J Inf Process Syst, Vol.14, No.4, pp.926~940, August 2018
https://doi.org/10.3745/JIPS.03.0102

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Clustering Algorithm Considering Sensor Node Distribution in Wireless Sensor Networks

Boseon Yu*, Wonik Choi**, Taikjin Lee*, and Hyunduk Kim***

Abstract

In clustering-based approaches, cluster heads closer to the sink are usually burdened with much more relay traffic and thus, tend to die early. To address this problem, distance-aware clustering approaches, such as energy-efficient unequal clustering (EEUC), that adjust the cluster size according to the distance between the sink and each cluster head have been proposed. However, the network lifetime of such approaches is highly dependent on the distribution of the sensor nodes, because, in randomly distributed sensor networks, the approaches do not guarantee that the cluster energy consumption will be proportional to the cluster size. To address this problem, we propose a novel approach called CACD (Clustering Algorithm Considering node Distribution), which is not only distance-aware but also node density-aware approach. In CACD, clusters are allowed to have limited member nodes, which are determined by the distance between the sink and the cluster head. Simulation results show that CACD is 20%–50% more energy-efficient than previous work under various operational conditions considering the network lifetime.

Keywords

CACD, Clustering, EEUC, Node Distribution, WSN

1. Introduction

In wireless sensor networks, sensor nodes are generally not rechargeable or replaceable since they are deployed in large numbers; furthermore, the deployed areas are usually disaster areas that are difficult to reach. Thus, considering that sensor nodes are usually powered by limited energy resources, improving the lifetime of the network has always been a primary objective of wireless sensor networks.

Many efficient routing protocols have been proposed to extend the lifetime of wireless sensor networks [1-5]. Clustering-based approaches have especially attracted a great deal of interest because these approaches can achieve high energy efficiency and, at the same time, increase the network scalability. Typically, in clustering-based approaches, cluster heads are elected by local competitions. Usually the highest-energy nodes in the neighborhood are elected as cluster heads. Then, each non-head node picks a cluster head with which to form a cluster. After clustering is finished, cluster heads aggregate the sensed data from the member nodes and forward it to the sink [6-8]. To optimize the

Manuscript received December 8, 2015; first revision May 3, 2016; accepted July 27, 2016.

Corresponding Author: Hyunduk Kim (force80@ctp.or.kr)
* Sensor System Research Center, Korea Institute of Science and Technology, Seoul, Korea (Boseon@gmail.com, taikjin@kist.re.kr)
** School of Information & Communication Engineering, Inha University, Incheon, Korea (wichoi@inha.ac.kr)
*** Policy Planning Agency, Chungnam Techno Park, Cheonan, Korea (force80@ctp.or.kr)

transmission energy consumption, cluster heads eliminate duplicates and redundancies before the actual transmission.

Clustering-based approaches usually employ a multi-hop communication model, which is proven to be more efficient than direct communication model, to deliver the sensed data from the data sources to the sink. However, in such approaches, cluster heads closer to the sink tend to deplete their energy resources much faster than the other cluster heads due to equal clustering. More specifically, cluster heads placed closer to the sink are usually involved in many more traffic relays, and thus, they tend to deplete their energy resources much faster than other cluster heads (hot-spot problem). This may result in partitioning the sensor network as well as reducing the sensing coverage. To address this hot-spot problem, energy-efficient unequal clustering (EEUC) has been proposed in [9]. In EEUC, the cluster size decreases as the distance from the cluster head to the sink decreases. As a result, in EEUC, clusters closer to the sink have smaller cluster sizes than the other clusters. More specifically, EEUC places more, smaller clusters closer to the sink so that they consume less energy for sensing, aggregating, and intra-cluster communications. This unequal clustering may result in fewer burdens on specific cluster heads as well by selecting different routes for delivering data. However, EEUC is less effective for sensor networks with random distributions. EEUC achieves longer lifetimes by making a tradeoff between cluster energy consumption (i.e., sensing and intra-cluster transmissions) and energy consumption for inter-cluster communications. However, EEUC does not guarantee that a smaller-sized cluster will have a smaller node density in a randomly distributed sensor network. Placing small clusters in a crowded area may make EEUC less effective. To address this problem, we propose a novel approach, called CACD (clustering algorithm considering node distribution) that adjusts the size of each cluster considering the node density. Furthermore, CACD automatically adjusts the node density of each cluster on a per-cluster energy-consumption basis. In CACD, we first compute the optimized number of cluster heads for a given sensor network and then, adjust the size and the node density of each cluster according to the distance between the sink and the cluster head. Simultaneously, we estimate the energy level that each cluster requires for sensing and intra-cluster communication to allow for further adjustments. In addition, to keep the sensor network balanced, we partially reconstruct the cluster structures unlike previous methods that rebuild cluster structures at the beginning of each round. The reminder of this paper is organized as follows. In Section 2, we briefly review previous work for wireless sensor network. Section 3 gives the description of the proposed CACD approach. In Section 4, performance evaluation and analysis are given by simulation results. Finally, we conclude in Section 5.

2. Related Work

2.1 LEACH and LEACH-C

Low-energy adaptive clustering hierarchy (LEACH) is the first clustering-based approach for wireless sensor networks [10]. LEACH operates on a round-to-round basis and elects a set of cluster heads by a localized competition at the beginning of each round. To evenly distribute the energy dissipation among all the nodes, LEACH rotates the cluster heads among the nodes. In LEACH, each round has two phases: the *set-up* phase and the *steady-state* phase as shown in Fig. 1. At the beginning of the set-up phase, each node decides whether or not to be a cluster head with a certain possibility, $p_i(t)$:

$$p_i(t) = \begin{cases} \dfrac{CN}{N - CN * \left(r \, mod \, \dfrac{N}{CN}\right)} & , C_i(t) = 1 \\ 0 & , C_i(t) = 0 \end{cases} \tag{1}$$

where N is the total number of nodes, k is the number of clusters and r is the number of rounds that have passed. $C_i(t) = 0$, if node i has been a cluster head in most recent (r mod N/k) rounds and 1 otherwise. Thus, Eq. (1) ensures that all nodes are cluster heads the same number of times.

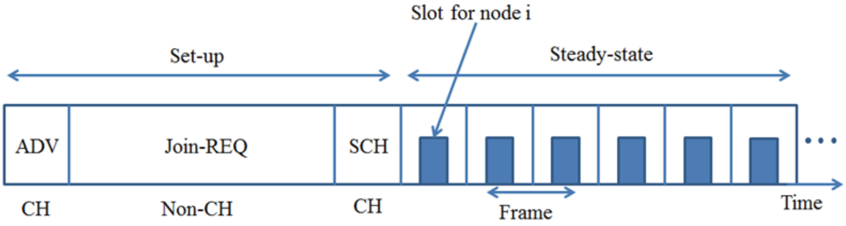


Fig. 1. Round composition of LEACH.

After the cluster head election, each cluster head for the current round broadcasts an ADV message using the CSMA (carrier sense multiple access) protocol to the rest of the nodes. Each non-cluster head node that receives the ADV message decides the cluster to which it belongs for the round, according to the strength of the ADV message. After each non-cluster head node has decided on its cluster, it informs its cluster head that it will be a member of the cluster by sending a Join-REQ message. Then, the cluster head sets up a TDMA (time division multiple access) schedule and broadcasts the schedule to the member nodes. In *steady-state* phase, sensor nodes send their data to the cluster heads during their assigned time slots. To prevent wasted energy, sensor nodes turn off their radio except during their allocated time slots. Cluster heads aggregate the sensed data and forward it to the sink using the CDMA module. Unfortunately, LEACH does not guarantee well-established cluster structures. Since LEACH elects the cluster heads according to residual energy, its cluster structure is highly dependent on the distribution of the sensor nodes. To address this problem, a centralized LEACH clustering algorithm (LEACH-C) has been proposed. However, because LEACH-C assumes that the sink knows the locations of the sensor nodes, this approach is more expensive and requires more transmission cost to form a cluster structure.

2.2 EEUC

Clustering-based algorithms for wireless sensor networks generally use a multi-hop communication model to forward data to the sink. This indicates that cluster heads closer to the sink are involved in many more traffic relays, and thus, tend to deplete their energy resources much faster than other cluster heads. To address this problem, EEUC proposes the unequal clustering scheme shown in Fig. 2. EEUC adjusts the cluster size based on the distance between a cluster head and the sink, such that clusters closer to the sink are smaller. More specifically, EEUC decides the cluster size by Eq.(2).

$$S_i.R_{COMP} = \left(1 - c\frac{d_{max} - d(S_i, sink)}{d_{max} - d_{min}}\right) R^0_{COMP,} \tag{2}$$

where R^{0}_{comp} is the pre-defined maximum cluster size. d_{max} and d_{min} denote the maximum and minimum distances between the sink and a cluster head, respectively. $d(s_i$, sink) indicates the distance between the cluster head, s_i and the sink, and c is a constant coefficient between 0 and 1. According to Eq. (2), the size of a cluster varies from $(1\text{-}c)\,R^{0}_{comp}$ to R^{0}_{comp}. In EEUC, clusters closer to the sink consume less energy for activities inside the clusters (i.e. sensing, aggregating and intra-cluster communications) and thus can afford more traffic relays. However, EEUC becomes less effective for randomly distributed sensor networks because it does not guarantee that a cluster with a smaller cluster size has a smaller node density for such sensor networks. It is obvious that small clusters with a large node density make the unequal clustering-based approach less effective.

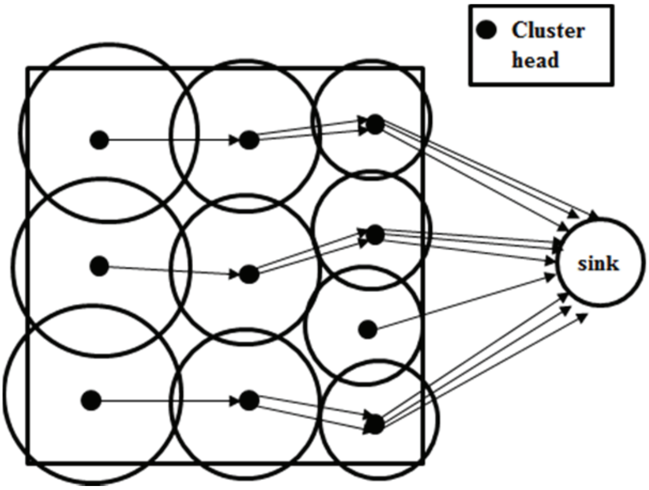


Fig. 2. EEUC (unequal clustering scheme).

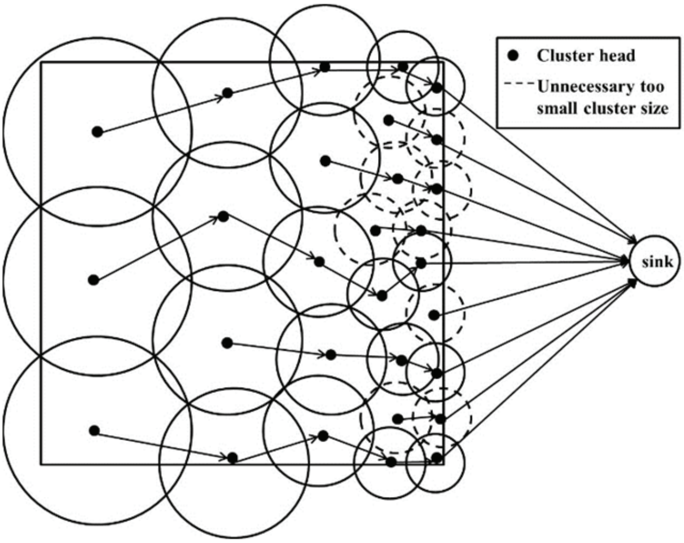


Fig. 3. EEUC.

In addition, for wireless sensor networks with random distributions, EEUC may have clusters that are not involved in any traffic relays due to their small size, as shown in Fig. 3. In Fig. 3, clusters depicted by the dashed line are not involved in any data transmissions and thus the cluster heads will consume much less energy. To extend the network lifetime, it is more desirable for such clusters to be larger. This motivates us to design a protocol with partial re-construction. In our protocol, clusters that are involved in fewer inter-cluster communications are allowed to have more member nodes for optimizations. A more detailed explanation is given in Subsection 3.3.

3. The CACD Scheme

The CACD approach proposed in this paper has three phases: *set-up*, *data-transmit*, and *re-construct*. In the set-up phase, CACD elects a set of cluster heads, and each non-cluster head node chooses a cluster head to form a cluster. Similar to EEUC, CACD adjusts the node density of each cluster according to its distance from the sink. Sensor nodes transmit data to their cluster heads, and then the cluster heads aggregate and forward the data to the sink, using a multi-hop communication model in the data-transmit phase. In the re-construct phase, if certain clusters consume much more energy during the set-up and data-transmit phases, CACD balances the energy consumption by partially re-constructing those clusters. Fig. 4 shows how each round of CACD is organized. Please note that after re-construct phase, we omit for the next several rounds the set-up phase that is quite costly in order to fully utilize the well-established cluster structure as shown in Fig. 4. In Fig. 4, set-up phase is replaced with re-construct phase in round 2 and for the following rounds, the set-up phase is omitted and thus, the result of re-construct phase is maintained. By maintaining a well-established cluster structure and omitting the expensive set-up phase, CACD preserves more energy for inter-cluster communication. More operational details will be described in the next subsections. Without a loss of generality, we assume the following properties about the sensor network:

- The sink is stationary and has unlimited energy resources.
- Each sensor node is capable of measuring its distance from the sink according to the strength of receiving signal.
- Sensor nodes are stationary and have similar capabilities (processing, communication, energy resources).
- Sensor nodes can control their transmission power level according to the distance.

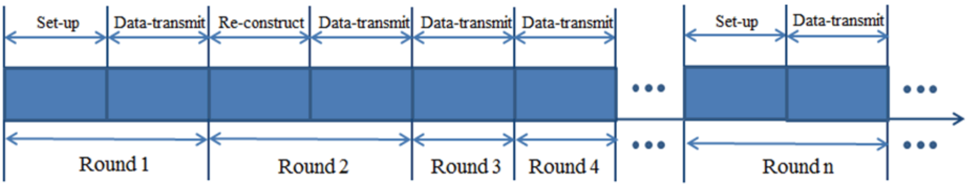


Fig. 4. Time-line of CACD.

3.1 Set-Up Phase

In the set-up phase, CACD elects a set of cluster heads and then, forms clusters with the remaining nodes. CACD employs the hybrid energy-efficient distributed (HEED) cluster-head selection mechanism

[11]. With a given initial percentage of cluster heads, C_{prob}, which is set to 5%, HEED rotates the cluster heads based on their residual energy. Eq. (3) is given for cluster head selection.

$$CH_{prob} = C_{prob} * \frac{E_{residual}}{E_{max}}, \tag{3}$$

where CH_{prob} indicates the probability of becoming a cluster head. $E_{residual}$ and E_{max} are the estimated remaining energy and the maximum energy of a node, respectively. After the cluster-head election, CACD forms clusters using the ADV and Join messages. The sink that overhears the ADV message estimates the distance between a cluster head and the sink according to the signal strength and assigns proper cluster sizes and member nodes to the cluster heads. With this information, each cluster head forms a cluster. The number of member nodes assigned to a cluster follows Eq. (4).

$$OSN = \frac{N}{CN} + P, \tag{4}$$

where, optimal sensor node (OSN) is the number of sensor nodes that should be included in a cluster while *N* and *CN* are the number of sensor nodes and cluster heads for the sensor network, respectively. *P* is a constant to adjust the node density of each cluster according to the distance from the sink. For effective assignment of *P*, CACD follows Eq. (5).

$$P = \frac{d(s_i, sink)}{F} * v, \tag{5}$$

where, $d(s_i$, sink) is the distance between a cluster head, s_i, and the sink. *F* indicates the size of the sensor field, while *v* is a user-defined constant to assign the proper number of member nodes to a cluster. If a cluster has more member nodes than OSN, the cluster is partitioned or passes member nodes to neighboring clusters to build a more energy-efficient cluster structure. We define the threshold, *M,* for cluster partitioning as follows:

$$M = OSN * 1.5 \tag{6}$$

Figs. 5 and 6 give a running example of a cluster that requires partitioning. As shown in Fig. 5, cluster *B* has more sensing nodes than M (=9); thus, cluster B should be partitioned to balance the energy dissipation. For partitioning, the two most-distant nodes are elected as new cluster heads and the new cluster heads form clusters using the ADV and Join messages, as mentioned above. Fig. 6 shows the result of the partitioning procedure. In Fig. 6, cluster B is partitioned into two clusters B' and D. As shown in the running example, CACD balances the energy consumption of each cluster by limiting the maximum number of member nodes in each cluster. Even though the partitioning procedure generates more energy-efficient cluster structures, the procedure can be expensive. Accordingly, CACD applies a simply node transmission procedure to the clusters that have member nodes more or less than OSN for further optimization. In order for all clusters to have as many member nodes as their OSN, those clusters with member nodes more or less than their OSN send OVER or LESS message, respectively to the neighboring cluster heads. The OVER message and LESS message have information on excess and shortage of member nodes. If a cluster with more member nodes (C_O) and a cluster with less member node (C_L) are one-hop neighbor, some member nodes of C_O that are close to C_L are moved to C_L so that all clusters have member nodes as many as their OSN can, Please note that the maximum number of member nodes varies according to the distance from the sink. Thus, the results of CACD may be similar to that of EEUC in that the clusters close to the sink tend to be small. However, CACD guarantees more energy-efficient cluster structures even for crowded areas, by limiting the maximum number of nodes in a cluster, which varies according to the distance from the sink.

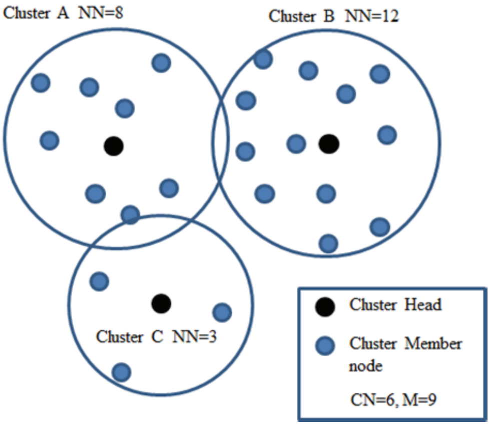


Fig. 5. Unbalanced cluster structure (before split).

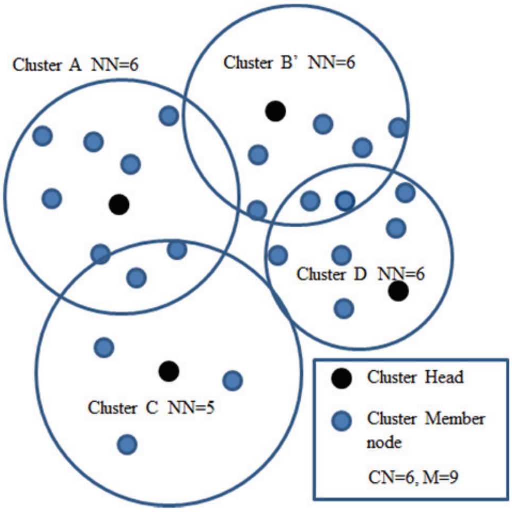


Fig. 6. Balanced cluster structure (after split).

3.2 Data-Transmit Phase

After finishing the set-up phase, cluster heads create TDMA schedules according to the number of member nodes, and pass the schedules to their member nodes. Then, the member nodes transfer their data to the cluster heads during their assigned time slots. After the cluster heads aggregate the data, they forward it to the sink, using CDMA to prevent data collisions. When delivering the data, if the distance between the sending and receiving nodes is less than d_0, CACD uses the Friis free-space model. Otherwise, the two-ray ground propagation model is used.

3.3 Re-construct Phase

During the set-up and data-transmit phases, the cluster heads estimate the energy consumed for building cluster structures and for aggregating and transmitting data to the sink (E_o). In the re-construct phase, cluster heads send E_o to the sink, which estimates the average energy cost (AEC) and re-construct energy cost (REC) of each cluster. Then, the sink sends the AEC and REC to the cluster heads. REC is a threshold for triggering the re-construct phase.

$$AEC = \frac{E_{total}}{CN'} \tag{7}$$

$$REC = AEC * 1.5, \tag{8}$$

where E_{total} indicates the total energy consumption of the sensor network and CN′ is the number of cluster heads. If a cluster head (C_t) has E_o larger than REC, C_t broadcasts a *reconstruct* message to its neighboring cluster heads to re-construct the cluster. Please note that in Eq. (8), the constant coefficient 1.5 can be varied by simulation conditions and thus should be carefully decided by experimental results. If cluster heads (C_r), whose E_o is less than AEC, exist, the member nodes of C_t that are close to C_r become member nodes of C_r. Then, C_r recreates the TDMA schedule and sends it to the member nodes. After the re-construct phase, the set-up phase is omitted for the next k round to utilize this re-constructed cluster structure, as shown in Fig. 4. A more detailed explanation for the best setting of k will be given in Section 4. If all neighboring clusters have E_o larger than AEC, the re-construct phase is aborted and CACD begins the next round so that the entire cluster structure can be re-built in the set-up phase.

CACD algorithm

```
Cluster heads election // HEED;
for each Cluster Head(CH)
 broadcasts ADV message;
 Receive join-request message from non-CH nodes;
 if # of Join request node > M then  // node partitioning
   selects cluster heads;
   partitions the cluster;
 else if # of Join request node > OSN then
   broadcasts OVER message to neighboring CHs;
   if CH receives LESS messages then
     adjusting clusters;
   end if
 else
   broadcasts LESS message to neighboring CHs;
   if CH receives OVER messages then
     adjusting clusters;
   end if
```

```
18:     end if
19:   if $E_0 > REC$ then
20:     broadcasts reconstruct message to neighboring CHs;
21:     if CH receives cluster extend message then
22:       re-construction;
23:     else
24:       goto line 1;
25:     end if
26:   else if CH receives reconstruct messages then
27:     if $E_0 < AEC$ then
28:       broadcast cluster extend message to neighboring CHs;
29:       re-construction;
30:     end if
31:   end if
32: end for
```

The above algorithm gives a sketch of CACD. In line 1, we elect the cluster heads according to the mechanism used in HEED. Then, CACD builds a cluster structure in the remainder of the algorithm. While building the cluster structure, a cluster whose member nodes are more than M is partitioned in lines 5-7. When a cluster head has more member nodes than OSN, the cluster head finds clusters that are able to have more member nodes, and passes its member nodes to them in lines 8-18. In line 19, each cluster decides whether to trigger the re-construction phase. If E_o > REC, the cluster head triggers re-construct phase by broadcasting a *reconstruct message* in line 20 and waits for the *cluster-extend message*. If the cluster head receives the *cluster-extend message*, the re-construct phase is triggered in line 22. Otherwise, CACD simply moves to the next round so that the entire cluster structure is re-built in the set-up phase in line 24. If a cluster whose E_o is less than AEC receives a *reconstruction messages*, it broadcasts a *cluster-extend message* and triggers the re-construction phase in lines 26-31.

4. Performance Evaluation and Analysis

In this section, we evaluate the performance of our approach, CACD by comparing it against LEACH, HEED, and EEUC. We use Network Simulator 2 (NS2; http://www.isi.edu/nsnam/ns) for simulations and, without a loss of generality, the energy consumption model used in the simulations is the first-order radio model (Eq. (9)), which is also used for LEACH and EEUC.

$$\begin{aligned} E_{TX}(t,d) &= E_{elec} * t + \varepsilon_{amp} * t * d^n \\ &= \begin{cases} E_{elec} * t + \varepsilon_{fs} * t * d^2, d < d_0 \\ E_{elec} * t + \varepsilon_{mp} * t * d^4, d > d_0 \end{cases} \\ E_{RX}(t,d) &= E_{elec} * t \end{aligned} \tag{9}$$

E_{TX} in Eq. (9) indicates the energy consumed by transmitting data. E_{elec} represents the energy consumed for processing one bit of data, while t and d represent the data size and the transmission distance, respectively. $\mathcal{E}_{amp}$ is the energy required to amplify the signal. When $d < d_0$, we use $\mathcal{E}_{fs}$, which is from the Friis free space model. Otherwise, $\mathcal{E}_{mp}$ from the two-ray ground propagation model is used. E_{RX} represents the energy consumed when receiving data. The required d_0 here is defined by Eq. (10).

$$d_0 = \frac{4\pi\sqrt{L}h_r h_t}{\lambda}, \tag{10}$$

where L is the system loss factor, which is not related to the wave propagation while λ is the wave length of the carrier signal. h_t and h_r are the heights above the ground of the transmitting and receiving antennas, respectively.

L = 1 indicates no loss in the system hardware. $h_t = h_r$ = 1.5 m, and λ is related to the carrier frequency. If we use a carrier frequency of 914 MHz, λ is as follows:

$$\lambda = \frac{3 * 10^8}{914 * 10^6} = 0.328 \tag{11}$$

Using these parameters, d_0 = 86.2 m. The simulation parameters are given in Table 1. All of the alternatives are compared under the same setup, and we use the scenario generator of NS2 to generate randomly deployed sensor nodes, as shown in Fig. 7.

Table 1. Simulation parameters

Parameter	Value
Network coverage (m)	(0, 0)–(200, 200)
Sink location (m)	(-100, 100)–(100, 100)
N	400
Initial energy (J)	2
t (bit)	4,000
d_0 (m)	86.2
E_{elec} (nJ/bit)	50
$\mathcal{E}_{fs}$ (pJ/bit/m^2)	10
$\mathcal{E}_{mp}$ (pJ/bit/m^4)	0.0013
E_{DA} (nJ/bit/signal)	5
C	0.5
Initial cluster head rate (%)	5
P	0.05

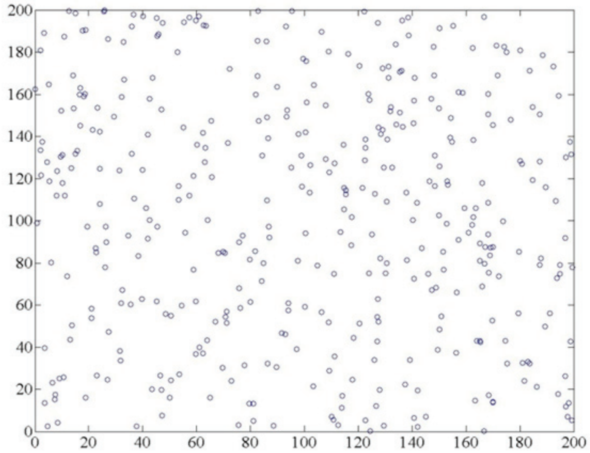


Fig. 7. Random deployment of sensor nodes.

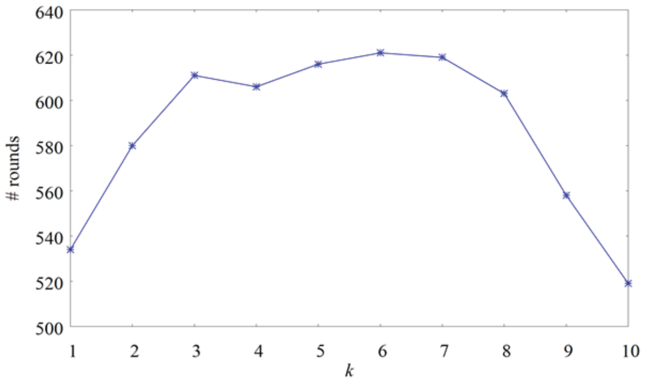


Fig. 8. The impact of *k* on the network lifetime of CACD.

The first set of our simulation is to find the best value for *k* (Section 3.3). By omitting the set-up phase for *k* rounds after the re-construct phase, CACD preserves more energy for intra-cluster communication and thus can extend the lifetime of the sensor networks. In Fig. 8, we count the rounds until the first node dies, while *k* varies from 1 to 10. As shown in the figure, CACD shows poor performance when *k* is less than 3. This is because, with such small *k* values, CACD cannot fully utilize the benefit of the re-construct phase. More specifically, during the re-construct phase, CACD produces a more energy-efficient cluster structure, but consumes extra energy in the process. Thus, in order to compensate for the extra energy and preserve energy for inter-cluster communications, CACD omits the set-up phase and maintains the result of the re-construct phase for the next *k* rounds. However, Fig. 8 shows that CACD cannot achieve these goals with such small *k* values. In addition, CACD becomes less effective when *k* is greater than 8. This is because maintaining a cluster structure for such a long period causes a rapid depletion of the cluster heads' energy resources. This rapid depletion makes our approach less effective, as shown in Fig. 8. As a result, in order to optimize the CACD's performance, we omit the set-up phase for the next 6 rounds after re-construct phase, which is the best setting of *k*, as shown in Fig. 8.

The next set of our simulation is to study the effectiveness of CACD. We evaluate and compare the energy consumption of each sensor node as shown in Fig. 9. Fig. 9 shows the average energy consumption of each round for the four alternatives. CACD is up to 3 times more energy efficient than LEACH, and up to 1.6 times more energy efficient than HEED and EEUC. LEACH's poor performance is because it does not take into account the node density and the size of each cluster. Furthermore, the direct communication between each cluster head and the sink highly contributes to the poor performance.

HEED outperforms LEACH mainly due to the multi-hop communication model for forwarding data, but may suffer from the hot-spot problem since cluster heads closer to the sink are burdened with forwarding data [12]. This hot-spot problem accelerates the energy consumption of neighboring nodes and, in turn, may result in partitioning the sensor network as well as reducing the sensing coverage, which is undesirable for wireless sensor networks.

As shown in Fig. 9, EEUC consumes less energy than HEED due to its distance-aware clustering approach, but may become less effective for a wireless sensor network with a random distribution. CACD extends EEUC by considering the node density of each cluster and successively reduces the energy consumption of each round, as shown in Fig. 9.

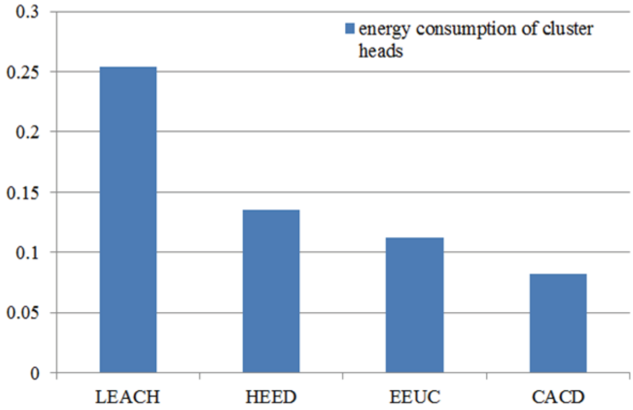


Fig. 9. Average energy consumption per round.

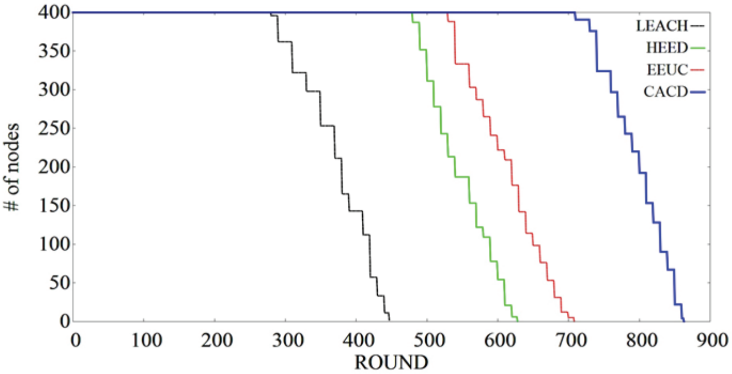


Fig. 10. The number of alive sensor nodes over time.

Fig. 10 shows the number of sensor nodes that are still alive over the simulation time of the four alternatives. *CACD* clearly improves the lifetime of the network over LEACH, HEED, and EEUC.

When forming cluster structures, other alternatives do not consider the node density and energy consumption of each cluster and may create unbalanced cluster structures. These unbalanced cluster structures may waste unnecessary energy while processing and transmitting data. In contrast, CACD forces a cluster to have fewer member nodes than OSN, which varies according to the distance from the sink. This property of CACD guarantees more balanced energy load over the sensor network, even if the sensor network has a random distribution.

The last set of our experiments is to observe the tradeoff resulting from keeping the cluster structures. Fig. 11 shows energy consumptions of cluster heads on CACD and HEED. EEUC selects cluster heads considering the energy level of each node as well but we compare CACD only with HEED because detailed selection algorithm is not given in the paper. As shown in the figure, CACD reports that cluster heads consume more energy than HEED until 2 ROUND due to set-up phase and re-construct phase but after the round, CACD consumes less energy since CACD maintains the cluster structure for 7 rounds. The cluster heads maybe more burdened in the CACD but the stable cluster structure minimizes the burdens and makes CACD more energy efficient considering the overall energy consumption.

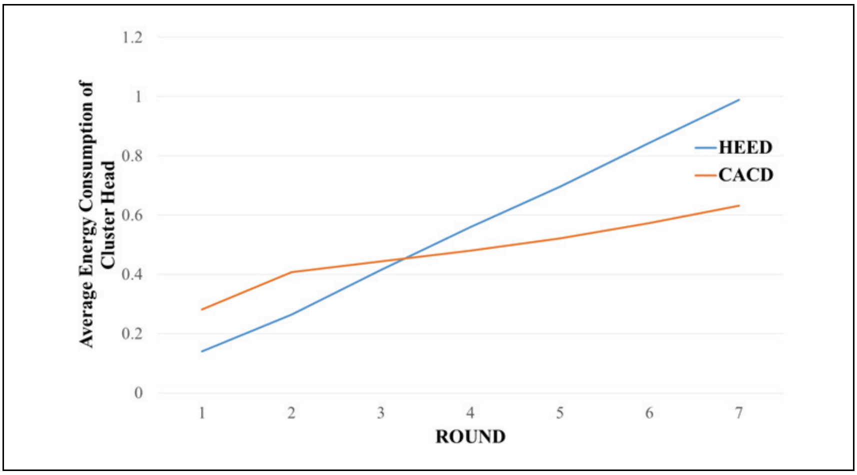


Fig. 11. Average energy consumption of each cluster over ROUNDs.

5. Conclusion

In this paper, we proposed the CACD approach that aimed to prolong the lifetime of wireless sensor networks, which are usually powered by limited energy resources. The CACD approach creates cluster structures that can balance energy consumption, even for randomly distributed wireless sensor networks, by considering the node density and the energy consumption of each cluster. Analysis and performance evaluations showed that the CACD approach is up to three times more energy-efficient than previous methods.

As future work, we plan to extend our approach by exploring the cooperative communication model [13,14] to create a more energy-efficient cluster structure.

References

[1] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325-349, 2005.

[2] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 551-591, 2013.

[3] V. Snasel, L. Kong, P. Tsai, and J. S. Pan, "Sink node placement strategies based on cat swarm optimization algorithm," *Journal of Network Intelligence*, vol. 1, no. 2, pp. 52-60, 2016.

[4] L. Kong, J. S. Pan, P. W. Tsai, S. Vaclav, and J. H. Ho, "A balanced power consumption algorithm based on enhanced parallel cat swarm optimization for wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 11, no. 3, article no. 729680, 2015.

[5] F. C. Chang and H. C. Huang, "A survey on intelligent sensor network and its applications," *Journal of Network Intelligence*, vol. 1, no. 1, pp. 1-15, 2016.

[6] T. T. Nguyen, T. K. Dao, M. F. Horng, and C. S. Shieh, "An energy-based cluster head selection algorithm to support long-lifetime in wireless sensor networks," *Journal of Network Intelligence*, vol. 1, no. 1, pp. 23-37, 2016.

[7] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113-11153, 2012.

[8] P. Kumarawadu, D. J. Dechene, M. Luccini, and A. Sauer, "Algorithms for node clustering in wireless sensor networks: a survey," in *Proceedings of the 4th International Conference on Information and Automation for Sustainability*, Colombo, Sri Lanka, 2008, pp. 295-300.

[9] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," in *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, Washington, DC, 2005, pp. 597-604.

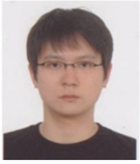
[10] W. Heinzelman, "Application specific protocol architectures for wireless networks," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[11] 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.

[12] K. Kim and B. Cha, "A topology control scheme for avoiding sensing hole in wireless sensor networks," in *Proceedings of International Conference on Computational Science and its Applications*, Kuala Lumpur, Malaysia, 2007, pp. 405-408.

[13] H. Y. Kong, "Energy efficient cooperative LEACH protocol for wireless sensor networks," *Journal of Communications and Networks*, vol. 12, no. 4, pp. 358-365, 2010.

[14] H. Kim, C. Kim, J. Kim, M. Seo, S. Lee, and T. Lee, "A context aware cooperative communication method in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 6, article no. 357509, 2015.



Boseon Yu https://orcid.org/0000-0002-0336-5211

He received the M.S. and Ph.D. degree in School of Information and Communication Engineering from Inha University, Incheon, Korea in 2010 and 2015, respectively. From September 2015 to now, he has been working as a post-doctoral researcher for Sensor System Research Center in Korea Institute Science and Technology, Korea. His research interests include database system, GPU and sensor network system.

Wonik Choi https://orcid.org/0000-0001-8533-2829

He is an Associate Professor in the School of Information and Communication Engineering at Inha University, where he runs Ubiquitous Computing Lab. He received his Ph.D. in Computer Engineering from Seoul National University, Korea. His research interests include spatio-temporal databases, sensor network topology and GIS/LBS. He was a visiting scholar in the school of Computer Science at Georgia Institute of Technology in 2012. He is a member of IEEE since 2006.

Taikjin Lee https://orcid.org/0000-0001-6158-0705

He received the B.S. and Ph.D. degrees in School of Mechanical and Aerospace Engineering in Seoul National University, Seoul, Korea in 2001 and 2008, respectively. In 2008, he was with the School of Mechanical and Aerospace Engineering, Seoul National University where he was a postdoctoral fellow. Since 2010, he has been with Korea Institute of Science and Technology as a senior researcher. The areas of interests are indoor navigation system, pattern recognition, and sensor network.

Hyunduk Kim https://orcid.org/0000-0002-3901-1327

He received M.S. and Ph.D. degrees in School of Information and Communication Engineering from Inha University in 2009 and 2014, respectively. From August 2014 to March 2016, he worked as a post-doctoral researcher for Sensor System Research Center in Korea Institute Science and Technology, Korea. He is now with the Policy Planning Agency in the Chungnam Techno Park as a Senior Researcher, since April 2016. His current research interests include WSN routing protocol, cooperative communication, wireless mobile communication, bio-sensor device and indoor LBS.