Load Balance and Energy Efficient Data Gathering in Wireless Sensor Networks

Devendar Mandala†, Fei Dai†, Xiaojiang Du‡and Chao You†
Department of Electrical and Computer Engineering†
Department of Computer Science‡
North Dakota State University, Fargo, ND 58105

Abstract— **Many data gathering protocols for wireless sensor networks use clustering technology for prolonging the network lifetime. Cluster-based protocols reduce the total energy consumption via data aggregation, and balance energy consumption via clusterhead rotation. However, most existing protocols focus on load balance within each cluster. The energy consumption of the entire network is still unbalanced, and this uneven energy dissipation can significantly reduce the network lifetime. We propose an even energy dissipation protocol (EEDP) for efficient cluster-based data gathering in wireless sensor networks. In EEDP, the sensor data are forwarded to the base station via multiple chains of clusterheads. Each chain uses a rotation scheme to balance energy consumption among clusterheads and avoid the formation of a hot spot. We developed efficient algorithms to organize clusterheads into multiple chains, such that the traffic load is evenly distributed among different chains. Analysis and simulation results show that EEDP achieves better load balance than several existing protocols and significantly increases network lifetime.** [1]

I. Introduction

A fundamental problem in wireless sensor networks (WSNs) is to maximize the network lifetime under given energy constraints. Network lifetime is defined as the time elapsed until the first node in the network completely depletes its energy. A sensor node is typically battery operated and therefore constrained in energy. The major role of a WSN is data gathering; that is, for every node to send its sensor data to the base station. Numerous energy efficient protocols have been proposed to route sensor data to the base station. Many of them take a cluster-based approach [1], [2], [3]: a few sensor nodes are elected as clusterheads to collect data from their neighboring nodes (called cluster members). The data traffic can be greatly reduced by applying data aggregation [4] at clusterheads. Data aggregation in WSNs is defined as the process of combining multiple data packets into one packet based on correlation in data. Cluster members have low energy consumption, as they transmit sensor data to a nearby clusterhead. For better load balance, the role of clusterhead is rotated among cluster members.

Although existing cluster-based protocols achieve good load balance in a small area, the energy dissipation is unbalanced in the entire network. Two popular strategies to route data from clusterheads to the base station are direct connection [1] and shortest path routing [3]. In direct transmission, as shown in Figure 1(a), nodes far away from the base station dissipate their energy much faster than those close to the base station. In the shortest path routing, as shown in Figure 1(b), the small area close to the base station form a hot spot that relays sensor data for the entire network. In both cases, one portion of the network dies before the others. Obviously, the network lifetime can be further improved by using a more balanced routing strategy.

We propose an cluster-based even energy dissipation protocol (EEDP) that balances energy consumption among different areas of the network. In EEDP, clusterheads are organized into several parallel chains, as shown in Figure 1(d). The intra-chain routing scheme is similar to shortest path routing: each node forwards its data and its predecessors' data to its successor, and the last node forwards the data to the base station. However, based on the intra-chain scheduling scheme, each node will occasionally skip its successors and transmit directly to the base station. This scheme balances energy consumption and avoids forming a hot spot. Note that a chain-based scheme [5], as shown in Figure 1(c), has been proposed before. However, this scheme assumes network wide data aggregation, which is not practical in many applications [6].

To form balanced chains, the network is divided into tiers based on the distance to the base station. Each chain contains one clusterhead from each of these tiers. We developed two algorithms for each chain to select clusterheads. The first one is a fast heuristic algorithm; the second is optimal in terms of balanced energy consumption. The performance of EEDP is evaluated via numeric analysis and simulations. Both studies show that EEDP outperforms existing cluster-based data gathering protocols, such as LEACH and HEED, in terms of better load balance and longer network lifetime.

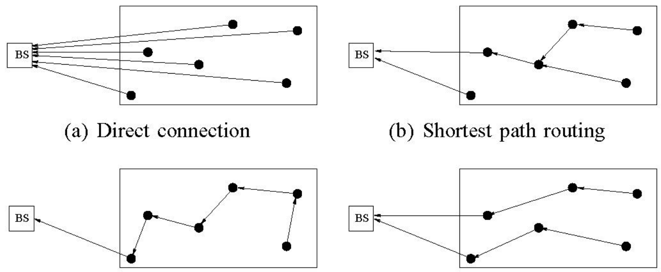
II. Related Work

Existing cluster-based data gathering protocols consist of two components: a clusterhead election and rotation scheme for effective data aggregation and an inter-cluster routing scheme that delivers the aggregated data to the base station. Existing cluster election schemes include:

Random selection, which is used in LEACH [1]. In this scheme, each node v becomes a clusterhead with a probability $p(v)$ that depends on the expected number of clusterheads and previous election results. Random election is simple and

[1]This work was supported in part by NSF EPSCoR grants EPS 0447679.
Contact address: fei.dai@gmail.com

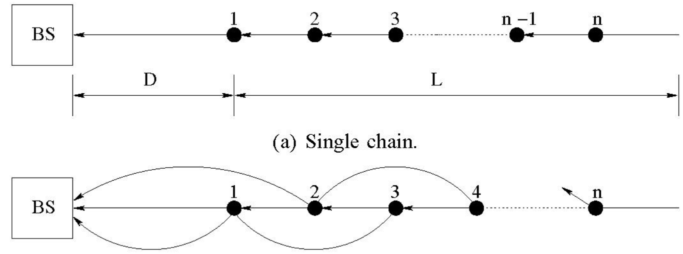
1-4244-0507-6/06/$20.00 ©2006 IEEE



(a) Direct connection (b) Shortest path routing

(c) Single chain (d) Multiple chain

Fig. 1. Inter-cluster routing strategies.



(a) Single chain.

(b) Two chains.

Fig. 2. Multiple chain formation in the one-dimensional network.

incurs little cost. Its major drawback is that the resultant set of clusterheads may be unevenly distributed, which causes variable cluster sizes and higher intra-cluster communication cost.

Dominant set formation, which is used in HEED [3] and EEUC [2]. A dominating set (DS) [7] is a group of nodes that covers the entire network; that is, every node in the network is either in the dominating set, or a neighbor to a node in the dominating set. Traditional DS formation algorithms [8] elect nodes with local maximum properties (e.g., maximal node degree or minimal node ID) as clusterheads, and have a high time complexity in large networks. In HEED, a probabilistic algorithm is employed to form a DS in a fixed number of rounds. This scheme builds higher quality clusters than random selection. The penalty is the higher election overhead due to information exchanges among neighbors.

In our simulations, we use DS formation in all protocols for a fair comparison of inter-cluster routing strategies. Existing routing structures fall into the following categories.

Direct connections, where each clusterhead transmits aggregated sensor data directly to the base station, as shown in Figure 1(a). This scheme is used by LEACH. The major problem of this simple strategy is the uneven energy consumption. Clusterheads far away from the base station have to transmit data over a long distance and suffer a high energy consumption rate. In a large network, such a disparity will cause nodes in the far corners of the sensing area to die quickly, leaving these corners un-monitored.

Shortest path tree. In HEED, all clusterheads send aggregated data to the base station via the shortest path. These paths form a shortest path tree (SPT), as shown in Figure 1(b). This scheme minimizes the total energy consumption. However, neighbors of the base station have high load and form a hot spot. When nodes in this area deplete their energy, not only does their energy depletion create a un-monitored spot, it will also disconnect other parts of the network from the base station.

Single chain. Although this scheme is used by PEGASIS [5], a non-cluster-based data gathering protocol, it can be potentially used in a cluster-based scheme. As shown in Figure 1(c), a single chain is formed connecting all the clusterheads. Each clusterhead communicates only with the closest neighbor and takes turns in transmitting data to the base station. This scheme assumes global data aggregation, i.e., the sensor data from all nodes can be aggregated into a single packet. This is a strong assumption that is not always true. When it is not true, the cost of passing each packet along the entire chain will cause a very short network lifetime.

III. Balanced Inter-Cluster Routing

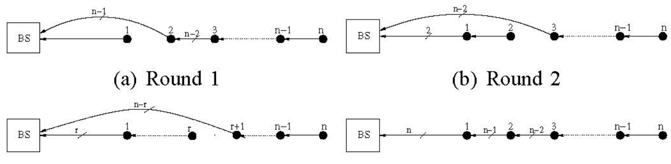
We propose an even energy dissipation protocol (EEDP) for energy efficiency and balanced inter-cluster routing. The basic idea is to form multiple chains of clusters as shown in Figure 1(d). Traffic load is evenly distributed among these chains to avoid a single hot spot. Each chain contains nodes far away from as well as those close to the base station, and uses a scheduling scheme to balance the energy consumption of each clusterhead. The total energy consumption of all nodes in EEDP is slightly higher than the shortest path routing, but the maximal energy consumption of a single node is significantly lower, which means a longer network lifetime.

In this work, we consider a WSN where all nodes are location aware and have the equal initial energy and similar capabilities; sensor nodes are dispersed randomly in the sensing field, and the base station location is fixed and outside of the sensor field. For the ease of discussion, we first use a simple one-dimensional network model in this section, where multiple chain formation is a trivial task. The intra-chain scheduling strategy will be discussed and its performance be compared with LEACH and HEED via a numeric analysis under this model. In the next section, we will discuss the general case of two-dimensional random networks.

A. Intra-cluster scheduling

In the simplified one-dimensional network model, the sensing field is a line of length L. n clusterheads $1, 2, \ldots, n$ are placed from left to right on this line, as shown in Figure 2(a). The distance between any two adjacent clusterheads is L/n. The base station (BS) is to the left of the sensing field. The distance between BS and node 1 is D. The task of forming balanced chains in the above model is trivial. To form m chains, one can simply select nodes $i, m+i, 2m+i, \ldots$ for each chain i ($1 \leq i \leq m$). Figure 2 shows the formation of one and two chains in such a network.

We use the single chain scenario to explain our intra-chain scheduling scheme (Algorithm 1). Note that this scheme can be used in scenarios of multiple chains and two-dimensional



(a) Round 1 (b) Round 2

(c) Round r (d) Round n

Fig. 3. Intra-chain scheduling. The label of each link represents the number of packets forwarded via this link.

networks without modification. In a chain with n nodes, the data gathering activities is divided into n rounds. To alleviate the hot spot problem, the chain is split into two sub-chains except for the last round. In each round r, only r packets are relayed via node 1, the node closest to BS, and the remaining $n-r$ packets take a short cut from node $r+1$. As the burden of relaying sensor data to BS is distributed among all nodes in the chain, node 1 is not a hot spot as in the shortest path routing. Unlike the direct connection scheme, node n, which is the farthest away from BS, will not consume much more energy either, as it directly communicates with BS in only one out of n rounds.

Algorithm 1 Intra-Chain Scheduling

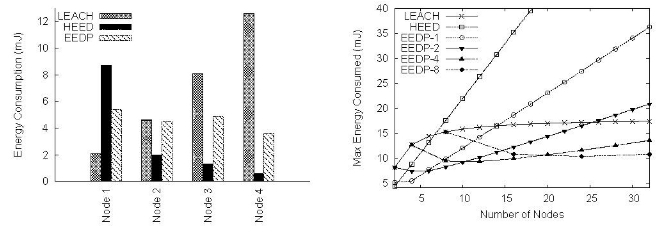
```
for each round r ← 1 to n do
    if r < n then
        Route packets via two chains (r, r − 1, ..., 1, BS)
        and (n, n − 1, ..., r + 1, BS).
    else
        Route packets via a single chain
        (n, n − 1, ..., 2, 1, BS).
    end if
end for
```

Figure 3 shows an example of intra-chain scheduling in a single chain. For the first round, clusterhead 1 forwards only one packet to BS and the remaining $n-1$ packets from clusterheads 2, 3, ..., n are relayed to BS by clusterhead 2. For the second round, clusterheads 1 and 2 forward 2 and $n-2$ packets respectively to BS. For any round r, clusterheads 1 and $r+1$ forward r and $n-r$ packets respectively to BS. At the last round n, there is no splitting of the chain and the clusterhead 1 forwards n packets to BS.

When a single chain is used and when n exceeds a certain threshold, the relaying overhead may overweight the benefit of the scheduling scheme, i.e., a hot spot will be formed at clusterhead 1. In this case, a multiple chain structure is more attractive, where the average number of relays is under control.

B. Performance analysis

In this subsection, we compare the performance of EEDP with direct connection (LEACH) and the shortest path routing (HEED) using the one-dimensional network model. Simulation results for two-dimensional networks will be presented in Section IV. Our focus is on load balance in terms of the maximal single node energy consumption. A scheme with a low maximal energy consumption value is better balanced and has a longer network lifetime. We show that EEDP has the best performance of all these schemes.



(a) Energy consumption distribution of 4 clusterheads. (b) Maximal energy consumption.

Fig. 4. Load balance in 1-D network ($l = 2000, D = 100, L = 200$).

We use the same energy model as considered in [1]: To transmit a l-bit packet to a distance d, the corresponding transmission power $E_{tx}(d) = E_{elec}l + E_{amp}ld^2$ and receiving power $E_{rx} = E_{elec}l$, where $E_{elec} = 50nJ/bit$ and $E_{amp} = 100pJ/bit/m^2$. In LEACH, the energy to deliver a packet at node k is

$$E^{dc}(k) = E_{tx}(D + \frac{k-1}{n}L) \tag{1}$$

In HEED, each node k receives $n-k$ packets from a previous hop, and transmits $n-k+1$ packets to the next hop. That is,

$$E^{spt}(k) = \begin{cases} (n-1)E_{rx} + nE_{tx}(D) & : k = 1 \\ (n-k)E_{rx} + (n-k+1)E_{tx}(\frac{1}{n}L) & : k > 1 \end{cases} \tag{2}$$

Now we consider EEDP using a single chain. Since the energy consumption varies in each of the n rounds, we calculate the average cost per round at a node k.

$$E^{eedp}(k) = \frac{1}{n}\sum_{r=1}^{n}[N_{rx}^{k,r}E_{rx} + N_{tx}^{k,r}E_{tx}(\frac{1}{n}L) + N_{tx'}^{k,r}E_{tx}(D + \frac{k-1}{n}L)] \tag{3}$$

where $N_{rx}^{k,r}$, $N_{tx}^{k,r}$, and $N_{tx'}^{k,r}$, respectively, are the number of packets received, transmitted to the successor, and transmitted to the base station by node k in round r. From Algorithm 1,

$$N_{rx}^{k,r} = \begin{cases} n-k & : r \le k \\ r-k-1 & : r > k \end{cases} \tag{4}$$

$$N_{tx}^{k,r} = \begin{cases} 0 & : k = 1 \vee k = r \\ N_{rx}^{k,r} + 1 & : otherwise \end{cases} \tag{5}$$

$$N_{tx'}^{k,r} = \begin{cases} 0 & : k \neq 1 \wedge k \neq r \\ N_{rx}^{k,r} + 1 & : otherwise \end{cases} \tag{6}$$

For example, consider a one-dimensional network with $n = 4$, $L = 200$, $D = 100$, and $l = 2000$. As shown in Figure 4(a), the maximal and minimal energy consumption in LEACH are $12.6mJ$ (at node 4) and $2.1mJ$ (at node 1), respectively. Those for HEED are $8.7mJ$ (node 1) and $0.6mJ$ (node 4). In EEDP, the maximal single node energy consumption per round is $5.4mJ$ at node 1, and the minimal energy consumption is

$3.6mJ$ at node 4. Obviously, EEDP is more balanced than the other schemes. In this specific case, the network lifetime of EEDP is 60% longer than that of HEED, and more than twice that of the LEACH.

Then we consider the case of using m chains in EEDP. For simplicity, we assume n is a multiple of m and $n = n'm$. For the trivial chain formation scheme described in Section III-A, each chain i consists of node $i, m+i, \ldots, (n'-1)m+i$. The energy consumption of each node $k = k'm + i$ is

$$E^{eedp}(m,k) = \frac{1}{n'}\sum_{r=1}^{n'}[N_{rx}^{k',r}E_{rx} + N_{tx}^{k',r}E_{tx}(\frac{1}{n'}L) + N_{tx'}^{k',r}E_{tx}(D + \frac{k-1}{n}L)] \qquad (7)$$

where $N_{rx}^{k',r}$, $N_{tx}^{k',r}$, and $N_{tx'}^{k',r}$ are defined by equations (4-6).

In order to compare the load balance of different inter-cluster routing schemes, we compute the maximal single node energy consumption per round in LEACH, HEED, and EEDP using one (EEDP-1), two (EEDP-2), four (EEDP-4) and eight (EEDP-8) chains, in a one dimensional network with $L = 200$, $D = 100$, $l = 2000$, and the number of clusterheads n varying from 2 to 32. As shown in Figure 4(b), for each cluster number n, there exists a best chain number m, such that using m chains in EEDP outperforms both LEACH and HEED routing schemes.

IV. Tier-Based Multiple Chain Formation

When applying EEDP to random two-dimensional sensor networks, the intra-chain scheduling algorithm discussed in the previous section can be used without modification. However, the trivial chain formation scheme is no longer practical and must be replaced by more sophisticated schemes. We propose two heuristic chain formation algorithms, both based on dividing the sensor network into vertical strips called *tiers*. Clusterheads in the same tier have similar distances to the base station, as shown in Figure 5. While forming multiple chains, each chain selects one clusterhead from each tier. The difference between the two proposed algorithms is the selection method. The first is a simple greedy algorithm that tries to minimize the distance between each node and its successor in a chain. Then an optimal but slower algorithm is proposed to guarantee minimal maximal distance between two consequent clusterheads in each chain.

A. Tiers and chains

In order to divide the network into tiers, we first sort the clusterheads by the ascending order of their distances to the base station, and give them labels $1, 2, \ldots, n$, where node 1 is the closest to the base station, and node n is the farthest. When forming m chains, the network will be divided into tiers $T_1, T_2, \ldots T_{\lceil n/m \rceil}$.

Definition 1 (Tier): The h-th tier of a sensor network is $T_h = \{(h-1)m+1, (h-1)m+2, \ldots, hm\}$. Each clusterhead $i \in T_h$ is called a h-hop clusterhead.

In the above definition, the number of chains m is also called *tier width*. Figure 5 shows a sample network with

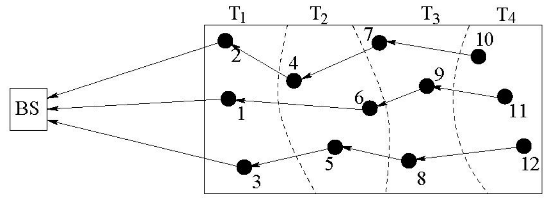


Fig. 5. Tiers and multiple chain formation.

$n = 12$ clusterheads. When $m = 3$, the network is divided into $\lceil 12/3 \rceil = 4$ tiers: $T_1 = \{1, 2, 3\}, T_2 = \{4, 5, 6\}, T_3 = \{7, 8, 9\}$, and $T_4 = \{10, 11, 12\}$.

Forming multiple chains is equivalent to the process of selecting a successor $\pi(i)$ for each non-1-hop clusterhead i ($m < i \leq n$). After the successor selection, a clusterhead j is called a *chainhead* if it is not a next hop of any other clustherhead, i.e., $\nexists i : \pi(i) = j$. For convenience, we use $\pi^{(k)}(i)$ to denote the k-th successor of node i. Specifically, $\pi^{(0)}(i) = i$, $\pi^{(1)}(i) = \pi(i)$, and $\pi^{(2)}(i) = \pi(\pi(i))$.

Definition 2 (Chain): Given a chainhead i, the sequence of its successors $(i, \pi^{(1)}(i), \pi^{(2)}(i), \ldots, \pi^{(k-1)}(i))$ is called a chain and denoted as $Chain(i)$.

In Figure 5, nodes 10, 11, and 12 are chainheads. For chainhead 10, the sequence of its successors are $\pi^{(1)}(10) = 7$, $\pi^{(2)}(10) = 4$, and $\pi^{(3)}(10) = 2$; that is, $Chain(10) = (10, 7, 4, 2)$.

B. Greedy successor selection

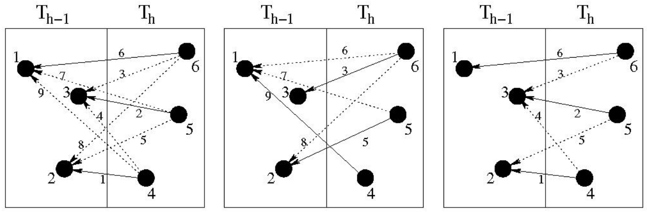
Our first chain successor selection method (Algorithm 2) is a greedy one. Each node selects the closest node from the next tier (T_{h-1}). When one node j in the next tier is the closest with respect to several nodes in the current tier (T_h), the node with the highest label (i.e., the one farthest to the base station) wins and marks j as selected. The other competing nodes have to select from the remaining unmarked nodes in the (T_{h-1}).

Algorithm 2 Greedy Successor Selection

```
Mark all clusterheads as unselected
for i ← n down to m + 1 do
    h ← ⌈i/m⌉
    π(i) ← an unselected clusterhead in T_{h−1} that is closest
        to i.
    Mark π(i) as selected.
end for
```

When applying Algorithm 2 to the WSN in Figure 5, clusterhead 12, which is farthest from the base station, is the first to select its successor. Clusterhead 12 selects the closest clusterhead 8 in the next tier (T_3) and marks it as selected. Then the clusterhead 11 selects the closest unselected clusterhead 9 in the next tier (T_3), after which 10 selects 7. Similarly, clusterheads 9, 8, 7, 6, 5, and 4 selects 6, 5, 4, 1, 3, and 2 respectively as their next hop.

Although Algorithm 2 is easy to implement and has low ($O(mn)$) computation complexity, there are cases where it



(a) Balanced (b) Greedy (c) MinMax

Fig. 6. Greedy and MinMax successor selection. The label associated with the link represents the rank based on ascending order of distances.

Algorithm 3 MinMax Matching

$V \leftarrow T_h \cup T_{h-1}$
$E \leftarrow \{(i,j) | i \in T_h \wedge j \in T_{h-1}\}$
Sort E by edge weight (distance)
for $k \leftarrow |T_h|$ **to** $|E|$ **do**
  $E_k \leftarrow$ the set of k minimal weight edges in E
  Compute a maximum cardinality matching M in bipartite graph $G_k = (V, E_k)$
  if $|M| = |T_h|$ **then**
    $\pi(i) \leftarrow j \quad : \quad \forall (i,j) \in M$
    return
  end if
end for

fails to form balanced chains. For example, consider the case when Algorithm 2 is applied to two adjacent tiers as shown in Figure 6. To form balanced chains, clusterheads 4, 5, 6 must select clusterheads 2, 3, 1, respectively, as their successors in the next tier. When Algorithm 2 is applied, clusterhead 6 which is the farthest from the base station selects the closest clusterhead 3 in next tier, clusterheads 5 and 4 then select clusterhead 2 and 1 in the next tier respectively. The chains formed are therefore unbalanced as shown in Figure 6(b). We will give an algorithm that guarantees balanced successor selection in the next subsection.

C. Balanced successor selection

We propose an optimal solution (Algorithm 3) to minimize the maximal distance between a node and its successor in the multiple chain formation. This task can be modeled as a matching problem in bipartite graphs [9]. An undirected graph $G = (V, E)$ is a bipartite graph if the vertex set V is the union of two disjoint sets V_1 and V_2 such that no two adjacent vertices belongs to the same set. G is a complete bipartite graph if every vertex in V_1 is adjacent to all vertices in V_2. Given a bipartite graph G, a matching $M \subseteq E$ is a set of vertex pairs where each vertex appears at most once. A perfect matching is one that covers every vertex in V.

Selecting successors for nodes in each T_h can be viewed as a matching problem in a complete bipartite graph $G = (T_h \cup T_{h-1}, E)$. Each edge $(i, j) \in E$ is associated with a weight $d(i, j)$ which is the distance between a h-hop node i and a $(h-1)$-hop node j. Our goal is to find a perfect matching M such that the maximal weight of edges in M is minimized. Traditional algorithms exist to compute a matching with the maximal total weight [10] or cardinality [11]. However, no existing method finds a perfect matching with minimal total cost or minimal maximum cost.

In Algorithm 3 a bipartite graph $G_k = (T_h \cup T_{h-1}, E_k)$ is grown by adding edges in the ascending order of distance. The graph stops growing when a perfect matching is found, which is used to select successors for all nodes in T_h. It guarantees minimal maximum distance: it is impossible to find a perfect matching using only edges in E_{k-1}. When Algorithm 3 is applied to the WSN in Figure 6, a bipartite graph with perfect matching is found after 6 edges are added, as shown in Figure 6(c).

Algorithm 3 has a time complexity of $O(m^{9/2})$. In the worst case, the for-loop is executed $|T_h| \times |T_{h-1}| = O(m^2)$ times, and the most time consuming part in the for-loop is to compute a maximum cardinality matching, which complexity is $O(m^{5/2})$ [11]. The total time to process all n/m tiers is $O(nm^{7/2})$. The overall time complexity can be reduced to $O(nm^{3/2} \log m)$ when using a binary search to replace the linear search process, but is still slower than the greedy algorithm. It is also more complex and harder to implement.

V. SIMULATION

We compare the performance of EEDP, in terms of network lifetime, with LEACH and HEED, via a simulation study.

A. Implementation

All protocols are simulated via a custom simulator written in C++. The simulator generates random WSNs by randomly scattering 600 nodes in a $100m \times 200m$ rectangular sensing area. The base station is located outside of the sensing area, and is by default $100m$ from the left side of the rectangle. A dominating set formation algorithm, similar to the one used by HEED, is used to elect clusterheads, using a coverage radius (r) of $25m$.

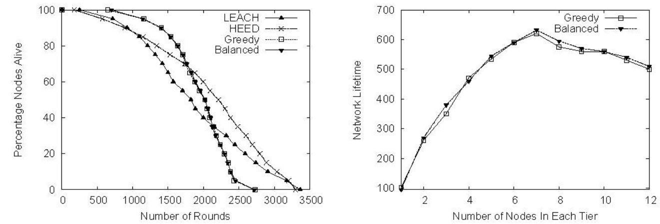
In the beginning of each simulation, each node has an initial battery power of $1J$. During each round of simulation, each cluster member sends a packet to its clusterhead, and the clusterheads use an inter-cluster routing scheme to forward the aggregated data to the base station. The energy model discussed in Section III-B is used to calculate the energy consumption of each transmission and reception, assuming that all data packets, aggregated or non-aggregated, have a fixed length of 2000 bits. Four inter-cluster routing strategies are simulated: LEACH, HEED, EEDP with greedy chain successor selection (Greedy), and EEDP with balanced successor selection (Balanced).

The lifetime of an individual node is measured as the number of rounds before this node depletes its battery power. We define the network lifetime as the number of rounds when the first node dies. Some WSNs can continue functioning when a certain percentage of nodes die. We also measure the network lifetime when 50% of nodes die and when all nodes die. All simulation results are means of 25 tries.

TABLE I

NETWORK LIFETIME ($m = 7$).

Number of Rounds	First death	50% dead	All dead
LEACH	254	1820	3400
HEED	173	2210	3350
Greedy	620	2020	2740
Balanced	632	2030	2750



(a) Percentage of alive nodes. (b) Lifetime vs. tier width.

Fig. 7. Simulation results

B. Results

Routing strategy. The energy efficiency and load balance of all four routing strategies are compared in Table I. The number of chains (m) in EEDP is selected for maximal load balance and is based on experiment results. Compared with LEACH and HEED, EEDP shows significant improvement in terms of when the first node dies. Among the two EEDP variations, the balanced successor selection method is slightly better than the greedy method. Although the greedy algorithm can produce very unbalanced selections as previously discussed, its overall performance is quite close to the optimal one. It may be more practical to implement the simple greedy algorithm than the optimal but more complex one. Both EEDP variations outperform LEACH when 50% of nodes dead, but their performance is close to HEED, which has the lowest total energy consumption. Finally, in the case of all nodes dead, HEED and LEACH show an improvement over EEDP, as in EEDP the total energy consumption per round is higher.

Figure 7(a) shows the percentage of nodes alive over the simulation time. Both EEDP variations improve the network lifetime over LEACH and HEED. The EEDP variations show a more steep curve where 80% nodes die between rounds 1500-2500. on the other hand, LEACH and HEED show a gradual decreasing curve between 800-3100 rounds where 80% nodes die. Obviously, the energy consumption in EEDP is more balanced.

Number of chains. In the second set of simulations, we try to find the optimal number of chains (m) in EEDP that maximize the network lifetime. Figure 7(b) shows the results when clusters of $r = 25$ are formed, where the average number of clusterheads (n) is 28. Initially the network life increases as the m increases, and reaches the peak value when $m = 7$. After that, the network lifetime decreases when m continues to increase. For small values of m, as the number of clusterheads increases, the hot spot is formed at the clusterhead of the chain that is closest to base station, as it has to relay a large number of data packets to the base station. For large values of m, EEDP performs close to LEACH and when m equals the total number of clusterheads, EEDP is the same as LEACH. Note that from the one-dimensional analysis in Section III-B, forming 8 chains has the highest performance when $n = 28$, which is quite close to simulation results.

Simulation results can be summarized as follows: 1) The network lifetime of EEDP is significantly longer than LEACH and HEED when measured as the time that the first node dies, and in case of 50% nodes dead EEDP shows improvement over LEACH and is close to HEED. 2) The performance of the simple greedy chain successor algorithm is similar to that of the optimal but complex algorithm. 3) In EEDP, it is important to find an optimal number of chains that maximize the network lifetime.

VI. CONCLUSION

We have proposed a hybrid inter-cluster routing strategy for energy efficient and balanced data gathering in WSNs. In this new data gathering protocol (called EEDP), every clusterhead alternates direct communication and multi-hop relaying methods in forwarding aggregated sensor data to the base station. This hybrid strategy achieves a fair distribution of communication cost among clusterheads in different areas of a network. It avoids the formation of hot spots that usually cause the early death of some nodes, and expands the overall network lifetime. Numeric analysis and simulation results confirm that EEDP outperforms LEACH and HEED.

REFERENCES

[1] W. R. Heinzeman, A. Chandrakasan, and H. Balakrishnan, “Energy efficient communication protocol for wireless microsensor networks.” *In Proc. of HICSS*, Jan. 2000.

[2] C. Li, M. Ye, G. Chen, and J. Wu, “An energy efficient unequal clustering mechanism for wireless sensor networks.” *In Proc. of MASS*, Nov. 2005.

[3] O. Younis and S. Fahmy, “Distributed clustering in ad hoc sensor networks: A hybrid, energy-efficient approach.” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, Dec. 2004.

[4] V. Erramilli, I. Matta, and A. Bestavros, “On the interaction between data aggregation and topology control in wireless sensor networks.” *In Proc. of SECON*, pp. 557–565, Oct. 2004.

[5] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics.” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, Sep. 2002.

[6] P. Rickenbach and R. Wattenhofer, “Gathering correlated data in sensor networks.” *In Proc. of DIALM-POMC*, pp. 60–66, Oct. 2004.

[7] F. Dai and J. Wu, “Distributed dominant pruning in ad hoc networks.” *In Proc. of the IEEE International Conference on Communications*, vol. 1, pp. 353–357, May 2003.

[8] F. Kuhn, T. Moscibroda, and R. Wattenhofer, “Initializing newly deployed ad-hoc and sensor networks.” *In Proc. of MobiCom*, pp. 260–274, Sep./Oct. 2004.

[9] Z. Galil, “Efficient algorithms for finding maximum matching in graphs.” *ACM Computing Surveys*, vol. 18, no. 1, pp. 23–38, Mar. 1986.

[10] H. W. Kuhn, “The Hungarian method for the assignment problem.” *Naval Research Logistics Quaterly 2*, pp. 83–97, 1955.

[11] J. E. Hopcroft and R. M. Karp, “$N^{5/2}$ algorithm for maximum matchings in bipartite graphs.” *SIAM Journal on Computing*, vol. 2, pp. 225–231, 1973.