Active node selection in flowing wireless sensor networks

Susumu Ishihara[†]and Daisuke Sato[‡]

[†]Graduate School of Science and Technology, Shizuoka University
3-5-1 Johoku, Naka-ku, Hamamatsu-Shi, Shizuoka, 432-8011, Japan, ishihara@ishilab.net
[‡]Faculty of Engineering, Shizuoka University
3-5-1 Johoku, Naka-ku, Hamamatsu-Shi, Shizuoka, 432-8011, Japan, sato@ishilab.net

Abstract

In this pape, we describe the concept of *flowing sensor networks* for monitoring drainage pipes or riverfront scenes with a group of flowing sensor nodes and discuss the design of algorithms for selecting active nodes to collect large amounts of sensing data with small battery-powered flowing sensor nodes. We assume that the chance of communicating with a sink node in the flowing sensor network is limited to the narrow areas around access points (APs) which are generally located on to the inside of the manholes of drainage pipes or at limited points along a riverfront. In order to backup measured data and to send the data to APs which the sensor nodes encounter, some nodes have to be ready to receive data from other nodes and to identify signals from APs so that sensor nodes can forward the data to the APs. We used simulation to investigate the performance for selecting active nodes of based on LEACH and HEED as well as that of am iproved HEED-based algorithm we developed. The proposed algorithm easily outperformed the other two.

Keywords: Wireless sensor networks, Energy saving, Clustering

1 Introduction

It takes a lot of time and human resources to monitor the condition of underground drainage pipes which are widely spread. Although using sensors to monitor such a condition is useful, it is difficult to place many sensors in drainage pipes, because the environment in drainage pipes is dangerous due to poisonous gasses and the risk of water accidents. Sensor nodes equipped with cameras, gas sensors, humidity sensors, etc. have shown promise as a way to reduce both the time and the labor to place and maintain many stationary sensors.

We have been investigating the feasibility of floating multiple sensor nodes down a water flow in a drainage pipe and collecting measured data from the nodes using access points (APs) attached to a limited number of manholes. Similar techniques can be used for monitoring the condition of rural riverfront areas and geographical features after major disasters such as floods, landslides, and earthquakes. We term such networks *flowing sensor networks.*

Figure 1 shows a typical overview of a *flowing sensor network*. We assume that sensor nodes can communicate with APs only when they are in the narrow areas near the APs due to the limited wireless communication range of sensor nodes and APs. We also assume that, in order to reserve its limited battery power, each node wakes up only when it measures values. To send the measured data to APs, each sensor node first has to establish a link with an AP and must therefore keep the communication interface active so that it can receive the signals from APs. It is problematic because keeping a wireless interface active wastes energy, so it would be more suitable to have some sensor nodes be always active and others work only intermittently. In such a strategy, a sensor node wakes up and measures a value, sends the data to one of its neighboring active nodes and then goes to sleep. When an *active node* receives a signal from an AP, it connects to the AP and forwards all the data which it has received from its neighboring sensor nodes.

The active nodes should be selected carefully so that the sensor nodes can work long enough and reliably enough to send the measured data to the APs. We must therefore select actives node on the basis of their residual energy and connectivity with other nodes.

In this paper, we describe the basic concept of a flowing sensor network, and discuss the design of algorithms for selecting active nodes that can collect a large amount of sensing data with small battery-powered flowing sensor nodes.

The rest of this paper is organized as follows. We discuss background and related work in Section 2. In Section 3, we introduce the basic concept and design of flowing sensor networks. The design of active node selection algorithms based on existing clustering algorithms designed for stationary sensor networks, LEACH [1] and HEED [2] is presented in Section 4. We also propose an improved HEED-based algorithm we developed. In section 5, we use simulation to compare the performance of the algorithms in terms of reliability of data collection. We conclude the paper in Section 6.

2 Related work

Zebranet[3] is the pioneer work of mobile sensor network. In this system, sensor nodes attached to wild animals autonomously form a network to transmit sensor data including the position of the animals. The protocol used in Zebranet does not support cooperative work (such as clustering) for saving energy; moreover, the mobility pattern of animals are assumed to be random. This system uses a mobile sink for collecting data from the wild animals: it is a system that uses mobile sensors and mobile sinks.

Systems that feature both mobile sinks and mobile sensors are not popular due to the difficulty of predicting a given

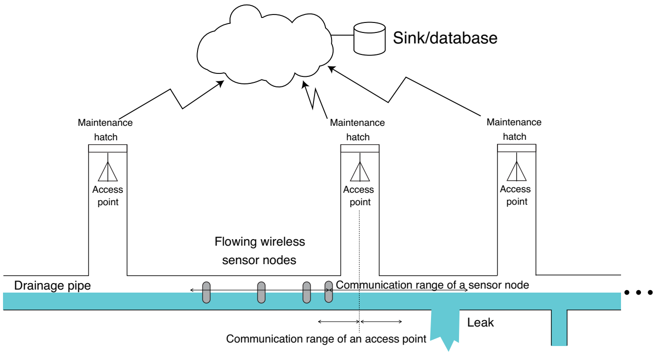


Figure 1: A typical flowing sensor network

node's mobility. In contrast, much research attention has been devoted to sensor networks using mobile sinks or relays [4][5]. In flowing sensor networks, sink nodes or access points do not move, but sensor nodes move and work in a small duty cycle. This makes it difficult to apply algorithms designed for sensor networks with mobile sinks to flowing sensor networks by directly replacing the relationship between sinks and sensor nodes.

Many clustering algorithms have been proposed for reducing the energy consumption of sensor networks. Most of them — LEACH [1], HEED[2], PEGASIS[6], DWEHC[7], etc. — are designed for static sensor networks. Most of the clustering algorithms for mobile nodes ()such as [8][9]) are designed for mobile ad hoc networks in which the intermittent behavior of sensor nodes is not considered. Though some clustering algorithms and topology control algorithms (such as [10][11]) are designed for saving energy have been proposed, they assume random mobility of nodes. In flowing sensor networks, the mobility pattern works to decrease the density of sensor nodes as time passes, and thus the algorithms for clustering or selecting active nodes should be designed in consideration of the characteristic mobility pattern.

There have been some recent research on the use of floating sensors. Members of the Floating Sensor Network Project at UC Berkeley are building a water monitoring system that can be deployed in estuarine environments and rivers[12][13]. They have developed drifters comprised of a GPS unit, Zigbee and GSM interfaces, and differential drive motors to move to a desired GPS point, etc. Kim et al. have proposed a drifting sensor for in-situ sewer gas monitoring, SewerSnort, and have shown that the position of a drifting sensor node in a drainage way can be estimated by using RSSI from access points[14]. However, to the best of our knowledge, algorithms for the scheduling or clustering of sensor nodes for reliable data delivery and longevity of the network in which sensor nodes have limited chances to connect to access points have not been proposed.

3 Flowing sensor network

The goal of a flowing sensor network is to uniformly collect a great variety of data (such as images of the wall of a water pipe, gas density, temperature, etc.) that are measured by small battery-powered sensor nodes with cameras, gas sensors, temperature sensors, etc. The group of sensor nodes is put into a waterway such as a drainage way or river and then transported along with the water flow. The sensor nodes do not have any controllable mobility functions, which makes it difficult to accurately predict the positions of the nodes after they are put into the waterway.

We assume that the sensor nodes have to send measured data to one of neighboring nodes and that the data are forwarded by the node to an AP when it can communicate with the AP. We explain this assumption as follows.

If the waterway is an underground drainage way, the positions in which APs can be installed are limited. Manholes are suitable locations for installing APs because it is easy to place and maintain APs there, and to connect them to the sink node of the network via wired or wireless links. Considering the convenience of maintaining the APs and the connectivity between the APs and the other side of the manholes, it is feasible to install APs near the covers of manholes, as shown in Fig. 1.

The distance between the manhole cover and the drainage pipe is about 3–10m and there are land masses and the walls of the manhole and the drainage pipe, it is not easy to make the communication range from the APs to sensor nodes floating in the waterway long. This makes the communication between APs and sensor nodes intermittent. However, the communication range between sensor nodes is broader than between APs and sensor nodes.

When a sensor node wants to communicate with an AP, it first has to find the AP. To receive the signal from the AP, the sensor node has to keep the communication interface active in order to receive the signal from the AP, which is problematic because it consumes so much of the limited energy in the battery. However, if we keep the communication interface of only a small portion of the sensor nodes active, we can reduce the total energy consumption.

We assume that active nodes receive data measured by their neighboring nodes and forward the data to APs when they can communicate with one of them. Non-active nodes, which are called *normal nodes* periodically wake up and measure the condition of their current position and then send the data to whichever active node they can communicate with. After that, they go back to sleep.

In the rest of the paper, we assume the following conditions.

- The number of sensor nodes that can measure and forward data may suddenly decrease due to waterway branches, water leakage, and node failure.

- The distance between sensor nodes gradually increases as they drift downstream due to the effect of the water current. This means that the number of one-hop neighbors per node decreases over time.

- Since GPS cannot be used underground, it is impossible for each sensor node to know its precise position. The sensor nodes can therefore not predict when they will find the next AP with any accuracy.

- The battery capacity on sensor nodes is limited, and wireless communication is the dominating factor of the energy consumption of the sensor nodes. If the communication interface is always on, the battery runs out before the sensor node reaches the end of the waterway.
- The system clocks of the sensor nodes are synchronized so that sleeping nodes can wake up at the same time and communicate with each other within a reasonable time margin. This can be achieved by synchronizing the clock of each node with APs when the node is active.

4 Active node selection

4.1 Objectives of active node selection

When selecting active nodes in flowing sensor networks, the following conflicting objectives need to be considered.

1. *Longevity*: Prolonging the system lifetime so that the sensor nodes can observe a wide area.
2. *Reliability*: Maintaining the connectivity between normal nodes and active nodes so that the measured data can be forwarded to APs quickly and reliably.

If the first objective is given priority, the system has to decrease the number of active nodes even if some normal nodes lose connectivity as a result. Thus the measured data may not be forwarded to an active node immediately. The normal nodes that cannot send data to active nodes may break down before they even get a chance to meet the active nodes.

On the other hand, if the system gives priority to the second objective, the number of active node in the systems will increase. This is because if the the distance between nodes increases due to the water flow, many active nodes are needed to guarantee the connectivity between normal nodes and active nodes. As a result, even if the active nodes are changed depending on the situation, sensor nodes in the system will consume their energy too quickly, thus shortening the system lifetime.

As the start point of the development of algorithms for selecting active nodes in flowing sensor networks, we investigated the performance of active node selection algorithms based on two famous clustering algorithms designed for static sensor networks: LEACH[1] and HEED[2]. The LEACH-based algorithm prioritizes the first objective while the HEED-based algorithm prioritizes the second one. We also designed an additional algorithm, which is a modification of the HEED-based algorithm, that can balance the two objectives.

4.2 LEACH-based algorithm

4.2.1 LEACH

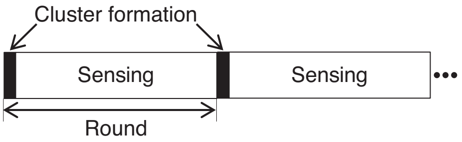


Figure 2: Structure of a round of LEACH and HEED.

Low Energy Adaptive Clustering Hierarchy (LEACH) is one of the most popular clustering algorithms for WSNs [1]. It forms clusters of sensor nodes on the basis of received signal strength and uses the cluster head (CH) nodes as routers to the base station. Note that in this paper the "CH" in LEACH and "active node" have the same meaning.

LEACH periodically forms clusters by using a distributed algorithm, in which nodes make autonomous decisions without any centralized control. A cluster is formed at the beginning of each round, which consists of a cluster-forming period and a sensing period, as shown in Fig. 2. Initially, a node decides to be a CH with a probability p and broadcasts its decision. Each non-CH node (normal node) determines its cluster by choosing the CH that can be reached using the least communication energy — in other words, the closest one. The CH role is rotated periodically among the nodes of the cluster in order to balance the load. The rotation is performed by getting each node to choose a random number T between 0 and 1. A node becomes a CH for the current rotation round if the number is less than the following threshold:

$$T(i) = \begin{cases} \frac{p}{1-p(r \bmod (1/P))} & \text{if } i \in G \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where p is the desired percentage of CH nodes in the sensor population, r is the current round number, and G is the set of nodes that have not been CHs in the last $1/p$ rounds.

4.2.2 LEACH-based active node selection in flowing sensor networks

Although LEACH assumes that all sensor nodes can communicate with a base station, it is unrealistic to assume that all flowing sensor nodes in a pipe or river can communicate with all APs. We assume that the communication range of the sensor nodes is limited. Thus, if the distance between a normal node and its closest active node is great, the sensor node cannot communicate with the active node. This prevents the sensor node from sending measured data.

4.3 HEED-based algorithm

Since LEACH selects cluster heads on the basis of given probability and the history of each node being a CH, the positions of CHs, the distribution of the size of clusters, and the distribution of the number of clusters in each round may be unbalanced. This unbalance causes the concentration of energy consumption on some nodes. Therefore a large number of algorithms have been proposed to improve LEACH, including HEED[2], PEGASIS[6], etc.

Hybrid Energy-Efficient Distributed Clustering (HEED) [2] is a distributed clustering scheme in which CH nodes are picked from among the deployed sensors. HEED considers a hybrid of energy and communication cost when selecting CHs. Unlike LEACH, it does not select CH nodes randomly, only sensor nodes that have a high residual energy can become CH nodes. HEED has three main characteristics:

- The probability of two nodes within their transmission range becoming CHs is small. This means that, unlike LEACH, CHs are well distributed in the network.
- Energy consumption is not assumed to be uniform for all the nodes.
- For a given sensor's transmission range, the probability of CH selection can be adjusted to ensure inter-CH connectivity.

In HEED, each node is mapped to exactly one cluster and can directly communicate with its CH. The algorithm is divided into three phases:

1. *Initialization phase:* The algorithm first sets an initial percentage of CHs among all sensors. This percentage value, C_{probe}, is used to limit the initial CH announcements to the other sensors. Each sensor sets its probability of becoming a cluster head, CH_{prob}, as follows:

   $$CH_{\text{prob}} = C_{\text{prob}} \cdot E_{\text{residual}}/E_{\max}, \qquad (2)$$

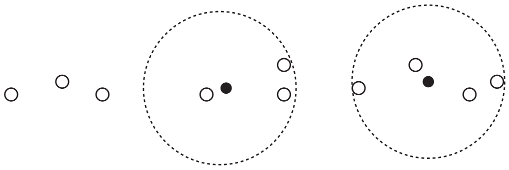
   where E_{residual} is the current energy in the sensor and $E_{\max}$ is the maximum energy (which corresponds to a fully charged battery). CH_{prob} is not allowed to fall below a certain threshold $p_{\min}$, which is selected to be inversely proportional to $E_{\max}$.

2. *Repetition phase:* During this phase, every sensor node goes through several iterations until it finds the CH that it can transmit to with the least *cost*. The *cost* can be the density of nodes, the distance between other nodes or the residual energy of the node. If it hears from no CH, the node elects itself to be a CH and sends an announcement message to its neighbors informing them about the change of status. Finally, each sensor doubles its CH_{prob} value and goes to the next iteration of this phase. It stops executing this phase when its CH_{prob} reaches one. There are two types of CH status that a sensor could announce to its neighbors:

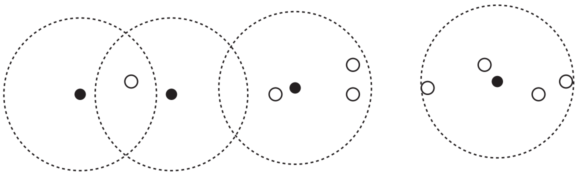
   - *Tentative status:* The sensor becomes a tentative CH if its CH_{prob} is less than 1. It can change its status to a regular node at a later iteration if it finds a lower-cost CH.
   - *Final status:* The sensor permanently becomes a CH if its CH_{prob} has reached 1.

   The number of iteration for obtaining *Final* CHs, N_{iter} does not depend on the number of nodes in the network, $N_{\text{iter}} = O(1)$.

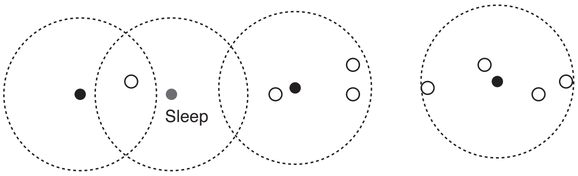
3. *Finalization phase:* During this phase, each sensor makes a final decision on its status. It either picks the least-cost CH or declares itself a CH.



(a) LEACH-based algorithm: Some nodes may lose connectivity to active nodes.



(b) HEED-based algorithm: Every node has connectivity to at least one active node.



(c) Improved HEED-based algorithm: Active nodes go to sleep at probability calculated on the basis of the node density.

Figure 3: Difference between three active node selection algorithms

4.3.1 HEED-based active node selection in flowing sensor networks

In the HEED-based algorithm, nodes select active nodes so that at least one active nodes (= CH) exists in their one-hop communication area in consideration of their residual energy at the beginning of each round.

HEED uses messages sent periodically from each node, heart beat messages (HBMs), to calculate the costs for choosing CHs. If a node receives an HBM, it saves the sender's ID, the signal strength, geographical position, and any other relevant data to its neighbor list. Each node then uses this information to calculate the density of nodes, distances between neighboring nodes, etc. to use as a final cost.

HEED is designed to be used in stationary sensor networks, and thus the network topology rarely changes. The cost value can only be changed due to battery exhaustion, a decrease in the number of nodes due to node failure, and the deployment of new nodes. This means that, after the initial HBM exchange at the beginning of the network, nodes do not have to send HBMs very frequently.

Flowing sensor networks of course are different in that they assume that nodes always move and the topology of the network frequently changes. Thus nodes have to send HBMs frequently to calculate the *cost*. However, the frequent transmission of HBMs wastes node energy. In this paper, we assume that nodes do not send HBMs and use the ratio of residual energy and the initial energy as the *cost*.

4.4 Improved HEED-based algorithm

The HEED-based algorithm selects active nodes so that every normal node can communicate with at least one active node directly. Thus the number of active nodes increases as the sensor nodes spread widely when drifting downstream. When the number of neighboring nodes is small, it is especially important that each node frequently becomes an active node. This is problematic because it consumes battery energy very quickly. To avoid such a condition, we developed an improved HEED-based algorithm.

Our improved HEED-based algorithm allows each active node to go to sleep with a probability calculated on the basis of the number of neighboring nodes. In this paper we assume the probability p_{sleep} of an active node is calculated as follows.

$$p_{\mathrm{sleep}} = 1 - \min(\alpha N_{\mathrm{neighbors}}/N_{\mathrm{all}}, 1) \qquad (3)$$

where $N_{\mathrm{neighbors}}$ is the number of neighbors of the active node, N_{all} is the number of all sensor nodes that are placed in the waterway at the same time, and α is a positive real number. Active nodes periodically calculate p_{sleep} and go to sleep according to the value until the next timing for calculating p_{sleep}.

Thus the connectivity between normal nodes and active nodes and between active nodes and APs decreases as the node density becomes small. However, the nodes — especially those with a small number of neighbors — save energy by rarely becoming active, which means they can survive longer and the system can collect more data from a wide area. Figure 3 shows a comparison of the LEACH, HEED, and improved HEED-based algorithms.

5 Simulation

5.1 Simulation Model

To evaluate the performance of the three algorithms in flowing sensor networks, we constructed a cellular automaton-based node mobility model (Fig. 4). The model is designed so that it can present partitions of sensor node groups due to the random behavior of water flow. A waterway is presented as 5 × 200 cells. The size of each cell is 10 × 10 m. In the model, only one node can exist in a cell and nodes move to the closest cell to the right if there is no node with probability p_m. If p_m is 1.0, all nodes move to the right cell simultaneously. If p_m is 0.5, half of the nodes try to move right and the rest remain stationary. Thus, the degree of node dispersion is highest when $p_m = 0.5$. The decision about node movement is made from the right-end nodes to the left. At the beginning of the simulations, two nodes existed at the two left-end columns of each line. APs are located every 20 cells. An active node can only send data to an AP if it exists in the same column as the AP. Each normal node can send data to the closest active node provided the horizontal distance between them is less than 6 cells. The vertical distance is ignored.

At each time step (= 10 seconds), all nodes, including active nodes, measure the sensor value of their current cell and send the data to the closest active node if there is one in their communication range. Each normal node discards the data item after sending it to an active node. Each active node stores the received data items until it encounters an AP, to which it then forwards the data. If an active node can communicate with an AP, it sends the stored data items to the AP and then discards them. After these operations, the positions of all nodes are updated using the mobility model. In the improved HEED-algorithm, all active nodes calculate p_{sleep} at the beginning of each time step. Each active node sleeps according to p_{sleep} value during the time step. When it is sleeping, it does not receive data from neighboring normal nodes.

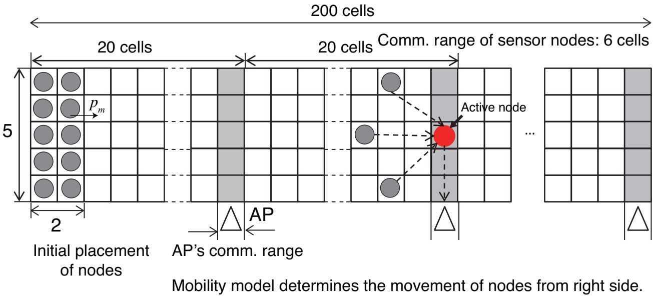


Figure 4: Simulation model

Table 1: Simulation Parameters.

	Parameter	Value
Mobility Model	Round length: N	5 steps
	p_m	0.7–1.0
	p_e	0, 0.001
Energy model	E_{elec}	50nJ/bit
	E_{amp}	10pJ/bit/m^2
	E_{i}	95.1mJ/s
	d_0	75m
	Initial energy	15J

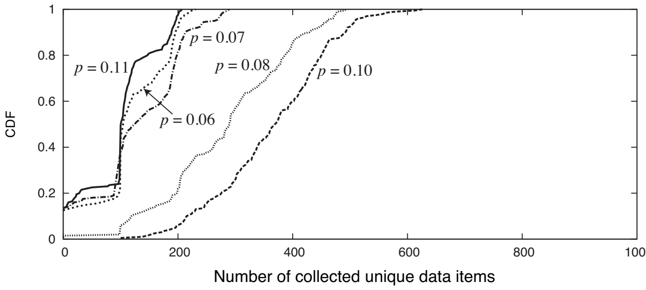
The length of each round is five time steps. At the beginning of each round, active nodes are selected using one of the three algorithms. If an active node that has stored data items becomes a normal node, it forwards the data items to a new active node and then discards the items. Each node can hold up to 100 data items. If a node obtains a new data item when it already has 100, it chooses one of the original items to discard.

We used an energy consumption model similar to a model presented in [2] (Eq. (4)(5)) and typical energy consumption of MicaZ [15] (Eq. (6)). In the following equations, d is the distance between nodes and n_b is the packet length. The values of E_{elec}, E_{amp}, E_{idle}, and d_0 are shown in Table 1.

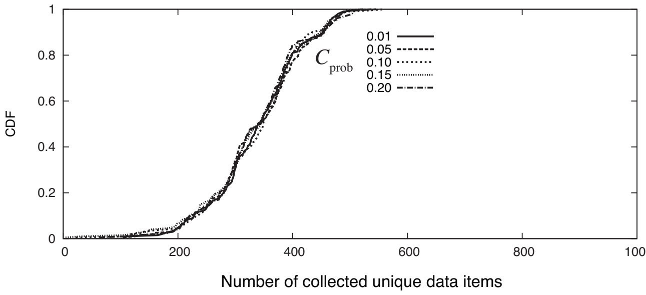
$$\text{Send} : E_{\mathrm{T}} = \begin{cases} n_{\mathrm{b}}(E_{\mathrm{elec}} + E_{\mathrm{amp}} \cdot d^2) & \text{if } d < d_0 \\ n_{\mathrm{b}}(E_{\mathrm{elec}} + E_{\mathrm{amp}} \cdot d^4) & \text{if } d \geq d_0 \end{cases} \qquad (4)$$

$$\text{Receive} : E_{\mathrm{R}} = n_{\mathrm{b}} \cdot E_{\mathrm{elec}} \qquad (5)$$

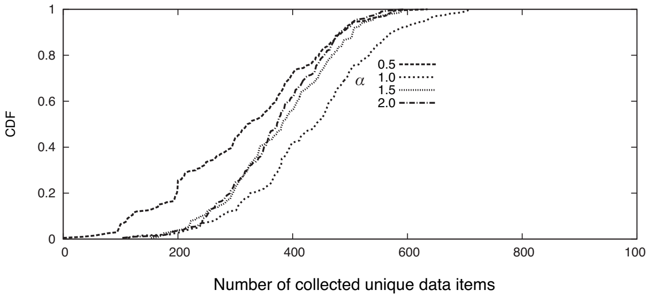
$$\text{Idle Listening} : E_I = E_{\mathrm{elec}} \cdot t \qquad (6)$$



(a) LEACH-based algorithm



(b) HEED-based algorithm



(c) Improved HEED-based algorithm

Figure 5: Effect of parameters of LEACH and HEED on the number of collected unique data items

In this simulation model, packet transmissions for forming clusters and data are considered. We assume that the size of all control packets for forming LEACH and HEED cluster is 50 bytes and the data size per data item measured at each time step is 50 bytes. We do not consider the effect of the size of the packet header or MAC operation.

Each node breaks down at each round with probability p_e. Once a node breaks down or has consumed all its energy, it does not communicate with other nodes.

We performed a preliminary simulation to determine the optimal performance of each algorithm.We tested p for LEACH in Eq. (1), C_{prob} for HEED and improved-HEED in Eq. (2), and α for improved-HEED in Eq. (3) when $p_m = 1.0$. We obtained the distribution of the number of unique data items of each cell. Duplicated data items obtained by different sensor nodes are treated as one item.

Figure 5 (a), (b), and (c) show the results of the preliminary simulation. The graphs show the cumulative distribution function (CDF) of the number of collected unique data items obtained by 100 simulation runs of each parameter. These results promoted us to use $p = 0.1$, $C_{\text{prob}} = 0.2$, and $\alpha = 1.0$.

5.2 Simulation Results

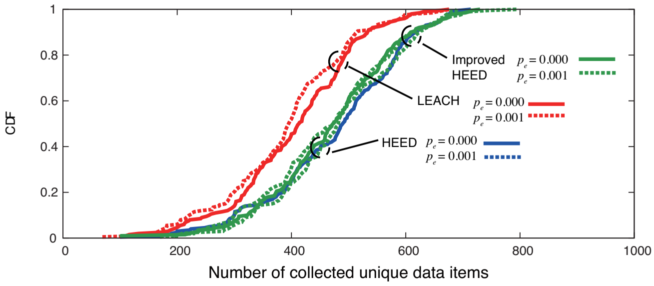
Figures 6(a)–(e) show the CDF of the number of collected unique data items. These results were obtained by 100 simulation runs for each of the scenarios with various p_m and p_e. We used the same parameters for the LEACH and HEED operations that were used in the previous subsection. Note that the number of collected unique data items implicitly presents the similar characteristic presented by network life time which is often used for the performance evaluation of stationary wireless sensor networks. This is because, if many nodes have exhausted their battery and lost the connectivity to active nodes, APs or sink nodes, only small amount of data can be collected in both stationary sensor networks and flowing sensor networks.

For all p_m values, our improved HEED-based algorithm was able to collect data from the widest area. The effect of node failure was small for all algorithms.

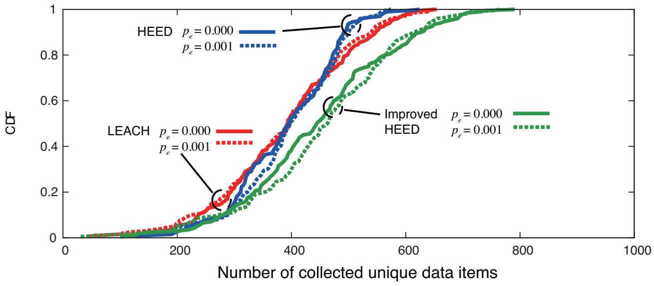
When all nodes moved in a group, $p_m = 1.0$, both the HEED and the improved HEED-based algorithms outperformed LEACH-based algorithm. This is because the selection of active nodes in the LEACH-based algorithm depends on fixed probability. Assuming N_{all} is the number of nodes, all nodes are not active at probability $(1-p)^{N_{\text{all}}}$ in the LEACH-based algorithm. Thus, normal nodes may lose the chance to forward data to an active node even if all nodes move as a group. Adding to this, even if all nodes are in cells where they can communicate with an AP, none of the nodes are active at the probability. Furthermore, multiple nodes may become active simultaneously and consume excessive energy even though the network might not need more than one active node. In contrast, only one node is active when the HEED-based algorithm is used when $p_m = 1.0$. Consequently, measured data can be forwarded to APs reliably and the sensor nodes can save their energy.

When p_m became less than 1.0 and approaches 0.5, the positions of sensor nodes tend to disperse. As p_m approached 0.5, the performance of HEED worsened. This is because, in the HEED-based algorithm, active nodes are selected so that every normal node can communicate with at least one active node. Therefore, as the sensor nodes disperse, more active nodes exist in the system, energy is consumed more quickly and the nodes exhaust their energy before they reach the end of the waterway. In contrast, the LEACH-based algorithm is less sensitive to the dispersion of nodes because the number of active nodes is not affected by the connectivity between nodes. However, this algorithm does not guarantee that every normal node can communicate with an active node, and measured data by a normal node cannot be sent to an AP immediately after it obtains the data if there are no active neighbor nodes. Note that measured data items can be stored locally at each node. Thus, even if the node cannot send the measured data item immediately, it can always forward the data item to an active node in the future.

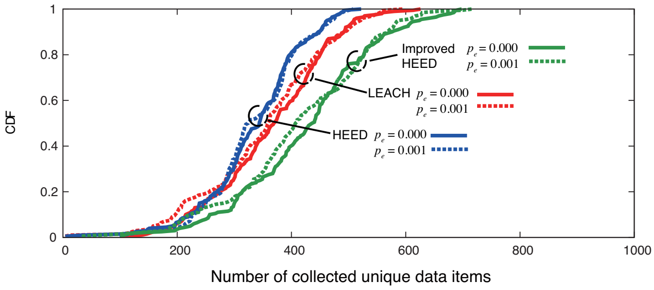
The improved HEED-baaed algorithm outperformed both



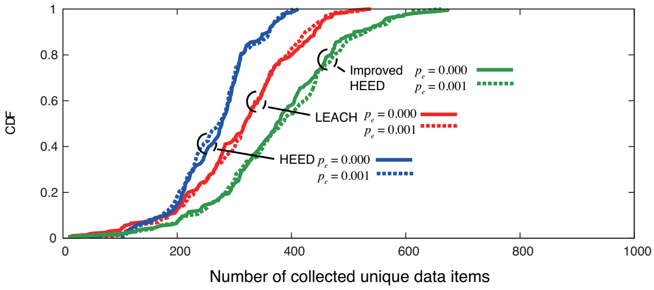
(a) $p_m = 1.0$



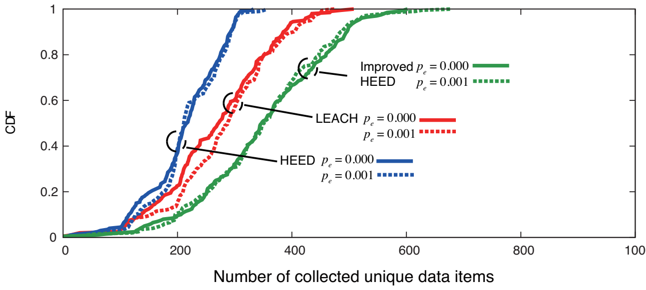
(b) $p_m = 0.95$



(c) $p_m = 0.9$



(d) $p_m = 0.8$



(e) $p_m = 0.7$

Figure 6: Effect of p_m on the number of collected unique data items

the LEACH-based and HEED-based algorithms. This is because it balances the energy consumption by multiple active nodes and the connectivity between normal nodes and active nodes by allowing active nodes go to sleep depending on the number of neighboring nodes.

6 Conclusions

We described the concept of flowing sensor networks and evaluated two active node selection algorithms for such networks, LEACH-based algorithm and HEED-based algorithm, that are based on existing clustering algorithms originally designed for static sensor networks. We also proposed an improved HEED-based algorithm for overcoming the weaknesses of the LEACH-based algorithm and HEED-based one. We conducted a simulation of these algorithms using a simplified node mobility and communication model. Results demonstrate the importance of balancing the connectivity of active nodes with other nodes, which is the priority with HEED, and the total number of active nodes in the network, which is priority with LEACH. Our improved HEED-based algorithm balances these two factors and outperforms the other two algorithms. In our future work, we need to perform a detailed simulation considering real node movements in waterways and wireless communication networks.

REFERENCES

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks. Proc. 33rd Annual Hawaii International Conference on System Sciences, pp. 1–10 (2000).

[2] O. Younis and S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Transactions on Mobile Computing, Vol. 3, No. 4, pp. 366–379 (2004).

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein, Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet, Proc. 10th Int'l Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), pp.96–107 (2002).

[4] C. Chen, J. Ma, and K. Yu, Designing energy-efficient wireless sensor networks with mobile sinks, Proc. WSW'06 at Sensys'06 (2006).

[5] W. Wang, V.Srinivasan, and K-C. Chua, Using mobile relays to prolong the lifetime of wireless sensor networks, Proc. ACM MobiCom 2005, pp.270–283 (2005).

[6] S. Lindsey, C.S. Raghavendra, PEGASIS: Power efficient gathering in sensor information systems, Proc. IEEE Aerospace Conference (2002).

[7] P. Ding, J. Holliday, A. Celik, Distributed energy efficient hierarchical clustering for wireless sensor networks, Proc. IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05) (2005).

[8] D.J. Baker, A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, IEEE Trans. Communications, Vol. 29, No. 11, pp. 1694–1701 (1981).

[9] K. Xu, M. Gerla, A heterogeneous routing protocol based on a new stable clustering scheme, Proc. IEEE Military Communications Conference (MILCOM 2002), (2002).

[10] J.Y. Yu and P.H.J. Chong, An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs), Computer Communications, Vol. 30, pp. 5–16 (2006).

[11] J. Li and S. Jiang, A scalable clustering algorithm in dense mobile sensor networks, J. of networks, Vol. 6, No. 3, pp. 505–512 (2011).

[12] Floating Sensor Network project, UC Berkeley, http://lagrange.ce.berkeley.edu/fsn/

[13] A. Tinka, I. Strub, Q. Wu, and A. Baye, Quadratic programming based data assimilation with Passive drifting sensors for shallow water flows, Proc. 48th IEEE Conference on Decision and Control, pp. 1686–1700 (2010).

[14] J. Kim, J.S. Lim, J. Friedman, U. Lee, L. Vieira, D. Rosso, M. Gerla, and M.B. Srivastava, Sewersnort: A drifting sensor for in-situ sewer gas monitoring, Proc. 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009), pp1–9 (2009).

[15] TIK Computer Engineering Group, ETH Zurich, The sensor network museum - MicaZ, http://www.snm.ethz.ch/Projects/MicaZ

Acknowledgments

This work was supported by Kakenhi 2265001.