignored.

We assume that sensor nodes in the network are randomly deployed within a 600×600 unit square with uniform distribution. Each sensor node has a transmission range of $r = 100$ unit and the initial energy of E_I unit. All data packets have a fixed length and take E_P unit energy per packet. D_I is defined as the time required for the mobile BS to take a tour with length equal to a half of the perimeter of the sensing field. The speed of the mobile BS is assumed $v = 1$ velocity unit. For each instance of deployment, the network lifetime, the percentage of the difference in energy consumption among the cluster heads, the maximum energy consumption of neighboring sensor nodes to cluster head and the maximum number of routing hops from a sensor node to the mobile BS are calculated. The result is the average over 100 instances for each set of network nodes.

Figure 10 shows the network lifetime delivered by using the mobile BS compared with the static BS, assuming that the static BS is located at the centroid of the sensing field area. Breadth First Search (BFS) algorithm is used to find a routing tree rooted at the BS. In the static BS, the BS neighboring sensor nodes consume more energy than any other sensor nodes in the network since they have to relay the packets received from child sensor nodes to the BS, while in the mobile BS, the cluster heads consume more
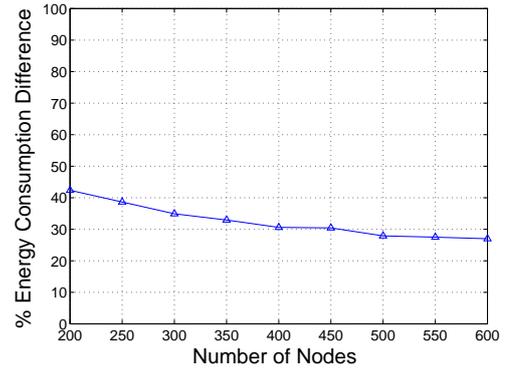
energy than the other sensor nodes in the network. The network lifetime using the static BS is compared with the theoretical and simulated network lifetime of the mobile BS with different number of clusters. The expected network lifetime, as described by Equation (8), is used to represent the maximum network lifetime that can be achieved for a given number of clusters and sensor nodes. The result shows that the network lifetime decreases as the number of sensor nodes increases. This makes sense since that increases the number of packets need to be forwarded to the BS. The result also shows that the network lifetime is longer for the static BS than that for the mobile BS when the number of clusters is $k = 4$ and $k = 6$. This is expected because the number of neighboring sensor nodes of the BS is larger than the number of cluster heads. Therefore, the maximum number of data packets that the cluster heads require to forward is higher than that the neighboring sensor nodes of the BS required to forward. It can also seen that as the number of nodes increases, the difference between the theoretical and simulation network lifetime decreases. The reason for this decrease is that the distribution of nodes in the clusters area becomes more uniform, with the increases in the number of sensor nodes. That means the number of packets the cluster heads need to forward become more balanced. Therefore, the simulated network lifetime tends to reach the theoretical expected network lifetime.

To evaluate the balanced energy consumption among the cluster heads, we need to calculate the percentage difference in energy consumption among the cluster heads, by calculating the ratio of the difference in cluster head energy consumption to the average energy consumption, as shown in Figure 11. The result shows that the load on the cluster heads becomes more balanced as the number of sensor nodes increases and tends to reach the ideal case when the number of sensor nodes in each cluster is equal.

To study the effect of the data gathering delay on the cluster sensor nodes load balance, the energy consumption for neighboring sensor nodes of a cluster head is calculated as shown in Figure 12. The BFS algorithm is used again to find the routing tree for each cluster where the cluster head is the root of the tree. The curves in Figure 12 show that the energy consumption increases, with the decrease in the end-to-end data gathering delay due to the decreasing
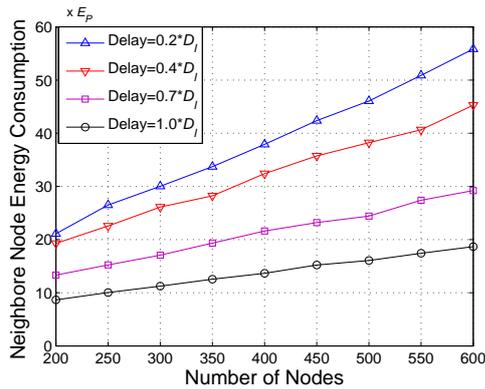
Figure 12: The minimum cluster head neighboring sensor nodes energy consumption as the number of network sensor nodes are varied, for various data gathering delay.
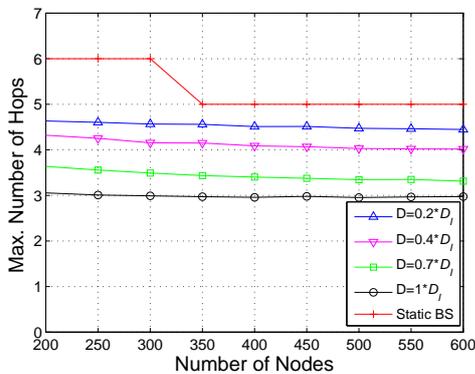
Figure 13: The maximum number of hops as it varies with the number of network sensor nodes, for static and mobile BSs.

length of BS tour towards the centroid of the sensing field area. Therefore, it leads to an increase in the number of sensor nodes, that the neighboring sensor nodes are responsible for forwarding their data packets to the cluster head. It is also shown that the energy consumption increases, with the increase in the number of network nodes.

Figure 13 illustrates the number of relay hops for the sensing data to reach the BS. To find the maximum number of hops, we have to consider the number of hops of sensor nodes located near the border of the cluster area for the mobile BS case, while the sensor nodes near the border of the entire sensing area are considered for the static BS case. The maximum number of hops increases with the decrease in data gathering delay for the mobile BS, since that increases the distance between a sensor node and its cluster head. The result shows that the maximum number of hops is still less than that for the static BS. Nevertheless, it also shows that the number of sensor nodes has a small effect on the maximum route length, due to the fact that the maximum number of hops is equal to the length of the shortest path, since the density of the sensor nodes in the network is high.

6. CONCLUSION

In this paper we dealt with the problem of data gathering in the mobile BS environment subject to the sensing data needed to be gathered at a specified delay. We presented a clustering-based heuristic algorithm to balance the energy consumption among sensor nodes. The proposed algorithm allows the BS to visit all cluster heads within a specified delay. Simulation is performed to evaluate the performance of the proposed algorithm against the static BS case and to evaluate the distribution of energy consumption among the cluster heads. The result has shown that, when incorporated with clustering, the use of clustering with a mobile BS increases the network lifetime significantly. Furthermore, the proposed solution for finding cluster heads results in a uniform balance of energy depletion among cluster heads.

7. REFERENCES

[1] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *GLOBECOM03, IEEE*, 2003.

[2] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang. Towards mobility as a network control primitive. In *MobiHoc04*, May 2004.

[3] P. S. Heckbert. *Graphics Gems IV*. Academic Press, 1994.

[4] B. Ishibashi and R. Boutaba. Topology and mobility considerations in mobile ad hoc networks. *Ad Hoc Networks*, 3(6):762–776, 2005.

[5] D. Jea, A. Somasundara, and M. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 244–257, June 2005.

[6] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services, ACM, MobiSys04*, pages 111–124, 2004.

[7] J. Luo and J.-P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *Proc. INFOCOM05, IEEE*, March 2005.

[8] M. Ma and Y. Yang. Sencar: An energy-efficient data gathering mechanism for large-scale multihop sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(10):1476–1488, October 2007.

[9] A. A. Somasundara, 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):958–973, August 2006.

[10] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava. Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):395–410, April 2007.

[11] R. Sugihara and R. K. Gupta. Improving the data delivery latency in sensor networks with controlled mobility. In *Proc. of the 4th IEEE International conference on Distributed Computing in Sensor Systems, DCOSS '08*, pages 386–399, 2008.

[12] B. Sun, S.-X. Gao, R. Chi, and F. Huang. Algorithms for balancing energy consumption in wireless sensor networks. In *Proc. of the 1st ACM international workshop on Foundations of wireless ad hoc and sensor networking and computing, FOWANC '08*, pages 53–60, May 2008.

[13] M. Tariq, M. Ammar, and E. Zegura. Message ferry route design for sparse ad hoc networks with mobile nodes. In *MobiHoc*, pages 37–48, May 2006.

[14] G. Xing, T. Wang, W. Jia, and M. Li. Rendezvous design algorithms for wireless sensor networks with a mobile base station. In *MobiHoc 08*, pages 231–240, 2008.

[15] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks:a hybrid, energy-efficient approach. In *proc. of the twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM*, pages 629–640, March 2004.

[16] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc '04*, pages 187–198, 2004.

[17] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *Proc. of the 24th IEEE Computer and Communications Societies, INFOCOM 2005*, volume 2, pages 1407–1418, 2005.