WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2008; **8**:645–659
Published online 29 January 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.492

Load balance and energy efficient data gathering in wireless sensor networks

Devendar Mandala[1], Xiaojiang Du[2*,†], Fei Dai[1] and Chao You[1]
[1]*Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58105*
[2]*Department of Computer Science, North Dakota State University, Fargo, ND 58105*

Summary

Many data-gathering protocols for wireless sensor networks use clustering technology for prolonging 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 network lifetime. We propose an even energy dissipation protocol (EEDP) for efficient cluster-based data gathering in wireless sensor networks. In EEDP, sensor data are forwarded to the base station (BS) 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.

KEY WORDS: data gathering; energy efficient routing; load balance; simulation ; wireless sensor network

1. Introduction

Recent advances in micro-electro-mechanical systems technology, wireless communications, radio frequency circuits, and digital electronics have resulted in the development of low-cost, low-power, multi-functional sensor nodes [1–3]. Each node has one or more sensors, embedded processors, and a transceiver and is normally battery operated. When these nodes coordinate with each other to perform a common task, they form a *wireless sensor network* (or simply sensor network). In a sensor network, nodes collect data from their sensors, process them, and forward the data to a BS or to a user, either directly or by relaying through neighboring nodes. Wireless sensor network is an emerging technology that has a wide range of potential applications including environmental monitoring, habitat monitoring, military surveillance, and so on [4–7].

A fundamental problem in wireless sensor networks is to maximize the network lifetime under given energy constraints. The network lifetime can be defined as the time elapsed until certain percentage of sensor nodes in the network completely deplete their energy. A sensor node is typically battery operated and therefore

*Correspondence to: Professor Xiaojiang Du, Department of Computer Science, North Dakota State University, Fargo, ND 58105, U.S.A.
†E-mail: Xiaojiang.Du@ndsu.edu

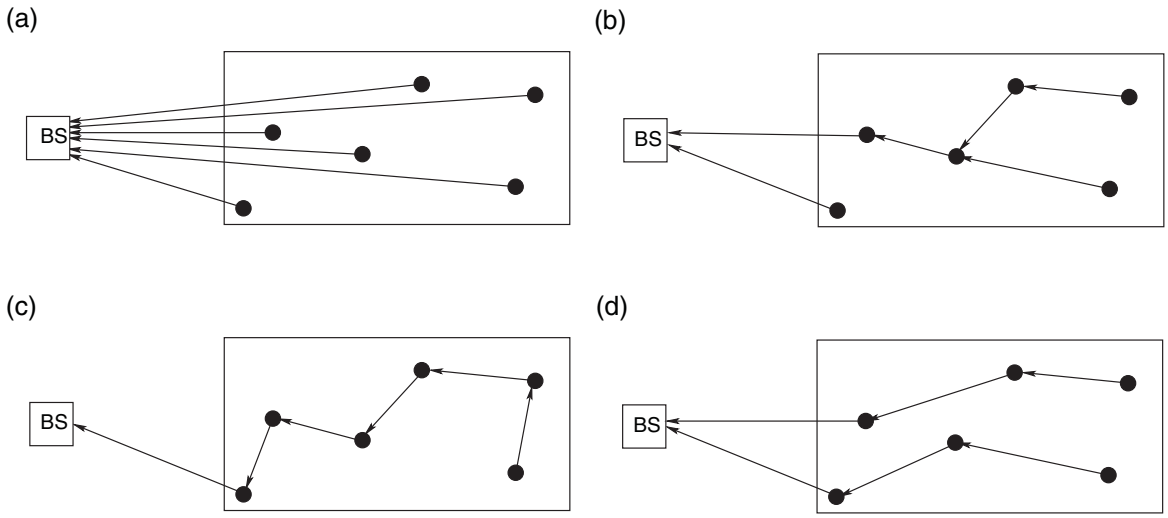

Fig. 1. Inter-cluster routing strategies; (a) Direct connection; (b) Shortest path routing; (c) Single chain; (d) Multiple chain.

constrained in energy. Thus, energy efficiency is a major concern in a wireless sensor network. To maximize the sensor nodes' lifetime, two different approaches can be used: increasing battery capacity or decreasing node energy consumption. Unfortunately, research in battery technology has not significantly increased battery capacity while restricting the battery weight and size. Our research goal is to use an energy saving method to decrease node energy consumption.

The energy of a sensor node is mainly consumed by its transceiver, computing component, and sensor, out of which the wireless transceiver uses a large portion of the energy. The major role of a sensor network is data gathering; that is, a large number of sensor nodes send their data to the base station (BS) . Several energy efficient protocols have been proposed [8–13] to route sensor data to the BS. Many of them take a cluster-based approach [8,9,11]: a few sensor nodes are elected as clusterheads to collect data from their neighboring nodes (called cluster members). The data traffic (as well as the data transmission and reception energy) can be greatly reduced by applying data aggregation [14–16] at clusterheads. Data aggregation in wireless sensor networks is the process of combining multiple data packets into one packet based on correlation in data. Cluster members have low energy consumption, as they transmit data to a nearby clusterhead. For better load balance, the role of clusterheads is rotated among cluster members. Previous simulation study shows that a cluster-based protocol can expand the network lifetime by eight times [8].

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 BS are direct connection [8] and shortest path routing [9]. In direct transmission, as shown in Figure 1(a), nodes far away from the BS dissipate their energy much faster than those close to the BS. In the shortest path routing, as shown in Figure 1(b), the small area close to the BS form a hot spot that relays sensor data for the entire network. In both cases, one portion of the network dies before others. Obviously, the network lifetime can be further improved by using a more balanced routing strategy.

We propose an energy efficient cluster-based data gathering protocol that balances energy consumption among different areas of the network. In the even energy dissipation protocol (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 BS. However, based on the intra-chain scheduling scheme, each node will occasionally skip its successors and transmit directly to the BS. This scheme balances energy consumption and avoids forming a hot spot. Note that a chain-based scheme [10], 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 [17].

To form balanced chains, the network is divided into *tiers* based on the distance to the BS. Each chain contains one clusterhead from each of these tiers. We developed two algorithms for a chain to select clusterheads

DOI: 10.1002/wcm

from each tier. 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 [8] and HEED [9] in terms of load balance and network lifetime.

The remainder of the paper is organized as follows: Section 2 reviews existing energy efficient data-gathering protocols and their limitations in uneven energy dissipation. Section-3 introduces EEDP and analyzes its performance in an ideal one-dimensional network, where chain formation is a trivial task. Section-4 extends this scheme to random two-dimensional networks and describes two chain formation algorithms. Section-5 shows the simulation results. Section-6 concludes this paper and discusses future work.

2. Related Work

Existing cluster-based data gathering protocols include LEACH [8], HEED [9], and their variations [11,18]. These 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 BS. Existing cluster election schemes include:

Random selection, which is used in LEACH. In this scheme, each node v becomes a clusterhead with a probability $p(v)$. The value of $p(v)$ depends on the expected number of clusterheads and previous election results. A node recently served as a clusterhead has a smaller $p(v)$. Random election is simple and 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 and EEUC [11]. Given a communication range r, a dominating set (DS) [19] is a group of nodes that covers the entire network; that is, every node in the network is either in the DS or a neighbor (i.e., within the distance r) to a node in the DS. Traditional DS formation algorithms [20,21] 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, with a penalty of slightly large DS size. This scheme builds higher quality clusters than random selection, which results in a longer network lifetime. The penalty is the higher election overhead due to information exchanges among neighbors.

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

2.1. Direct Connections

Where each clusterhead transmits aggregated sensor data directly to the BS, 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 BS 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.

2.2. Shortest Path Tree

In HEED, all clusterheads send aggregated data to the BS via the shortest path. These shortest paths form the shortest path tree (SPT), as shown in Figure 1(b). This scheme minimizes the total energy consumption. However, the energy consumption is still unbalanced. Note that the number of packets forwarded by each node depends on its position in the SPT. Nodes close to the root (i.e., the base station) have higher traffic load. Neighbors of the BS are responsible to relay all packets to the BS and have higher load. A hot spot is formed in the area surrounding the base station, which is congested with data traffic and consumes energy much faster than other areas of the network. When nodes in this area deplete their energy, not only does their energy depletion create an un-monitored spot, it will also disconnect other sensor nodes from the BS.

2.3. Single Chain

Although this scheme is used by PEGASIS [10], a non-cluster-based data gathering protocol, it can be used in a cluster-based scheme. As shown in Figure 1(c), a single chain is formed by connecting all the clusterheads. Each clusterhead communicates only with the closest neighbor and takes turns in transmitting data to the BS. Although a rotation scheme is used to share the cost of communication with the BS, uneven energy dissipation still exists due to the difference in clusterhead positions. In addition, this scheme assumes global data aggregation, that is, sensor data from all nodes can be aggregated into a single packet. This is

DOI: 10.1002/wcm

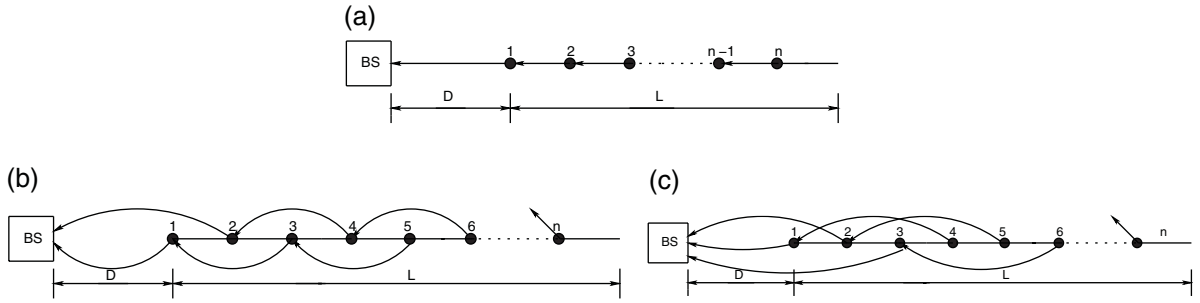

Fig. 2. Multiple chain formation in the one-dimensional network; (a)Single chain; (b) Two chains; (c) Three chains.

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.

In this paper, we propose a hybrid scheme that combines direct communication and multi-hop relaying. The objective is to avoid hot spots while maintaining low overall energy consumption. This scheme is similar to the single chain approach in that every node takes a turn to directly communicate with the BS. The difference is that our scheme does not assume global aggregation and avoids excessive relaying overhead by forming multiple chains.

3. Balanced Inter-Cluster Routing

We propose an EEDP for energy efficient and balanced inter-cluster routing. The basic idea is to form multiple chains of clusterheads as shown in Figure 1(d). Traffic load is evenly distributed among these chains to avoid a single hot spot. Each chain contains clusterheads far away from the BS as well as those close to the BS, 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 sensor network where all nodes are location aware and have the same initial energy and similar capabilities; sensor nodes are dispersed randomly in the sensing field and the BS location is fixed and outside of the sensor field. For the ease of discussion, we first consider 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 networks.

3.1. Intra-Chain 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 in this line, as shown in Figure 2(a). These clusterheads are static nodes. The distance between any two adjacent clusterheads is L/n. The BS is to the left of the sensing field. The distance between BS and node 1 is D and that between BS and node n is $D + (n-1)L/n$. 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, two, and three 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 networks without modification. In a chain with n nodes, data gathering 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 $r+1$. Since 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.

Figure 3 shows an example of intra-chain scheduling in a single chain. In the first round, as shown in Figure 3(a) clusterhead 1 forwards only one data packet to BS and the remaining $n-1$ data packets from clusterheads $2, 3, \ldots,$ n are relayed to BS by clusterhead 2. In the second round, as shown in Figure 3(b), clusterheads 1 and 3 forward 2 and $n-2$ data packets respectively to BS. In any round r, as shown in Figure 3(c), clusterheads 1 and $r+1$ forward

DOI: 10.1002/wcm

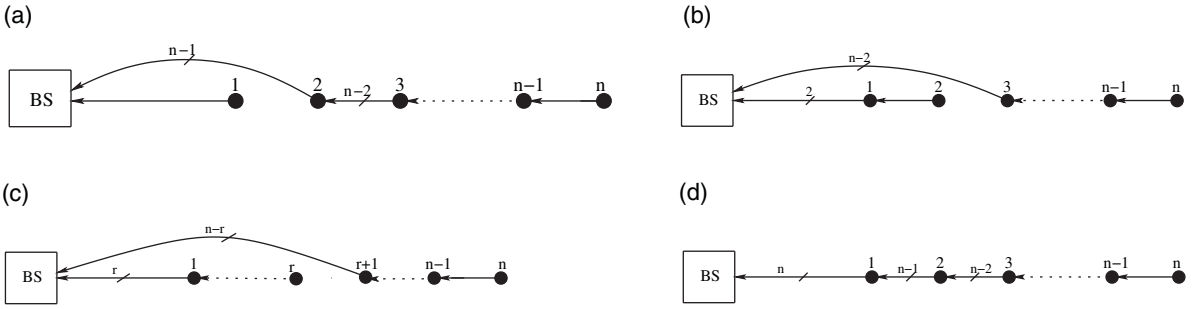

Fig. 3. Intra-chain scheduling. The label of each link represents the number of packets forwarded via this link. (a) Round 1; (b) Round 2; (c) Round r; (d) Round n.

Algorithm 1 Intra-Chain Scheduling

1: **for** each round $r \leftarrow 1$ **to** n **do**
2: **if** $r < n$ **then**
3: Route packets via two chains $(r, r-1, \ldots, 1, BS)$ and $(n, n-1, \ldots, r+1, BS)$.
4: **else**
5: Route packets via a single chain $(n, n-1, \ldots, 2, 1, BS)$.
6: **end if**
7: **end for**

r and $n - r$ data packets respectively to BS. In the last round n, as shown in Figure 3(d), there is no splitting of the chain and the clusterhead 1 forwards n data packets to BS. After n rounds, the scheduling scheme is repeated again starting from the first round.

When a single chain is used, each packet is relayed by $n/2 - 1$ nodes on average. For any L and D, the relaying overhead may overweigh the benefit of the scheduling scheme when n exceeds a certain threshold. In this case, a multiple chain structure is more attractive, where the average number of relays is under control. In the following subsection, we will discuss the optimal number of chains as a function of L, D, and n.

3.2. Performance Analysis

In this subsection, we compare the performance of EEDP with direct connection routing (LEACH) and shortest path routing (HEED) using the one-dimensional network model. Simulation results for two-dimensional networks are presented in Section 4. 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 these schemes.

We use the same energy model as considered in Reference [8]: To transmit a l-bit packet over a distance d, the corresponding transmission power E_{tx} and receiving power E_{rx} are

$$E_{tx}(d) = E_{elec}l + E_{amp}ld^2 \tag{1}$$

$$E_{rx} = E_{elec}l \tag{2}$$

where $E_{elec} = 50$ nJ/bit is the amount of energy the radio dissipates to run the transmitter or receiver circuitry and $E_{amp} = 100$ pJ/bit/m^2 is the amount of energy consumed by the transmit amplifier.

In LEACH, the energy consumed to deliver a packet at node k is

$$E^{dc}(k) = E_{tx}\left(D + \frac{k-1}{n}L\right) \tag{3}$$

and the maximum energy consumption is

$$E^{dc}_{Max} = E^{dc}(n) = E_{tx}\left(D + \frac{n-1}{n}L\right) \tag{4}$$

In HEED, each node k receives $n - k$ packets from the upstream node (farther away from the BS), and transmits $n - k + 1$ packets to the downstream node (closer to the BS). That is, the energy consumed to deliver a packet at node k is

$$E^{spt}(k) = \begin{cases} (n-1)E_{rx} + nE_{tx}(D) & : \ k = 1 \\ (n-k)E_{rx} + (n-k+1)E_{tx}(\frac{1}{n}L) & : \ k > 1 \end{cases} \tag{5}$$

Suppose $D > 1/nL$. The maximum energy consumption is

$$E^{spt}_{Max} = E^{spt}(1) = (n-1)E_{rx} + nE_{tx}(D) \tag{6}$$

Now we consider EEDP using a single chain. Since the energy consumption varies in each of the n rounds,

DOI: 10.1002/wcm

we calculate the average cost per round at a node k.

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

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

$$N_{\text{rx}}^{k,r} = \begin{cases} n-k & : \quad r \le k \\ r-k-1 & : \quad r > k \end{cases} \quad (8)$$

$$N_{\text{tx}}^{k,r} = \begin{cases} 0 & : \quad k = 1 \vee k = r \\ N_{\text{rx}}^{k,r} + 1 & : \quad \text{otherwise} \end{cases} \quad (9)$$

$$N_{\text{tx}'}^{k,r} = \begin{cases} 0 & : \quad k \neq 1 \wedge k \neq r \\ N_{\text{rx}}^{k,r} + 1 & : \quad \text{otherwise} \end{cases} \quad (10)$$

Combining Equations (7–10), we have

$$E^{\text{eedp}}(k) = \begin{cases} \frac{(n-1)}{2}E_{\text{rx}} + \frac{n+1}{2}E_{\text{tx}}(D) & : \quad k = 1 \\ \frac{(n-k)(n+k-1)}{2n}E_{\text{rx}} & : \\ +\frac{(n-k+1)(n+k-2)}{2n}E_{\text{tx}}(\frac{1}{n}L) & : \\ +\frac{n-k+1}{n}E_{\text{tx}}(D + \frac{k-1}{n}L) & : \quad k > 1 \end{cases} \quad (11)$$

and

$$E_{\text{Max}}^{\text{eedp}} = \max_{1\le k\le n} E^{\text{eedp}}(k) \quad (12)$$

For example, consider a one-dimensional network with $n = 4$, $L = 200$, $D = 100$, and $l = 2000$. As shown in Figure 4, the maximal and minimal energy consumption in LEACH are $12.6\,mJ$ (at node 4) and $2.1\,mJ$ (at node 1), respectively. Those for HEED are $8.7\,mJ$ (node 1) and $0.6\,mJ$ (node 4). In EEDP, the maximal single node energy consumption per round is $5.4\,mJ$ at node 1 and the minimal energy consumption is $3.6\,mJ$ at node 4. Obviously, EEDP achieves better balance of node energy consumption than the other two schemes. In this specific case, the network lifetime of EEDP is 60 per cent longer than that of HEED and more than twice that of LEACH.

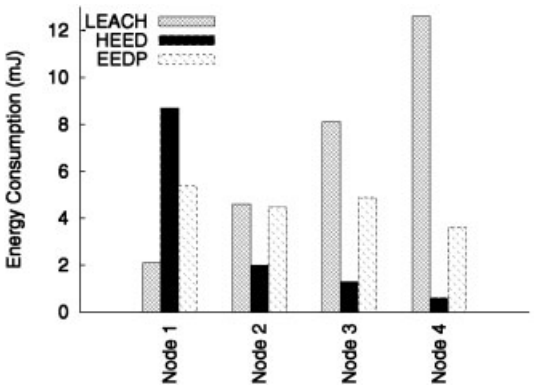

Fig. 4. Energy consumption distribution of 4 clusterheads in one-Dimensional network.

Next 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 chain formation scheme described in Section 3.1, 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

$$E^{\text{eedp}}(m,k) = \frac{1}{n'}\sum_{r=1}^{n'}[N_{\text{rx}}^{k',r}E_{\text{rx}} + N_{\text{tx}}^{k',r}E_{\text{tx}}(\frac{1}{n'}L) + N_{\text{tx}'}^{k',r}E_{\text{tx}}(D + \frac{k-1}{n}L)] \quad (13)$$

where $N_{\text{rx}}^{k',r}$, $N_{\text{tx}}^{k',r}$, and $N_{\text{tx}'}^{k',r}$ are defined by equations (8–10). Consequently,

$$E^{\text{eedp}}(m,k) = \begin{cases} \frac{(n'-1)}{2}E_{\text{rx}} + \frac{n'+1}{2}E_{\text{tx}}(D) & : \quad k = 1 \\ \frac{(n'-k')(n'+k'-1)}{2n'}E_{\text{rx}} & : \\ +\frac{(n'-k'+1)(n'+k'-2)}{2n'}E_{\text{tx}}(\frac{1}{n'}L) & : \\ +\frac{n'-k'+1}{n'}E_{\text{tx}}(D + \frac{k-1}{n}L) & : \quad k > 1 \end{cases} \quad (14)$$

and

$$E_{\text{Max}}^{\text{eedp}}(m) = \max_{1\le k\le n} E^{\text{eedp}}(m,k) \quad (15)$$

The performance of LEACH, HEED, and EEDP in one-dimensional network varies as the network parameters L, D and n varies. In order to compare the load balance of different 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 by varying one network

DOI: 10.1002/wcm

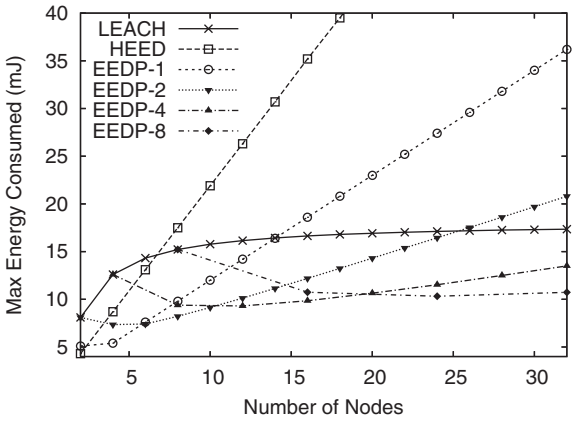

Fig. 5. Maximal single node energy consumption in one-dimensional network($L = 200$, $D = 100$, $l = 2000$).

parameter while keeping the other two parameters as constant. A scheme with lower maximal energy consumption value achieves better balance of node energy consumption and has a longer network lifetime.

In the one-dimensional network, we set $L = 200$, $D = 100$, $l = 2000$, and the number of clusterheads n varying from 2 to 32. For each n, there exists a best chain number m, such that the EEDP (using m chains) outperforms both LEACH and HEED routing schemes. The results are shown in Figure 5. As the value of n increases, we increase the number of chains in EEDP from 1 to 8. Figure 6 shows the situation when the distance D from the first node to the BS varies. As D increases, forming more chains becomes beneficial. Note that when $n = 8$, forming eight chains (EEDP-8) is identical to LEACH. Figure 7 shows the impact of the sensing area width L. It suggests that forming fewer chains is better under a larger L.

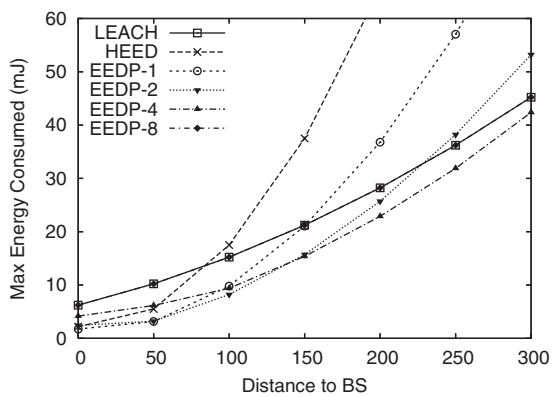

Fig. 6. Maximal single node energy consumption in one-dimensional network($L = 200$, $n = 8$, $l = 2000$).

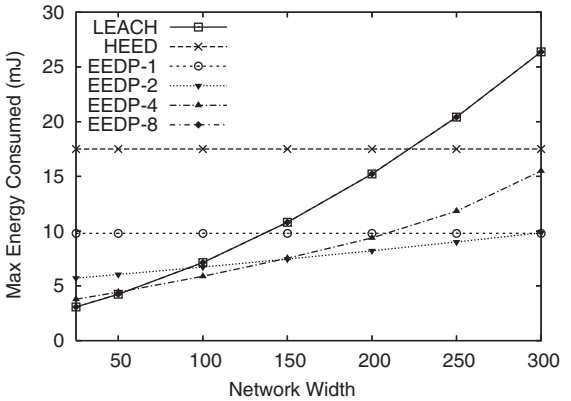

Fig. 7. Maximal single node energy consumption in one-dimensional network ($D = 100$, $n = 8$, $l = 2000$).

4. Tier-Based Multiple Chain Formation

When applying EEDP to 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. In this section, we propose two chain formation algorithms, both based on dividing the sensor network into vertical strips called *tiers*. The tiers formed are generally based on the distance of clusterheads to the BS. Clusterheads in the same tier have similar distances to the BS, as shown in Figure 8. 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. The second algorithm is optimal in the sense that it can guarantee minimal maximal distance between two consequent clusterheads in each chain. The performances of the algorithms are compared in a simulation study.

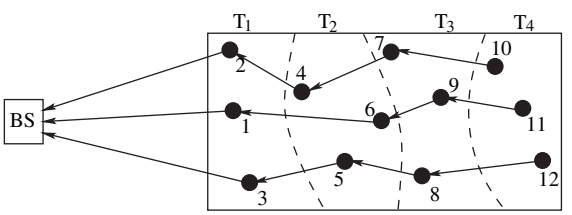

Fig. 8. Tiers and multiple chain formation.

DOI: 10.1002/wcm

4.1. 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 BS, and give them labels $1, 2, \ldots, n$, where node 1 is the closest to the BS, and node n is the farthest. When forming m chains, the network is divided into tiers $T_1, T_2, \ldots T_{\lceil n/m \rceil}$. The member of each tier is defined as follows.

Definition 1 (Tier). *The h-th tier of a sensor network is the set of clusterheads* $T_h = \{(h-1)m+1, (h-1)m+2, \ldots, hm\}$. *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 8 shows a sample network with $n = 12$ 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 $T_4 = \{10, 11, 12\}$. These nodes are called 1-hop, 2-hop, 3-hop, and 4-hop clusterheads, respectively.

Forming multiple chains is equivalent to the process of selecting a successor $\pi(i)$ for each non-1-hop clusterhead i ($1 < i \leq n$). After the successor selection, a clusterhead j is called a *chainhead* if it is not a successor of any other clusterhead, that is, $\nexists i : \pi(i) = j$. 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 $\pi^{(2)}(i) = \pi(\pi(i))$. The resultant chain corresponding to each chainhead is defined as follows.

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 8, 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, $Chain(10) = (10, 7, 4, 2)$. The rest of this section will discuss methods to select successors for non-1-hop clusterheads.

4.2. 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 to multiple nodes in the current tier (T_h), the node with the highest label (i.e., the one farthest to the BS) 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

1: Mark all clusterheads as unselected
2: **for** $i \leftarrow n$ **down to** $m+1$ **do**
3: $\quad h \leftarrow \lceil \frac{i}{m} \rceil$
4: $\quad \pi(i) \leftarrow$ an unselected clusterhead in T_{h-1} that is closest to i.
5: $\quad$ Mark $\pi(i)$ as selected.
6: **end for**

When applying Algorithm 2 to the sensor network in Figure 8, clusterhead 12, which is farthest from the BS, 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 nexthop, which form three chains: $Chain(10) = (10, 7, 4, 2)$, $Chain(11) = (11, 9, 6, 1)$, and $Chain(12) = (12, 8, 5, 3)$.

Although Algorithm 2 is easy to implement and has low ($O(mn)$) computation complexity, there are cases where it fails to form balanced chains. Here 'balanced chains' means that the maximal distance between a node and its successor is minimized. For example, consider the case when Algorithm 2 is applied to two adjacent tiers as shown in Figure 9. To form balanced chains as shown in Figure 9(a), 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 BS 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 9(b).

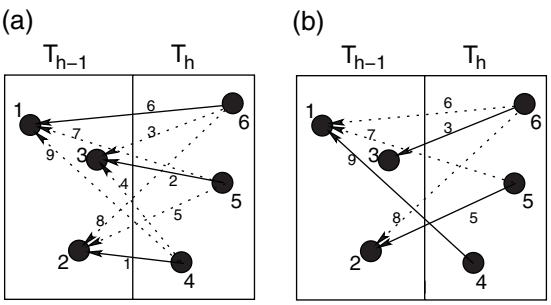

Fig. 9. Greedy and Balanced successor selection. The label associated with the link represents the rank based on ascending order of distances; (a) Balanced; (b) Greedy.

DOI: 10.1002/wcm

We present an algorithm that guarantees balanced successor selection in next subsection.

4.3. 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 [22]. 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 [23] or cardinality [24]. However, no existing method finds a perfect matching with minimal total cost or minimal maximum cost.

Algorithm 3 Balanced Matching

```
1: V ← T_h ∪ T_{h−1}
2: E ← {(i, j)|i ∈ T_h ∧ j ∈ T_{h−1}}
3: Sort E by edge weight (distance)
4: for k ← |T_h| to |E| do
5:    E_k ← the set of k minimal weight edges in E
6:    Compute a maximum cardinality matching M in
      bipartite graph G_k = (V, E_k)
7:    if |M| = |T_h| then
8:       π(i) ← j   :   ∀(i, j) ∈ M
9:       return
10:   end if
11: end for
```

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. The following theorem shows that Algorithm 3 guarantees minimal maximum distance.

Theorem 1. *Algorithm 3 selects a successor for each node $i \in T_h$, and minimizes the maximum distance between each node i and its successor $\pi(i)$.*

Proof. To prove the first part of the theorem, we need to show that the test in line 7 of Algorithm 3 succeeds at least once. This is true because, in the worst case, a maximal cardinality can be found to cover all nodes in T_h in the complete bipartite graph when $k = |T_h| \times |T_{h-1}|$. Note that $|T_h| = |T_{h-1}|$ for $h < \lceil \frac{n}{m} \rceil$ and $|T_h| \leq |T_{h-1}|$ for $h = \lceil \frac{n}{m} \rceil$.

We prove the minimal maximum property by contradiction. Suppose the first perfect match found by Algorithm 3 is $M = \{e_1, e_2, \ldots, e_{|T_h|}\}$ with the corresponding edge weights $d_1 \leq d_2 \leq \ldots d_{|T_h|}$. We further assume that M is found in step k, that is, $M \subseteq E_k$. If M is not a minimal maximum match, there exists another perfect match $M' = \{e'_1, e'_2, \ldots, e'_{|T_h|}\}$ with the corresponding edge weights $d'_1 \leq d'_2 \leq \ldots d'_{|T_h|}$ and $d'_{|T_h|} < d_{|T_h|}$. Therefore, there exists a $k' < k$ such that $M' \subseteq E_{k'}$. Since Algorithm 3 searches perfect matches in the ascending order of k, M' should be found before M, which contradicts the assumption that M is the first found perfect match. □

For example, when Algorithm 3 is applied to the network in Figure 9, E is the set of all the nine edges between the adjacent tiers which are numbered as 1 to 9 based on ascending order of distance, as shown in Figure 9(a). These edges are added in order. After six edges are added, a bipartite graph with perfect matching is found. A balanced successor assignment is shown in Figure 9(a).

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 has a complexity of $O(m^{5/2})$ [24]. 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.

5. Simulation

We compare the performance of EEDP, in terms of network lifetime, with two existing data gathering protocols, LEACH and HEED, via a simulation study. Simulation results confirm that EEDP significantly increases sensor network lifetime. The performances

DOI: 10.1002/wcm

of using different number of chains and different successor selection methods are also evaluated.

5.1. Implementation

All protocols are simulated via a custom simulator written in C++. The simulator generates random wireless sensor networks by scattering 600 nodes randomly and non-uniformly in a $100\,m \times 200\,m$ rectangular sensing area. The BS is located outside of the sensing area and is by default $100\,m$ from the left side of the rectangle that is, at (50,300). A DS formation algorithm, similar to the one used by HEED, is used to elect clusterheads, using a coverage radius (R) of $25\,m$ and $50\,m$, respectively.

In the beginning of each simulation, each node has an initial battery power of $1\,J$. During each round of simulation, each cluster member sends a packet to its clusterhead and the clusterhead uses an inter-cluster routing scheme to forward the aggregated data to the BS. The energy model discussed in Section 3.2 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: direct connection (LEACH), the shortest path routing (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 shortest node lifetime; that is, the number of rounds when the first node dies. Some sensor networks can continue functioning when a certain percentage of nodes die. We also measure the network lifetime when 50 per cent of nodes are dead and when all nodes are dead. All simulation results are means of 25 tries.

To form multiple chains after the clusters are formed, every clusterhead must send its ID to the BS. Based on the received signal strength, BS estimates the distance to clusterheads, assigns tier numbers to these clusterheads, and runs successor selection algorithm. BS then sends to each clusterhead its respective successor clusterhead ID to form multiple chains. As most of the work is done by BS, clusterheads have to just transmit and receive only one data packet for every round of clusterheads election. So, the additional energy consumption that is caused to establish multiple chains in EEDP is small. In the simulation study, we have not considered this energy consumption caused for communication to establish multiple chains.

Table I. Network lifetime using small clusters ($R = 25$, $m = 7$).

Number of rounds	First node death	50 per cent nodes dead	All nodes dead
LEACH	254	1820	3400
HEED	173	2210	3350
Greedy	620	2020	2740
Balanced	632	2030	2750

5.2. Results

In this subsection, we show simulation results under different settings, including inter-cluster routing strategy, the number of chains formed in EEDP, varying network width, varying network height, and the location of the BS, for small and large cluster sizes.

5.2.1. Routing strategy

The energy efficiency and load balance of all four routing strategies are compared in terms of the number of rounds for the first node, 50 per cent nodes and all nodes to die. Table I shows the results using small clusters ($R = 25$), where the number of chains (m) is seven. The value of m 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 per cent nodes dead, but their performance is close to HEED, which uses the shortest path routing strategy and 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.

Table II shows the results when large clusters ($R = 50$) and three chains are formed in EEDP. The

Table II. Network lifetime using large clusters ($R = 50$, $m = 3$).

Number of rounds	First node death	50 per cent nodes dead	All nodes dead
LEACH	450	2340	3500
HEED	656	2600	3480
Greedy	736	2420	3400
Balanced	710	2440	3420

DOI: 10.1002/wcm

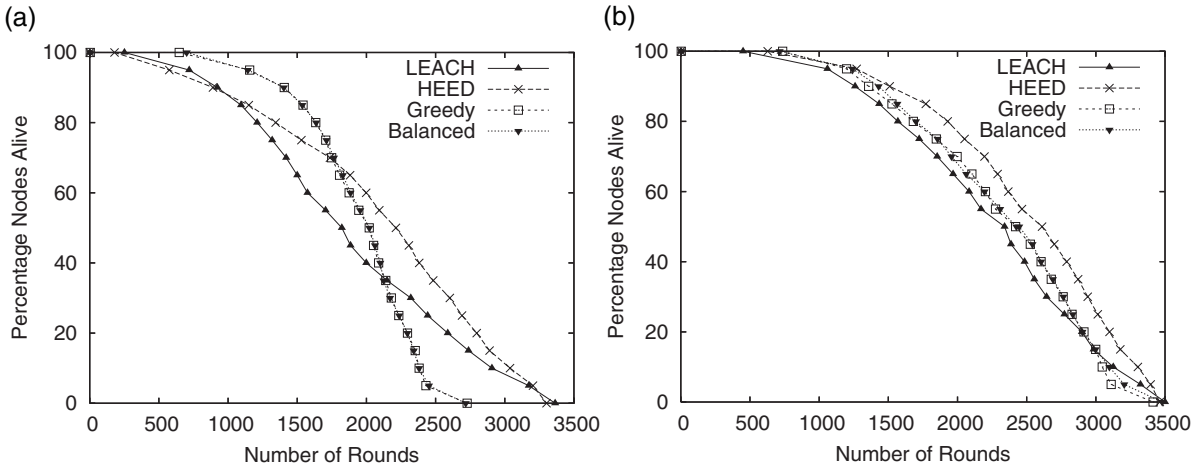

Fig. 10. Percentage of alive nodes; (a) $R = 25$; (b) $R = 50$.

relative performance is similar to that in Table I. As the coverage radius increased from 25 to 50, the average number of clusterheads formed decreased from 28 to 8. From Figure 5, we can observe that as the number of clusterheads decreases, the difference in the performance of all protocols becomes less obvious.

Figure 10(a) shows the percentage of nodes alive over the simulation time when coverage radius ($R = 25$) and seven chains ($m = 7$) are formed in EEDP. Both EEDP variations have longer network lifetime than LEACH and HEED. The EEDP variations show a more steep curve where 80 per cent nodes are dead between rounds 1500–2500. On the other hand, LEACH and HEED show a gradual decreasing curve between 800 and 3100 rounds where 80 per cent nodes are dead. Figure 10(a) demonstrates that node energy consumption in EEDP is more balanced.

Figure 10(b) shows the results when large clusters ($R = 50$) and three chains ($m = 3$) are formed in EEDP. The relative performance is similar to that in Figure 10(a), but the difference is less obvious because of the decrease in the number of clusterheads formed.

5.2.2. 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 11(a) shows the results when small clusters are formed ($R = 25$), 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 BS, as it needs to relay a large number of data packets to the BS. 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. Hence, we should select an optimum value for m.

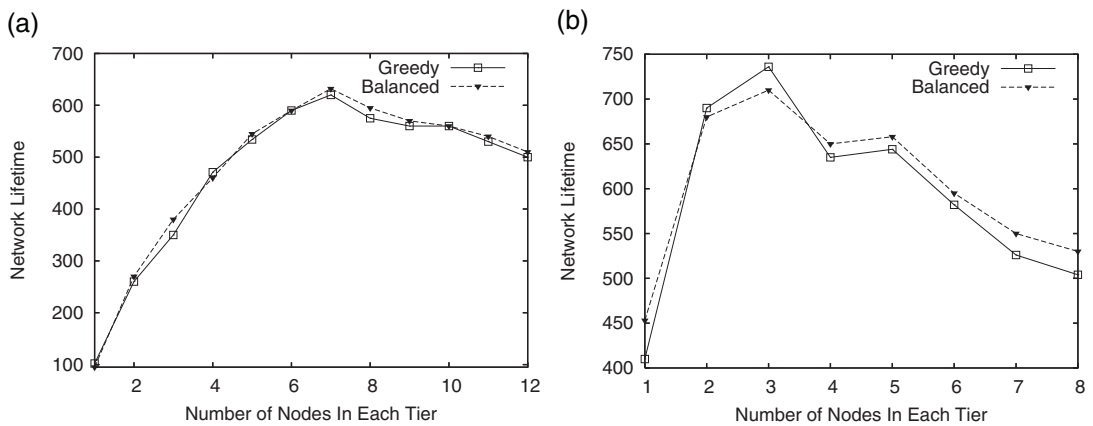

Fig. 11. Lifetime versus tier width; (a) $R = 25$; (b) $R = 50$.

DOI: 10.1002/wcm

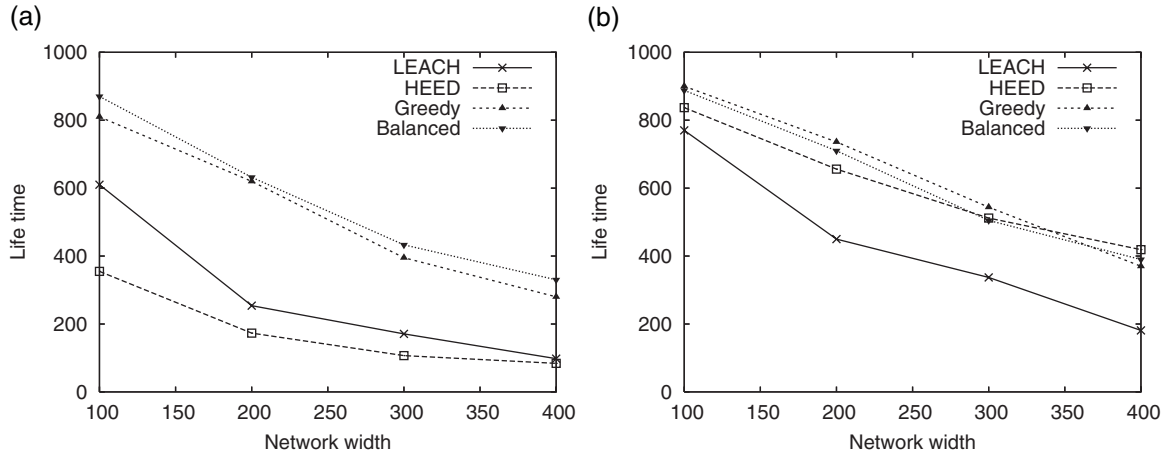

Fig. 12. Relationship between network width and network lifetime; (a) $R = 25$, $m = 7$ (b) $R = 50$, $m = 3$.

Figure 11(b) shows the results with large clusters ($R = 50$) with an average number of clusterheads $n = 10$. In this case, the network lifetime reaches its peak value when $m = 3$. Note that from the one-dimensional analysis in Section 3.2, forming eight and four chains has the highest performance when $n = 28$ and $n = 8$ respectively, which is quite close to simulation results.

5.2.3. Network width

We studied the effect of the network width on the network lifetime while maintaining the density of nodes as constant. From Figure 12(a) we can see, how the lifetimes for these routing schemes vary as the network width increases when small clusters ($R = 25$) and seven chains $m = 7$ are formed in EEDP. In case of LEACH, as network width increases, the distance between BS and the nodes that are far away from BS increases, thereby reduces the network lifetime. In case of HEED, as network width increases while maintaining the density of nodes as constant, the number of nodes along the width increases, which increases the load on the nodes that are close to BS, thereby reduces the network lifetime. Whereas, EEDP outperforms both LEACH and HEED.

Figure 12(b) shows the results when large clusters ($R = 50$) and three chains ($m = 3$) are formed in EEDP. The relative performance is similar to that in Figure 12(a), but the performance of HEED is close to EEDP because of decrease in the number of clusterheads formed.

5.2.4. Network height

We studied the effect of the network height on the network lifetime while maintaining the density of nodes as constant. From Figure 13 we can see, how the tier width of EEDP varies as the network height increases. Figure 14(a) shows how the lifetimes for these routing schemes vary as the network height increases for small clusters ($R = 25$). In case of LEACH, as network height increases, the distance between BS and the nodes that are far away from BS slightly increases, thereby reduces the network lifetime to a small extent. In case of HEED, as network width increases while maintaining the density of nodes as constant, the number of nodes increases, which increases the load on the nodes that are close to BS, thereby reduces the network lifetime. Whereas, EEDP outperforms both LEACH and HEED.

Figure 14(b) shows the results when large clusters ($R = 50$) formed in EEDP. The relative performance is similar to that in Figure 14(a), but the performance of HEED is close to EEDP because of decrease in the number of clusterheads formed.

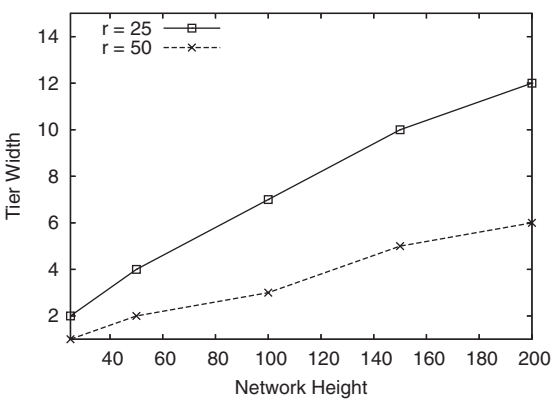

Fig. 13. Relationship between network height and tier width ($R = 25$ and $R = 50$).

DOI: 10.1002/wcm

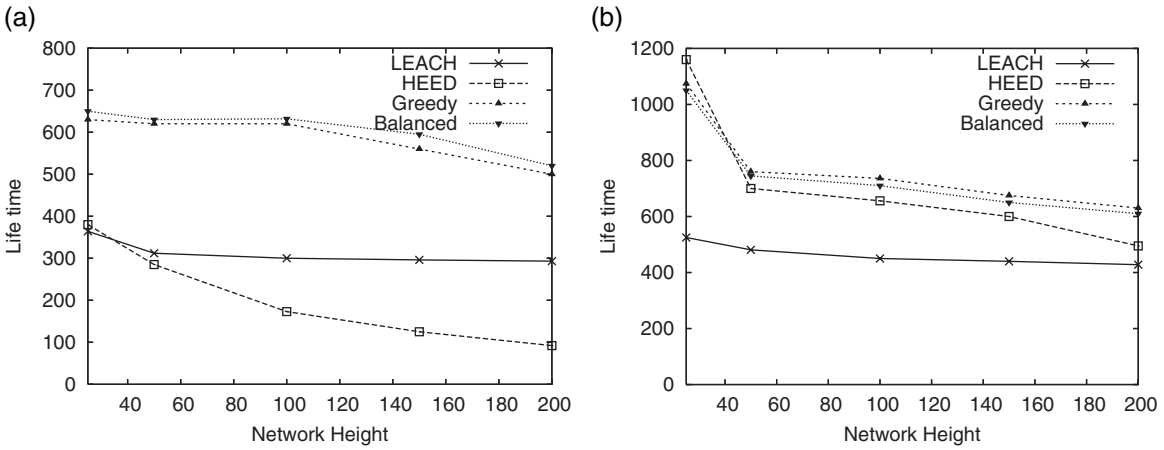

Fig. 14. Relationship between network height and network lifetime. (a) R = 25; (b) R = 50.

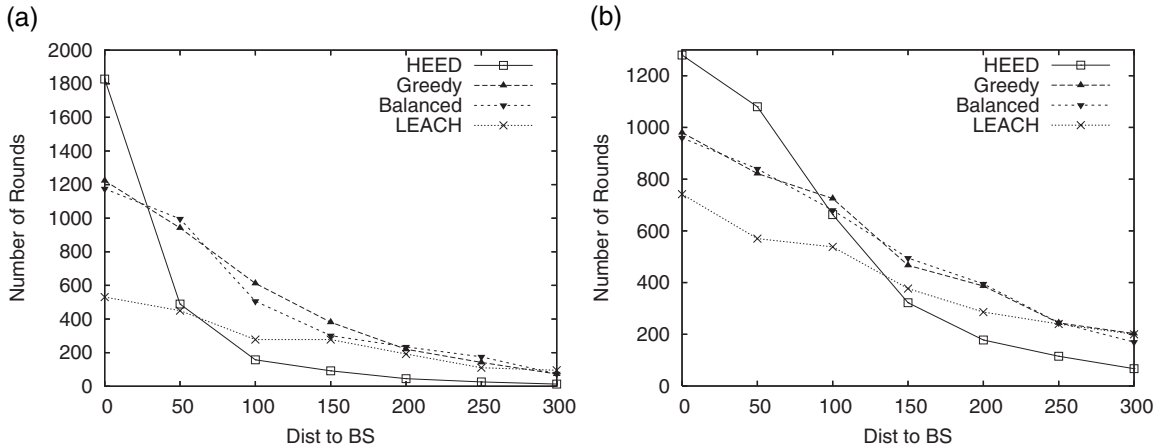

Fig. 15. Relationship between distance to BS and network lifetime. (a) R = 25, m = 7; (b) R = 50, m = 3.

5.2.5. Base station location

Finally, we studied the effect of the distance between BS and the network on network lifetime. From Figure 15 we can see, how the lifetimes for these routing schemes vary as the distance between BS and the sensor network increases. HEED shows an improvement over EEDP and LEACH when the BS is very close to the sensor network, as the clusterheads closer to BS have to transmit data packets to a smaller distance. As the distance increases slightly, EEDP outperforms both LEACH and HEED and as it increases to a very large value, EEDP performs close to LEACH.

The simulation results are summarized as follows:

1. The network lifetime under EEDP is significantly longer than that under LEACH and HEED when measured as the time that the first node dies, and in case of 50 per cent 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, that is, the balanced algorithm.
3. In EEDP, it is important to find the optimal number of chains that maximize the network lifetime.
4. EEDP outperforms both LEACH and HEED for various network widths.
5. EEDP outperforms both LEACH and HEED for various network heights.
6. HEED performs better when the BS is very close to the sensor network; as the distance slightly increases, EEDP performs better, and for a very large value of distance, the performance of EEDP is close to LEACH.

6. Conclusion

We have proposed a hybrid inter-cluster routing strategy for energy efficient and balanced data

DOI: 10.1002/wcm

gathering in wireless sensor networks. 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 BS. 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 early death of some nodes and increases overall network lifetime. Numeric analysis and simulation results confirm that EEDP outperforms two existing data gathering protocols—LEACH and HEED.

In the proposed scheme, each clusterhead communicates directly with the BS in every one of $\lceil n/m \rceil$ rounds. For better load balance, it might be more beneficial to assign different direct transmission rounds for different clusterheads in a chain. In the future work, we will develop an advanced scheduling scheme that allows each clusterhead to select heavy duty cycle based on the size of the sensing area, the distance to the BS, and its position in a chain. Another important parameter we plan to explore is the data aggregation ratio. In this paper we use a simple model where complete aggregation is achieved within each cluster and no aggregation is conducted between clusters. A piece of interesting work would be combining partial data aggregation and inter-cluster routing to further improve load balance and energy efficiency.

Acknowledgements

This work was supported in part by NSF EPSCoR grants EPS 0447679.

References

1. Chandrakasan A, Amirtarajah R, Cho S, *et al.* Design considerations for distributed microsensor systems. In *Proceedings of IEEE CICC*, pages 279–286, May 1999.
2. Clare L, Pottie G, Agre J. Self organizing distributed sensor networks. In *Proceedings of SPIE Conference on Unattended Ground Sensor Technologies and Applications*, pages 229–237, April 1999.
3. Dong M, Yung K, Kaiser W. Low power signal processing architectures for network microsensors. In *Proceedings of Internationl Symposium on Low Power Electronics and Design*, pages 173–177, August 1997.
4. Arora A, Dutta P, Bapat S, A line in the sand: a wireless sensor network for target detection, classification and tracking. *Computer Networks (Elsevier)* 2004; **46**(5): 1–15.
5. He T, Krishnamurthy S, Stankovic JA, *et al.* Energy efficient surveillance system using wireless sensor networks. In *Proceedings of ACM MobiSys*, pages 270–283, June 2004.
6. Mainwaring A, Culler D, polastre J, Szewczyk R, Anderson J. Wireless sensor networks for habitat monitoring. In *Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, September 2002.
7. Steere DC, Baptista A, McNamee D, Pu C, Walpole J. Research challenges in environmental observation and forecasting systems. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 292–299, 2000.
8. Heinzeman WR, Chandrakasan A, Balakrishnan H. Energy efficient communication protocol for wireless microsensor networks. *Proceedings of Hawaii International Conference on System Sciences*, January 2000.
9. Younis O, Fahmy S. Distributed clustering in ad hoc sensor networks: a hybrid, energy-efficient approach. *IEEE Transactions on Mobile Computing* 2004; **3**(4): 366–379.
10. Lindsey S, Raghavendra C, Sivalingam KM. Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel and Distributed Systems* 2002; **13**(9):924–935.
11. Li C, Ye M, Chen G, Wu J. An energy efficient unequal clustering mechanism for wireless sensor networks. In *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, November 2005.
12. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 56–67, August 2000.
13. Kulik J, Rabiner W, Balakrishnan H. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of MobiCom*, pages 174–185, August 1999.
14. Krishnamachari B, Estrin D, Wicker S. The impact of data aggregation in wireless sensor networks. In *Proceedings of International Workshop on Distributed Event-Based Systems*, pages 575–578, July 2002.
15. Erramilli V, Matta I, Bestavros A. On the interaction between data aggregation and topology control in wireless sensor networks. In *Proceedings of IEEE Communications Society Conference on Sensor and Ad-hoc Communications and Networks (SECON)*, pages 557–565, October 2004.
16. Krishnamachari B, Estrin D, Wicker S. Modelling data-centric routing in wireless sensor networks. In *Proceedings of IEEE Infocom 2002*, June 2002.
17. Rickenbach P, Wattenhofer R. Gathering correlated data in sensor networks. In *Proceedings of ACM DIALM-POMC*, pages 60–66, October 2004.
18. Choi W, Shah P, Das SK. A frame work for energy saving data gathering using two-phase clustering in wireless sensor networks. In *Proceedings of International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, pages 203–212, August 2004.
19. Dai F, Wu J. Distributed dominant pruning in ad hoc networks. In *Proceedings of the IEEE International Conference on Communications*, 1:353–357, May 2003.
20. Kuhn F, Moscibroda T, Wattenhofer R. Initializing newly deployed ad-hoc and sensor networks. In *Proceedings of MobiCom*, pages 260–274, Sep./Oct. 2004.
21. Lin CR, Gerla M. Adaptive clustering for mobile wireless networks. *IEEE Journal in Selected Areas in Communications* 1997; **15**(7):1265–1275.
22. Z Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 1986; **18**(1):23–38.
23. Kuhn HW. The Hungarian method for the assignment problem. *Naval Research Logistics Quaterly 2* 1995; **7**:83–97, 1955.
24. Hopcroft JE, Karp RM. $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 1973; **2**: 225–231.

DOI: 10.1002/wcm

Authors' Biographies

Devendar Reddy Mandala is a graduate student in the Department of Electrical and Computer Engineering at North Dakota State University (NDSU), Fargo, ND. Mr Devendar earned his B. Tech degree in Department of Electrical and Communication Engineering from Jawaharlal Nehru Technological University (JNTU), Hyderabad, India in 2005. He is currently an Research Assistant in ECE Department at NDSU. His research interests include wireless sensor networks, mobile ad-hoc networks, network security, computer networks and Internet routing. Mr Devendar has been a secretary of the NDSU cricket club since 2005.

Xiaojiang (James) Du is an Assistant Professor in the Department of Computer Science, North Dakota State University. Dr Du received his B.E. degree from Tsinghua University, Beijing, China in 1996, and his M.S. and Ph.D. from University of Maryland, College Park in 2002 and 2003, respectively, all in Electrical Engineering. His research interests are heterogeneous wireless sensor networks, security, wireless networks, computer networks, and network and systems management. Dr Du is an Associate Editor of Wireless Communication and Mobile Computing (Wiley) and International Journal of Sensor Networks (InderScience). He is (was) the Chair of Computer and Network Security Symposium of IEEE/ACM International Wireless Communication and Mobile Computing Conference 2007 (2006). Dr Du serves as a Vice-Chair of the Services and Applications Program of IEEE WCNC 2007. He is (was) a TPC member for many major IEEE conferences such as INFOCOM, ICC, GLOBECOM, WCNC, IM, NOMS, BroadNet, and IPCCC.

Fei Dai was an Assistant Professor in the Department of Electrical and Computer Engineering at North Dakota State University during 2005–2006. Dr Dai received his Ph.D. in the Department of Computer Science and Engineering at Florida Atlantic University and M.S. in the Department of Computer Science and Technology at Nanjing University, China. He has been a Senior Programmer in Greatwall Computer and a software architect and team leader in J and A Securities, both in China. He is now working for Microsoft, Redmond, WA. His research interests include networking, mobile computing, parallel and distributed computing, artificial intelligence, and software engineering.

Chao You received his B.S. degree in Physics from Nankai University, Tianjin, China, in 1999, and his M.S. and Ph.D. in Electrical Engineering from Rensselaer Polytechnic Institute, Troy, New York, in 2003 and 2005. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at North Dakota State University. His research interests are concentrated on integrated circuit design of high speed Field Programmable Gate Array (FPGA), VLSI, RFID, and energy harvesting.

DOI: 10.1002/wcm