Computer Networks 101 (2016) 144–157

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

ICP: Instantaneous clustering protocol for wireless sensor networks

Linghe Kong[a], Qiao Xiang[b,c], Xue Liu[d], Xiao-Yang Liu[a,*], Xiaofeng Gao[a], Guihai Chen[a], Min-You Wu[a]

[a] Shanghai Jiao Tong University, China
[b] Yale University, USA
[c] Tongji University, China
[d] McGill University, Canada

ARTICLE INFO

Article history:
Received 11 July 2015
Revised 15 November 2015
Accepted 6 December 2015
Available online 14 January 2016

Keywords:
Instantaneous clustering protocol
Parallel clustering
Wireless sensor networks

ABSTRACT

Wireless sensor network (WSN) is one of the mainstay technologies in Internet of Things. In WSNs, clustering is to organize scattered sensor nodes into a cluster-topology network for communications. Existing efforts on clustering intensively focus on the energy-efficiency issue. However, in mission-critical applications, a fast clustering scheme, which can not only gather sensory data immediately after deployment but also reduce the energy consumption, is more desired. In this paper, we study the clustering problem considering both time- and energy-efficiency. We propose a novel instantaneous clustering protocol (ICP) that groups sensor nodes into single-hop clusters in a parallel manner. ICP can instantaneously complete the clustering due to two key designs. First, to determine the cluster heads locally. Existing methods require a long duration on cluster head voting. To waive the voting consumption, a cluster head in ICP is locally determined by the pre-assigned probability and its present status. Second, to minimize the amount of transmissions. Parallel transmissions from different cluster heads and acknowledgments (ACKs) from multiple cluster members lead to severe time and energy consumption. On the contrary, ICP gets rid of the ACK mechanism, instead, only cluster heads contend to broadcast during a given period. This period is elaborately derived to guarantee the connectivity. Experiments on a 64-node testbed and simulations on large-scale WSNs are extensively conducted to evaluate ICP. Performance results demonstrate that ICP significantly outperforms existing clustering methods by reducing up to 55% time consumption and 89% amount of transmissions for energy-saving.

1. Introduction

The widespread deployment of wireless sensor networks (WSNs) [3,11,13,34] has fostered the rise of the Internet of Things such as monitoring for smart building [31] and crowdsensing based environment acquisition [27,35], attracting lots of interests due to their great potentials on perceiving the physical world. Generally, scattered sensor nodes require a reliable network structure for communications [22]. The cluster topology, which forms sensor nodes into several clusters and where every cluster is dominated by a cluster head (CH), is one of the most widely adopted structures. The procedure of setting up such a cluster topology is called clustering.

* Corresponding author.
E-mail addresses: linghe.kong@sjtu.edu.cn (L. Kong), xiangq27@gmail.com (Q. Xiang), xueliu@cs.mcgill.ca (X. Liu), yanglet@sjtu.edu.cn (X.-Y. Liu), gao-xf@cs.sjtu.edu.cn (X. Gao), gchen@cs.sjtu.edu.cn (G. Chen), mwu@sjtu.edu.cn (M.-Y. Wu).

http://dx.doi.org/10.1016/j.comnet.2015.12.021
1389-1286/

In the literature, there are plenty of clustering methods for WSNs. These methods have different objectives in order to facilitate various requirements from applications. The classic LEACH [14] and HEED [37] pay great attention to the energy-efficient clustering, which is the commonest objective. Some other objectives are as follows: WCA [6] considers the node degree as an important parameter, DSBCA [25] aims to build a load-balanced cluster topology, FT-EEC [18] designs a fault-tolerate clustering method, etc. However, the objective of fast clustering hardly receives investigation.

In this work, we focus on the study of fast and energy-efficient clustering. From the perspective of time-efficiency, a fast clustering is significant in WSNs, especially in mission-critical applications. For example, in military surveillance [7,28], gathering enemies' information immediately after deployment is critical to build up the informational competitive advantage. Disaster relief [21] is another example that an early detection of danger can save more lives. In general applications, a WSN usually demands several times of re-clustering in its lifetime. An instantaneous clustering can change its topology without stopping the sensing tasks. From the perspective of energy-efficiency, a fast clustering indicates a lightweight clustering process with low transmission overhead, and has the potential to cut down the total energy consumption.

In order to accelerate the clustering procedure, we aim to organize sensor nodes into single-hop clusters [38] in a parallel manner. On one hand, we adopt the cluster topology because all clusters have the potential to self-organize concurrently. In this way, the duration of clustering an entire WSN is reduced to the duration of organizing just one cluster. On the other hand, we adopt the single-hop pattern because organizing a single-hop cluster is obviously faster than organizing a multi-hop one. In this way, the duration of organizing every cluster is minimized. As a result, this parallel clustering method significantly reduces the time consumption.

There are three challenges on realizing the parallel clustering method. First, every single-hop cluster demands one CH dominating other sensor nodes. Conventional methods vote CHs via packet exchange among neighbors [14,37], which consumes a large amount of time and energy. It is challenging to waive this consumption while CHs need to be well determined. Second, the collision problem is another non-trivial issue. The parallelism design causes concurrent transmissions from different CHs, which results in collisions, especially in dense WSNs [20,40]. To deal with collisions, conventional methods usually exploit collision avoidance and acknowledgment (ACK) mechanisms to ensure the packet delivery, which causes extra consumption. Third, compared with the lifetime of a WSN, the clustering or re-clustering process is relatively short. Hence, a lightweight algorithm is desired to be easily implemented in off-the-shelf sensor nodes.

To address these challenges, we propose a novel Instantaneous Clustering Protocol (ICP) to cluster stochastically scattered sensor nodes. First, ICP derives the probability of a sensor node to be a CH and pre-assigns this probability to every node. Each CH is locally determined based on the pre-assigned probability and its present status, so ICP removes the redundant CH voting. Second, the amount of transmissions is minimized to eliminate the collisions. In ICP, the ACK mechanism is exempted and only CHs contend to broadcast their packets during a given period. We carefully derive this period as the minimal time consumption subject to guaranteeing the packet delivery. Third, compared with existing methods [14,37], ICP is much more lightweight on both the computational complexity and the communication overheads.

We implement and evaluate ICP on NetEye [17], a real WSN testbed with 64 TeloB nodes. To study the scalability of ICP, extensive simulations are also conducted to perform ICP in large-scale WSN scenarios. Both experiment and simulation results show that ICP significantly outperforms existing methods in terms of time and energy consumption. Benefitting from the parallelism design, ICP completes the clustering in a nearly constant duration even if the density of nodes grows up.

The main contributions of this paper are two-folds:

- Towards a fast and energy-efficient clustering, we propose the instantaneous clustering protocol (ICP) to cluster a WSN in a parallel manner. Specifically, a CH is locally determined with a pre-assigned probability and the duration of organizing a single-hop cluster is minimized.
- We implement and evaluate the lightweight ICP in a real WSN testbed. In addition, we conduct extensive simulations to further understand ICP. Performance results demonstrate the feasibility and effectiveness of ICP.

The rest of the paper is organized as follows. In Section 2, related work is discussed. In Section 3, we state the problem. The design of ICP is proposed in Section 4. In Section 5, we present and analyze the ICP algorithm. In Section 6, we implement and evaluate ICP. In Section 7, simulations are conducted to further understand ICP. Our work is concluded in Section 8.

Table 1
Comparison of typical clustering methods.

Methods	Objective	CH determination	ACKs
LEACH [14]	Energy-efficient	Pre-assigned	Yes
HEED [37]	Energy-efficient	Voting	Yes
ECDS [4]	Energy-efficient	Voting	Yes
WCA [6]	Load-balance	Voting	Yes
DSBCA [25]	Load-balance	Voting	Yes
GS^3 [39]	Fault-tolerant	Voting	Yes
FT-EEC [18]	Fault-tolerant	Pre-assigned	Yes

2. Related work

Numerous clustering methods have been studied in WSNs [1,2,24]. We classify existing methods into three categories according to their objectives: energy-efficient, load balanced, and fault-tolerant. A brief comparison of several typical methods belonging to these categories is summarized in Table 1.

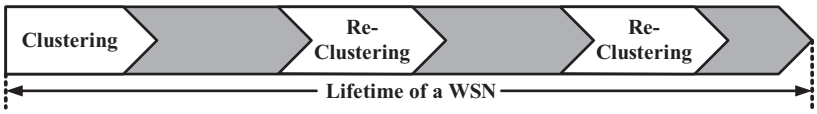


Fig. 1. The lifetime of a WSN includes clustering and re-clustering phases.

Energy-efficient clustering. LEACH [14] is the most classic clustering method in this category. In LEACH, every sensor node generates a random probability from 0 to 1 at the beginning of the clustering process. If this probability is larger than a pre-assigned threshold, this node becomes a CH and starts to organize its single-hop cluster. Many follow-up methods are put forward relying on LEACH.

Taking the intra-cluster communication overhead into account, HEED [37] overcomes the shortcoming of unevenly distributed CHs in LEACH. In addition, the residual energy is introduced as an important parameter to vote CHs in HEED.

The state-of-the-art energy-efficient clustering method is ECDS [4], which selects CHs using a constrained dominating set approach. Although ECDS outperforms HEED on both energy and time consumption, the improvement is not significant since there is no customized design to eliminate the effects of voting and ACKs.

Load-balanced clustering. WCA [6] is a classic clustering method for connectivity balance, in which the voting of CHs relies upon the node degree. The main drawback is that WCA requires the weights of nodes, so these communications cause extra consumption.

Instead of acquiring the weights, the recent DSBCA [25] calculates the radius of cluster based on distance and distribution. Moreover, it takes the number of neighbors and the residual energy into account when voting CHs.

Fault-tolerant clustering. The classic GS^3 [39] clusters a WSN into cellular hexagon structure. In order to provide a self-healing network, the CH in every hexagonal cell is re-voted when any sensor node joins, leaves, or fails.

In recent FT-EEC [18], the detection of failure is based on periodic reports. Once the failure of a sensor node is detected, the clustering process is triggered to re-organize the network using a LEACH-like energy-efficient manner.

Summary. Although existing methods cluster WSNs with various objectives, little work concentrates on the objective of minimizing the time consumption of clustering. In addition, no works pay attention to the ACK mechanism, which consumes much time and energy during intra-cluster organization. To operate a clustering, existing methods have to interrupt the sensing tasks in WSNs. Therefore, we are motivated to design a new clustering method, which could achieve an instantaneous clustering with low energy consumption.

3. Problem formulation

In this section, we present the system model and the problem statement.

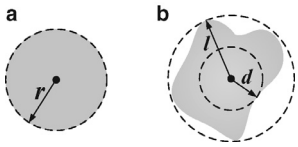


Fig. 2. The disk model *vs.* the quasi disk model.

3.1. System model

The disk model [32] is widely employed in the study of WSN communications, which assumes the transmission coverage is a disk in the plane with radius r as shown in Fig. 2(a). This model is a simplification of the reality. Even for homogeneous sensor nodes, the transmission range is affected by external factors such as link quality. Hence, the quasi disk model [23] is much closer to the reality. In this model, two sensor nodes can successfully receive messages from each other if their Euclidean distance is less than d. And in the range between d and l, the successful reception probability is unspecified as shown in Fig. 2(b). When the distance is larger than l, the successful reception probability is zero. In the remainder of this paper, we will resort to the disk model with transmission range $r = (d + l)/2$ in theoretical analysis for readability, and the quasi disk model in simulations for approaching reality.

We study the fast and energy-efficient clustering problem in the following scope: The total number of sensor nodes is n. These n sensor nodes are homogenous and stationary. These sensor nodes are scattered randomly into a given area and they follow a stochastic distribution [21]. All nodes start the clustering phase at the same time leveraging the preset synchronization protocol [9]. In addition, the area of interest is known. For simplicity in analysis, we assume that the area is a square in the plane, whose length of side is denoted by a.

3.2. Problem statement

It is necessary for a WSN to establish a connected and reliable network structure, so that it can transmit their sensory data. The procedure of establishing such a cluster topology is called *clustering*. Normally, the lifetime of a WSN (if the cluster topology is adopted) includes several clustering and re-clustering phases [39] as shown in Fig. 1. On one hand, the clustering process is required at the beginning of the lifetime, because newly deployed sensor nodes need to initialize a network structure. On the other hand, the re-clustering processes are required to repeat several times during the lifetime in order to balance the energy consumption at CHs [37]. We define that a clustering process begins when all sensor nodes are ready to

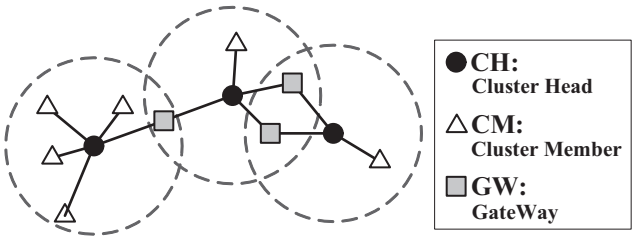


Fig. 3. An example of the single-hop cluster topology.

run the clustering algorithm, and ends when all nods are connected.

In this paper, we propose to study the *fast and energy-efficient clustering* problem in WSNs.

The major objective of this problem is to minimize the total time consumption of clustering $\min(T_{tot})$. Typically, the clustering is a procedure of message exchange. Sensors are required to transmit 'clustering' messages to neighbors. Assume that transmitting one message costs one time slot, where every slot has the same duration t. Thus, T_{tot} is formulated by $T_{tot} = xt$, where x presents the total number of time slots required to complete the clustering. Since t is fixed value determined by the size of 'clustering' message and the bit rate in ZigBee [16] standard, the objective $\min(T_{tot})$ can be treated as minimizing the total number of time slots $\min(x)$.

This problem also aims to mitigate the total energy consumption E_{tot}. During the clustering, energy is mainly consumed by message exchange. We assume that all sensor nodes are set at the same power level during clustering. Hence, the energy consumption of transmitting one message is denoted by E_{TX}, and the energy consumption of receiving/listening in one time slot is denoted by E_{RX}. Since a node is at either transmitting or receiving state in a time slot, we have

$$E_{tot} = \lambda E_{TX} + (xn - \lambda)E_{RX}, \tag{1}$$

where λ is the number of transmitted messages required to complete the clustering, x is the total number of time slots, and n is the number of sensor nodes. Since E_{TX} and E_{RX} (usually $E_{TX} > E_{RX}$) [29] are fixed values determined by hardware, $\min(x)$ and $\min(\lambda)$ can result in $\min(E_{tot})$.

4. Instantaneous clustering protocol

In this section, we present the idea about the parallel clustering of single-hop clusters first. Then we analyze the design challenges. Afterwards, the overview of ICP is introduced. Finally, we theoretically optimize the parameters in ICP to tackle the challenges.

4.1. Parallel clustering of single-hop clusters

Before introducing the explicit design, we present the core idea of ICP, which is the parallel clustering of single-hop clusters for scattered sensor nodes.

Parallelism [8] is popular in computing systems. We resort to the parallelism concept to achieve a fast clustering in WSNs. Unlike the tree topology's top-down structure [10], the cluster topology organizes sensor nodes into several groups. These groups have the potential to self-organize in a parallel manner. Thus, the parallel clustering can dramatically reduce the duration of clustering an entire network to be the duration of organizing only one cluster.

In order to further accelerate the clustering process, the duration of intra-cluster organization is required to be minimized. To this end, we adopt the single-hop cluster scheme [26] in our method. Compared with a multi-hop cluster [5], organizing a single-hop cluster is apparently faster, because it does not require time for multi-hop relays.

Sensors have three roles in the single-hop cluster scheme: cluster head (CH), cluster member (CM), and gateway (GW). Fig. 3 illustrates an example of these three roles, where any CM is only connected with one CH and one CH can connect multiple CMs/GWs. In addition, any two CHs are connected either directly or by single-hop, so the distance between any two CHs is $d(CH_1, CH_2) < 2r$.

4.2. Design challenges

We analyze three major challenges in the fast and energy-efficient clustering problem. Along with the challenges, we briefly describe the clues of our solution.

Challenge 1. Existing methods [37,39] usually determine CHs by voting, which costs much consumption on information exchanged among neighbors. In order to reduce such a consumption, we propose to waive the voting and determine CHs locally. Thus, x in clustering process can be decreased.

A challenging problem arises from the above proposal is how to determine CHs without voting? Our solution follows the clues: Firstly, since we know some information in advance such as the total number of sensor nodes n, the transmission range r, and the length of area side a, we can derive the expectation number of CHs in a WSN under stochastic distribution. Secondly, we introduce the redundancy coefficient of CHs against the uncertain positions during deployment. Thirdly, we pre-assign a probability to every sensor node. This probability is calculated based on the expectation number and the redundancy coefficient. Fourthly, CHs are locally determined by this pre-assigned probability and its present status such as residual energy. Theoretical derivation of them are provided in Section 4.4.

Challenge 2. A WSN completes its clustering on the following basic demands: a sensor node knows whether it is a CH, a CM or a GW. As a CM/GW, it must additionally know which CHs it links to. In other words, at least one path between any pair of nodes can be found, so that existing routing protocols [30,33] can be applied on the cluster topology. Since the CHs in our method are locally determined, the basic demands can be satisfied when ID messages broadcasted from CHs can be received by their CM/GWs.

Existing methods adopt the ACK mechanism [36] to confirm the successful reception, where a CM/GW responses an ACK to the CH once it receives the CH's ID. Nevertheless, concurrent ACKs from multiple CM/GWs in one cluster cause collisions, and consume much time and energy to deal with them. We propose to exempt the ACK mechanism. Then, the challenging problem is how to guarantee the delivery of CHs' ID messages. To this end, we introduce a period for CHs to contend and broadcast their IDs. This period includes the minimal number of time slots, which is adequate for at least one CH to transmit its ID without collisions. We derive the optimal number of time slots in Section 4.5.

Challenge 3. Compared with the whole lifetime of a WSN, the clustering and re-clustering processes are relatively short. In addition, the computational capability of sensor nodes are limited. Hence, the clustering method is desired to be lightweight, so it can be implemented easily in practice. We analyze the computational complexity and communication overheads of ICP to demonstrate its lightweight in Section 5.2, and then implement ICP in Section 6.

Other issues. There are some other minor but practical issues, which should be taken into account in ICP design. Since all sensor nodes are pre-assigned the same probability to be CHs, it is possible that multiple close nodes compete for one CH position. Hence, an abdication mechanism is necessary to avoid dense CHs in small area. We propose when a sensor node hears other CHs' ID messages before broadcasting its own, it abdicates from CH to be a CM/GW.

It is also possible that a few CM/GWs are not located within any CHs' transmission range, namely, isolated nodes. Moreover, several CM/GWs may fail to receive messages from CHs due to no ACK. To guarantee the connectivity of the entire network, a compensation mechanism is necessary to consider these sensor nodes into account.

4.3. Design overview

Based on the above analysis, we design the Instantaneous Clustering Protocol (ICP). The procedures of ICP include:

- *Pre-assignment*: ICP estimates the number of CHs, denoted by m, based on the area of the given field a^2, the total number of sensor nodes n, and the transmission range of sensor nodes r. Then, ICP computes the probability of a sensor node to be a CH $P_{CH} = \beta m/n$, where β is a constant coefficient. This probability is pre-assigned to every node before deployment.
- *Self-organization*: After the deployment, all sensor nodes start to self-organize synchronously. This step has T time slots. A sensor node becomes a CH candidate with probability P_{CH}. Then, CH candidates either contend to CHs by transmitting their ID messages or become CM/GWs by abdication mechanism. The present status is the jetton for competition during the given period. For example, a CH candidate with high residual energy has large probability to transmit its ID early. Any CM/GW is allocated into different clusters if it successfully receives messages from CHs.
- *Compensation mechanism*: If isolated nodes still exist after T, some of them will upgrade into CH candidates in additional Δ time slots and will repeat the self-organization step.

According to the procedure of ICP, the total number of time slots is $x = T + \Delta$. The expectation of transmission amount is m, because only m CHs transmit their IDs once per CH without retransmissions or ACKs.

4.4. The number of CHs

In order to address Challenge 1, ICP estimates the number of CHs m before deployment. Here, we derive how many CHs is sufficient to make a WSN connected by single-hop clustering. Even before the stochastic deployment, some information are known including the total number of sensor nodes n, all nodes placed randomly and independently in the a^2 area, and the transmission range of sensor nodes r. Let $\mathcal{G}(n, m)$ denote the graph of this WSN. And two nodes are connected if their Euclidean distance is no more than r, then we have:

Theorem 4.1. *When the number of cluster heads is* $m = C(\log n)\frac{a^2}{r^2}$, *the graph* $\mathcal{G}(n, m)$ *of a WSN is connected with probability one as* $n \to \infty$, *where C is a tunable parameter.*

We prove the necessary and sufficient conditions of Theorem 4.1 in the following two parts, respectively.

4.4.1. Necessary condition on m

For the sake of simplicity, we neglect edge effects when a node is close to the boundary of the area and we define a ratio $\gamma = \frac{r}{a}$. The proof consists of two technical lemmas and one corollary.

Lemma 4.1. *If* $m = \frac{\log n + \varpi}{\pi\gamma^2}$ *for any fixed* $\alpha < 1$ *and for all sufficiently large n, we have*

$$n(1 - \pi\gamma^2)^m \geq \alpha e^{-\varpi}, \tag{2}$$

where ϖ *is a given value.*

Proof. Taking the logarithm of the left hand side of Eq. (2), we get

$$\log(n(1 - \pi\gamma^2)^m) = \log n + m\log(1 - \pi\gamma^2). \tag{3}$$

Using the power series expansion for $\log(1 - x)$,

$$\log(n(1 - \pi\gamma^2)^m) = \log n - mg\left(\sum_{i=1}^{2}\frac{(\pi\gamma^2)^i}{i} + H(n)g\right), \tag{4}$$

where

$$H(n) = \sum_{i=3}^{\infty} \frac{(\pi\gamma^2)^i}{i} \le \frac{1}{3}\int_3^{\infty} (\pi\gamma^2)^x dx = \frac{1}{3(\pi\gamma^2)}(\pi\gamma^2)^x g|_3^{\infty} = \frac{1}{3}(\pi\gamma^2)^2 \quad (5)$$

for all large n. Substituting Eq. (5) in Eq. (4), we get

$$\log(n(1-\pi\gamma^2)^m) \ge \log n - mg(\pi\gamma^2 + \frac{5(\pi\gamma^2)^2}{6}g) = -\varpi - \frac{5\pi\gamma^2(\log n + \varpi)}{6} = -\varpi - \delta, \quad (6)$$

where $\delta = 5\pi\gamma^2(\log n + \varpi)/6$.

Taking the exponent of both sides of Eq. (6) and adopting $\alpha = e^{-\delta}$, the result of Lemma 4.1 is obtained. □

Lemma 4.2. *If* $m = \frac{\log n + \varpi(n)}{\pi\gamma^2}$, *then*

$$\liminf_{n\to\infty} P_d(n,m) \ge e^{-\varpi}(1-e^{-\varpi}), \quad (7)$$

where $\varpi = \lim_{n\to\infty} \varpi(n)$ *and* $P_d(n, m)$ *is the probability of a disconnected* $\mathcal{G}(n, m)$.

Proof. We first study the case when $m = \frac{\log n + \varpi}{\pi\gamma^2}$ for a fixed ϖ. Let $P^{(\phi)}$, $\phi = 1, 2, \ldots$, denote the probability that a graph $\mathcal{G}(n, m)$ has at least one order-ϕ component. Then, we have

$$\begin{aligned} P_d(n,m) &= P^{(1)}(n,m) \\ &\ge \sum_{i=1}^{n} P(\{i \text{ is the only isolated node in } \mathcal{G}(n,m)\}) \\ &\ge \sum_{i=1}^{n} (P(\{i \text{ is an isolated node in } \mathcal{G}(n,m)\}) \\ &\quad - \sum_{i=1}^{n}\sum_{j\ne i} P(\{i, j \text{ are isolated nodes in } \mathcal{G}(n,m)\}). \end{aligned} \quad (8)$$

Neglecting edge effects, we get

$$P(\{i \text{ is an isolated node in } \mathcal{G}(n,m)\}) = (1-\pi\gamma^2)^m. \quad (9)$$

Whether a sensor node is isolated is independent from other nodes. Thus, we obtain

$$P(\{i, j \text{ are isolated nodes in } \mathcal{G}(n,m)\}) = (1-\pi\gamma^2)^{2m}. \quad (10)$$

Substituting Eqs. (9) and (10) in Eq. (8), we obtain

$$P_d(n,m) \ge n(1-\pi\gamma^2)^m - n(n-1)(1-\pi\gamma^2)^{2m}. \quad (11)$$

Thus, using Lemma 4.1 and the equation of $(1-p) \le e^{-p}$ (Lemma 2.1 in [12]) for any fixed $\alpha < 1$, we have

$$\begin{aligned} P_d(n,m) &\ge \alpha e^{-\varpi} - n(n-1)e^{-2m\pi\gamma^2} \\ &\ge \alpha e^{-\varpi} - (1+\delta)e^{-2\varpi} \end{aligned} \quad (12)$$

for all large n.

Second, we consider the case when ϖ is a function of $\varpi(n)$ with $\lim_{n\to\infty} \varpi(n) = \hat{\varpi}$. For any constant $\delta > 0$ and $\varpi(n) \ge \hat{\varpi} + \delta$, the probability of disconnection is monotone decreasing in ϖ, then we have

$$P_d(n,m) \ge \alpha e^{-\hat{\varpi}+\delta} - (1+\delta)e^{-2(\hat{\varpi}+\delta)}. \quad (13)$$

Since it holds for all $\delta > 0$ and $\alpha < 1$, we take limits and then the result is obtained. □

As an obvious consequence of Lemma 4.2, we have:

Corollary 4.1. *Graph* $\mathcal{G}(n, m)$ *is asymptotically disconnected with positive probability if* $m = \frac{\log n + \varpi(n)}{\pi\gamma^2}$ *and* $\lim_{n\to\infty} \varpi(n) < +\infty$.

Thus, the necessary part of Theorem 4.1 is proved.

4.4.2. Sufficient condition on m

Supposing $m = \kappa\frac{\log n}{\gamma^2}$ CHs with transmission range r, we have the WSN graph $\mathcal{G}(n, \kappa\frac{\log n}{\gamma^2})$. It suffices to show that for some $\kappa > 0$,

$$\lim_{n\to\infty} P\left(\left\{\mathcal{G}\left(n, \kappa\frac{\log n}{\gamma^2}\right) \text{ is connected}\right\}\right) = 1.$$

For the simplicity of the proof for the sufficient condition, we equally divide the area into square cells as shown in Fig. 4, and the number of square cells is $S_n = \lceil \frac{a}{\sqrt{3}r} \rceil^2$, where $\lceil\rceil$ is the ceiling operation. We denote such a tessellation of area as $\mathcal{T}$. In addition, we treat $\frac{a}{\sqrt{3}r}$ as an integer for the sake of clarity of presentation. The error of this approximation can be ignored when a is sufficiently large compared to r, *i.e.*, the fraction of $\frac{a}{\sqrt{3}r}$ can be neglected. Then, we prove the sufficient part of Theorem 4.1 with the following lemma.

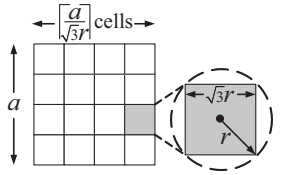


Fig. 4. Tessellation of the square into cells.

Lemma 4.3. *If there is a tessellation of area* $\mathcal{T}$ *and* $m = \kappa\frac{\log n}{\gamma^2}$ *cluster heads randomly deployed in* $\mathcal{T}$, *each cell has at least one cluster head with probability one as* $n \to \infty$.

Proof. Let E_i $(i = 1, 2, \ldots, S_n)$ be the event that a particular CH falls into a particular cell with probability $P(E_i) = \frac{1}{S_n} = (\frac{\sqrt{3}r}{a})^2 = 3\gamma^2$. So the probability that a particular cell has no CH is $\bar{P}(E_i) = (1 - P(E_i))^m$. Then, Q is used to denote the probability that at least one cell is empty. We have

$$Q \le \sum_{i=1}^{S_n} \bar{P}(E_i) = \frac{1}{3\gamma^2}(1-3\gamma^2)^m. \quad (14)$$

Applying $(1-p) \le e^{-p}$ and $m = \kappa\frac{\log n}{\gamma^2}$, we obtain

$$Q \le \frac{1}{3\gamma^2}e^{-3\gamma^2 m} = \frac{1}{3\gamma^2}e^{-3\kappa\log n} = \frac{1}{3\gamma^2 n^{3\kappa}}. \quad (15)$$

When $n \to \infty$, $Q \to 0$ in Eq. (15), *i.e.*, the probability of at least one cell being empty is 0. This result disproves the

Lemma 4.3 that each cell has at least one CH with probability one as $n \to \infty$. □

Thus, the sufficient part of Theorem 4.1 is proved.

4.4.3. Estimation on the bound of C

We have proved that the number of CHs $m = C(\log n) \cdot a^2/r^2$ can connect the WSN, where n, a, r, and stochastic distribution of sensor nodes are known, but C is still a unknown parameter. In order to determine m, we still need to obtain the value of C.

There exist C_1 and C_2, where $0 < C_1 < C_2$, and the optimal number of CHs is no less than $C_1(\log n) \cdot a^2/r^2$ and no more than $C_2(\log n) \cdot a^2/r^2$. According to the criteria that $d(CHs) < 2r$, at least $m = \lceil \frac{a}{2r} \rceil^2$ CHs are needed to guarantee the connectivity. In addition, to reduce the intra-cluster collisions, it is desired that there is only one CH in one cluster, *i.e.*, $d(CHs) > r$. Thus, at most $m = \lceil \frac{a}{r} \rceil^2$ are needed.

We estimate that the optimal number of CHs yields to Guassian distribution where $\lceil \frac{a}{2r} \rceil^2$ and $\lceil \frac{a}{r} \rceil^2$ act as two 3-delta thresholds. Thus, there exists a value, which maximizes the distribution. It is the average of the two 3-delta thresholds. Substituting these values into $m = C(\log n) \cdot a^2/r^2$, we have

$$C_1 = \frac{1}{4\log n}, \quad C_2 = \frac{1}{\log n}.$$

4.4.4. Estimation on the redundancy of CHs

We have derived that m CHs can theoretically make a connected WSN by single-hop clustering. However, in practice, such a number of CHs cannot guarantee the connectivity of entire network because these CHs may not be evenly distributed. Hence, we introduce a redundancy coefficient β ($\beta > 1$) into the determination of CH candidates. More CHs increase the connection probability but meanwhile bring about more collisions. Thus, the value of β depends on both the requirement of connection probability and the number of time slots. We present the total number of CH candidates by:

$$\beta m = \beta C \frac{(\log n) \cdot a^2}{2r^2}. \tag{16}$$

The redundancy coefficient β can be regarded as the number of CHs in one cell in Fig. 4. So the probability of clustering an entire WSN by βm CHs can be simply treated as the probability of organizing one cell by β CHs. It is equivalent to calculate the probability that at least one of the β CHs can broadcast its ID in a certain time slot without collisions. As an example, we conduct a numerical simulation to calculate the probability of successful clustering when the redundancy coefficient β varies from 2 to 10. As shown in Fig. 5, we find that the highest probabilities are always at $\beta = 3$ no matter the number of time slots varying from 3 to 8. Thus, we set empirical $\beta = 3$ in ICP.

All parameters in Eq. (16) are known, so the number of CH candidates can be obtained. Then, the pre-assigned probability $P_{CH} = \beta m/n$ is also obtained. Note if $\beta m/n > 1$, we set $P_{CH} = 1$ because a probability should be no larger than 1.

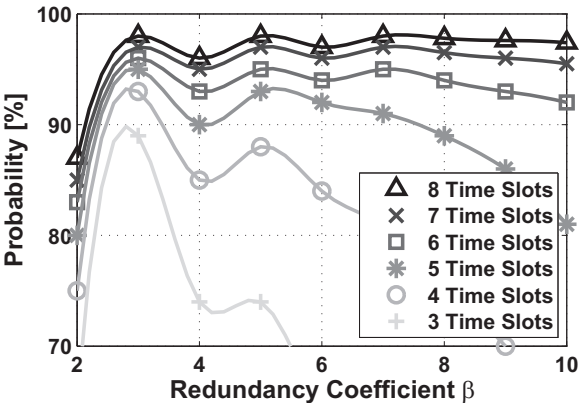


Fig. 5. PDF of clustering results for different β.

4.5. The optimal number of time slots in ICP

In order to address Challenge 2, we derive the optimal number of time slots T for self-organizing every single-hop cluster. This number is determined by two factors. First, the number is the smaller the better for low consumption of time and energy. Second, the number cannot be too small. Otherwise, the collision problem will be severe, further resulting in failure on intra-cluster organization.

A CM/GW can successfully connect into the network if it receives at least one CH's ID message in one certain time slot without collision. The collision happens when concurrent IDs are transmitted, and thus a sensor node cannot decode any message from this collision. Then, we can derive the connection probability of a CM/GW to a certain CH as the following Theorem.

Theorem 4.2. *Given T time slots, one CM/GW, z CH candidates, and $\forall d(\text{CH, CM/GW}) \le r$. Every CH randomly selects one time slot from T and transmits its ID message during this slot. The connection probability P_c, which a CM/GW receives at least one CH's message without collision, is*

$$P_c = 1 - \left(1 - \frac{z}{T} e^{-\frac{z}{T}}\right)^T. \tag{17}$$

Proof. According to Lemma 1 in [19], for every time slot j, $j \in T$, the probability $P_{TX}\{j = 1\}$ that only one CH transmits at this slot is

$$P_{TX}\{j = 1\} = \frac{z}{T}\left(1 - \frac{1}{T}\right)^{z-1} \approx \frac{z}{T} e^{-\frac{z}{T}}. \tag{18}$$

It is easy to obtain that $P_{TX}\{j \neq 1\} = 1 - P_{TX}\{j = 1\}$. And the probability that none of the T time slots having exact one message is $(P_{TX}\{j \neq 1\})^T = (1 - P_{TX}\{j = 1\})^T$. Hence,

$$P_c = 1 - (1 - P_{TX}\{j = 1\})^T. \tag{19}$$

Substituting Eq. (18) into Eq. (19), $P_c = 1 - (1 - \frac{z}{T} e^{-\frac{z}{T}})^T$ in Theorem 4.2 is derived. □

Although the theoretical expectation $\mathbb{E}(z) = \beta$, the actual numbers of CH candidates in different clusters are different (z may be a number around β) due to the stochastic deployment. Using the equation in Theorem 4.2, we can

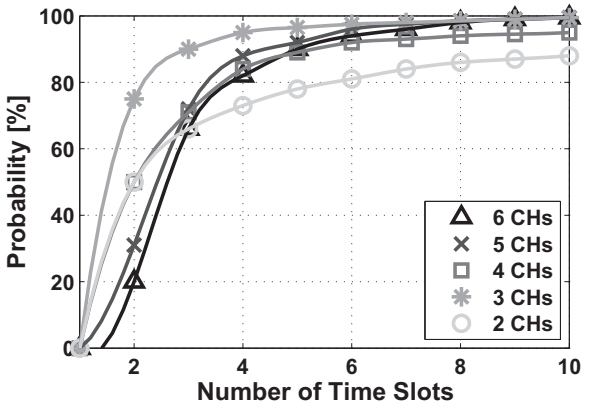


Fig. 6. PDF of a CM/GW being successfully connected with different number of CH candidates.

calculate the connection probability of any given number of z. For example, we assume that z is between 2 and 6, which is around $\beta = 3$. The cumulative probability figure (CDF) of the connection probability is shown in Fig. 6 when the number of time slots varies from $T = 2$ to 10. In Fig. 6, when $T \geq 8$, all curves approach their convergence and most connection probabilities are larger than 90%. In order to guarantee a high connection probability while minimizing the time consumption, we determine $T = 8$ as the optimal number of time slots for the self-organization step in this case. The optimal T in other cases can be found using the same method as above.

Recall that the total number of time slots $x = T + \Delta$, where Δ is the number of time slots for compensation mechanism. The operation of compensation mechanism is to repeat the self-organization several rounds until all isolated nodes connecting into the cluster topology. Hence, the optimal number of time slots for each round is equal to T. Then, $\Delta = \eta T$, where η is the round number. Performance results in Section 7 show that one round $\eta = 1$ of compensation (*i.e.*, $\Delta = T$) is adequate to connect all isolated nodes.

5. ICP algorithm and analysis

In this section, we present the algorithm of ICP and analyze its complexities.

5.1. ICP algorithm

Main procedure. Every senor node is pre-assigned the probability P_{CH} before deployment. After random deployment, every sensor node runs its algorithm including:

- A sensor node becomes a CH candidate with probability P_{CH}, then it runs Procedure 1,
- A sensor node becomes a CM/GW with probability $(1 - P_{CH})$, then it runs Procedure 2,

where P_{CH}, T, and Δ are given according to the derivation in Section 4.

When a sensor node serves as a candidate CH at the beginning, it executes the algorithm as shown in Procedure 1.

Procedure 1 ICH_CH_Algorithm (T, Δ)

```
1: Set k to be a random number [1, T] with seed ω()
   /*become a CH after k or abdicate within k time slots*/
2: Listening from 1st to (k − 1)-th time slots{
3:    if hear no message during these (k − 1) do
4:       be a CH, send ID at k-th slot, quit Procedure 1
5:    if hear one ID message from CH(i) (i ∈ N) do
6:       become a CM connecting CH(i)
7:    if hear multiple ID messages from CHs(i1, i2, ...) do
8:       become a GW connecting CHs(i1, i2, ...) }
   /*if not quit, serve as a CM/GW in the rest time slots*/
9: Listening from (k + 1)-th to T-th time slots
10:   if hear more ID messages do
11:      serve as a GW connecting more CHs
   /*keep listening during the compensation period*/
12:Listening from (T + 1)-th to (T + Δ)-th time slots
13:   execute line 10 to 11
```

Procedure 2 ICP_CM/GW_Algorithm (T, Δ)

```
   /*serve as a CM/GW within T time slots*/
1: Listening from 1st to T-th time slots{
2:    If hear one ID messages from CH(j) do
3:       become a CM connecting CH(j)
4:    If hear multiple ID messages from CHs(j1, j2, ...) do
5:       become a GW connecting CHs(j1, j2, ...) }
   /*compensation mechanism*/
6: If hear no message until T-th time slot do
7: become a CH, do line 1-11 of ICP_CH_Algorithm(Δ, 0)
```

First, it generates a random number k from 1 to T with the seed $\omega()$, where $\omega()$ is a distribution function determined by the present status. For example, at the beginning of the lifetime, all sensor nodes have the same energy, so $\omega(k)$ is set as a uniform distribution function and k has the same probability to be any number from 1 to T. However, at a re-clustering case, assume that a sensor node has only 30% residual energy, its $\omega(k)$ follows the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where the median value μ is $0.7T$ and σ =1 is the unit deviation. This seed $\omega()$ can include any available information about the present status such as residual energy, historical roles of CH/CM/GW, position, number of neighbors, and etc. In this paper, we only consider the residual energy in $\omega()$ as an example. Then, if this node receives no message (collision is also considered as no message) from 1st to $(k - 1)$-th time slot, it broadcasts its ID message and then exits this procedure directly. If it receives messages from other candidate CHs during these $(k - 1)$ time slots, it abdicates to be a CM/GW. As a CM/GW, it should work at the state of listening messages in the rest of the $(T - k)$ time slots and the compensation Δ time slots as well.

When a sensor node serves as a CM/GW at the beginning, it starts the algorithm as shown in Procedure 1. It keeps listening in T time slots. If it receives one CH's ID message, it is determined as a CM connecting to this CH. If it receives multiple messages from different CHs, it becomes a GW connecting to these CHs. If a CM/GW hears no message within T time slots, it indicates that this node is still an isolated one in WSN. Then it will be promoted to be a CH with compensation mechanism and execute the

code from line 1 to 11 in ICP_CH_Algorithm using another Δ time slots.

5.2. Complexity analysis

In order to address Challenge 3, we prove that ICP is lightweight. We analyze the computational complexity, communication overhead, and time cost of ICP to verify it.

Computational complexity: obviously, the computational complexity of ICP either at CH side or at CM/GW side is $O(1)$. Such lightweight algorithm is practical to implement on off-the-shelf sensor networks.

Communication overhead: In conventional methods [37,39], sensor nodes need to transmit their information to neighbors, and thus the communication complexity of the entire WSN is $O(n)$, which is also considered as the order of energy consumption. By contrast, ICP demands that only the CHs broadcast their ID messages, which is $O(m) = O(\log n)$ in Eq. (16). Hence, ICP reduces both communication overhead and energy consumption. According to Eq. (1), the total energy consumption E_{tot} in ICP can be calculated by $\lambda E_{TX} + (xn - \lambda)E_{RX}$, where $x = T + \Delta$ and $\lambda = m$ due to only m CHs transmitting once per CH without ACKs.

Time cost: The number of time slots in ICP is a constant $(T + \Delta)$ as depicted in Procedures 1 and 2. By contrast, the time cost in conventional methods such as LEACH [14] and HEED [37] is proportional to the density. Intuitively, ICP is much faster.

5.3. Discussion

Acknowledgment: the ACK mechanism is widely adopted in wireless communication to ensure that a message is successfully received. However, for low consumption, ICP exempts the ACK mechanism. The reason is that even a sensor node in ICP does not receive any message, it would upgrade to be a CH by the compensation mechanism. Hence, it is unnecessary to waste the time and energy on ACKs.

Load balance: For the fairness of energy consumption on sensor nodes (CHs usually consume more energy than CM/GWs), a load-balanced clustering is usually considered in existing methods [14]. In ICP, since every sensor node has the same probability to be a CH and there are multiple re-clustering processes during the lifetime, the expectation of energy consumption of every node is the same. In addition, some residual energy methods [37] can easily transplant into the seed $\omega()$ for generating k to achieve load balance. The performance of ICP's load balance will be shown in Section 7.

Fault tolerance: When a sensor node fails, ICP cannot react immediately. However, in the next re-clustering process, ICP will produce a connected WSN without needing to detect the failed nodes. Even if there are some isolated nodes due to failure, the compensation mechanism can upgrade these isolated nodes to be CHs to guarantee the connectivity.

Extensibility: Although this work mainly focuses on stochastically deployed WSNs, ICP can extend to determined deployed WSNs and nonuniform distribution of nodes as long as the local density (number of neighbors within transmission range) is known. For any sensor node, if the number of neighbors is θ, its pre-assigned probability is $P_{CH} = \beta/\theta$.

6. Implementation and testbed based experiment

To verify the feasibility and characterize the efficiency of ICP in practical WSNs, we implement ICP on a TelosB based WSN testbed and evaluate its performance. We first present the experimental methodology and then measure the results.

6.1. Experimental methodology

Testbed. We perform our experiments on the real WSN testbed, *NetEye* [17]. As shown in Fig. 7 TelosB nodes are deployed in an indoor environment, in which every two closest neighbors are separated by 2 feet. On each sensor node, a 3 dB signal attenuator and a 2.45 GHz monopole antenna are installed. In our experiments, we set the radio transmission power to be −25 dBm (*i.e.*, power level 3 in TinyOS). And we use the default MAC protocol provided in TinyOS 2.x.

Topology. We select 63 sensor nodes in NetEye to form a 7 by 9 grid distribution. We set another sensor node to serve as a base station. It issues a command beacon (with high transmission power), which can cover all 63 nodes, to start a new clustering process synchronously. To gain statistical results, we repeat every clustering method for 100 runs.

Compared methods. To demonstrate the performance improvement of ICP over existing methods, we comparatively study the following methods.

- *LEACH* [14]: the most classic energy-efficient clustering method in WSN research community;
- *ECDS* [4]: the state-of-the-art energy-efficient clustering method, which selects CHs using a constrained dominating set approach;
- *DSBCA* [25]: the state-of-the-art load-balanced clustering method, which first calculates the clusters by distance and distribution, then votes CHs according to number of neighbors and residual energy;
- *FT-EEC* [18]: the state-of-the-art fault-tolerant clustering method, which re-clusters the network after detecting a failure of sensor node;

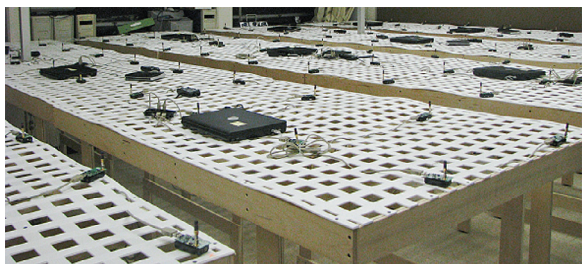


Fig. 7. NetEye testbed.

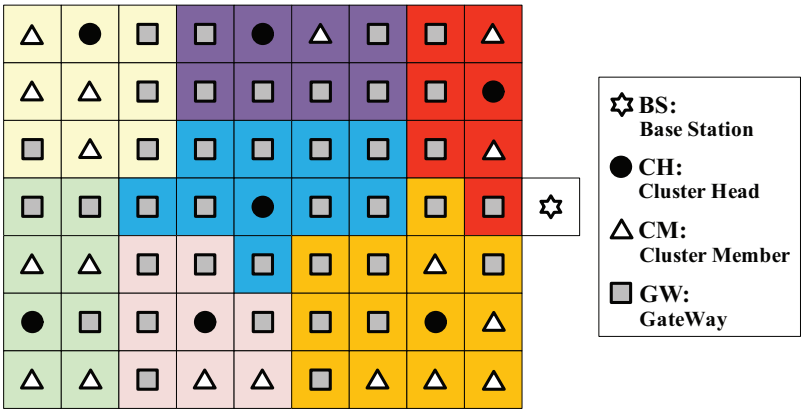


Fig. 8. An example of ICP result.

- *ICP*: Our proposed clustering method, whose CHs are locally determined by pre-assigned probability and the ACK mechanism is exempted.

Performance metrics. We evaluate above clustering methods based on the following metrics:

- *Time consumption of clustering*: the total time consumption T_{tot} to accomplish the clustering process. This is one main metric, which we consider in this work. An instantaneous clustering is desired.
- *Transmission amount*: the total number of transmissions λ required during a clustering process. This is another main metric in this work. We use this metric to represent the total energy consumption E_{tot} because of Eq. (1). An low-energy clustering is desired.
- *Isolation ratio*: the percentage of sensor nodes, which are not connected to any CH by the end of the clustering process. This metric indicates the connectivity performance of a clustering method. The lower this isolation ratio is, the higher connectivity is achieved.

6.2. Experiment results

The first experiment is to validate the feasibility of ICP. An example of the clustering result by ICP is illustrated in Fig. 8. In this illustration, we use different colors to distinguish clusters. For a GW, since it connects multiple CHs, we set its color to be the same as the CH, from which the received RSSI is the strongest. We find in Fig. 8 that there is no isolated node after ICP. All sensor nodes are organized into 7 single-hop clusters, where the sizes of these clusters are relatively balanced (from 7 to 13 nodes). More experiments show the similar results as in this illustration. Hence, the feasibility of ICP on clustering sensor nodes into a connected WSN is verified by our real WSN testbed.

We then study the time comparison among five clustering methods. The average time consumption and its standard deviation are plotted in Fig 9. It is shown that ICP yields the smallest time consumption among all methods, which is around 120 ms. On the contrary, the other four methods need 260 to 790 ms to accomplish the clustering processes. ICP reduces 55% time consumption compared with the second fastest one because it requires neither CH voting nor ACKs. LEACH and FT-EEC consumes a few larger time because both of them adopt the pre-assign method to generate CHs. However, they still require time for the ACK mechanism. ECDS and DSBCA incurs much larger duration due to no optimization on CH voting or ACKs. Furthermore, DSBCA requires an iteration process to guarantee the connectivity, leading to a 6.5x time consumption of ICP.

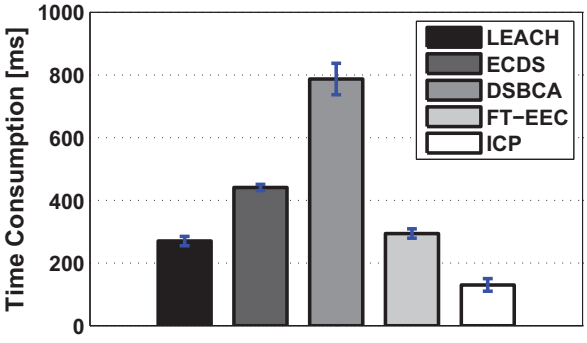


Fig. 9. Experiment of time consumption.

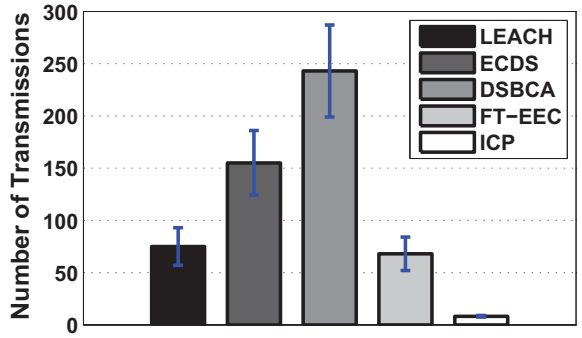


Fig. 10. Experiment of transmission amount.

The transmission amounts of five methods are compared in Fig. 10. We see that ICP significantly reduces the total number of transmissions, *i.e.*, about 89% less than the second best method. This is because ICP only allows the CH candidates to contend and transmit messages. In this way, only a very small portion of nodes, the final

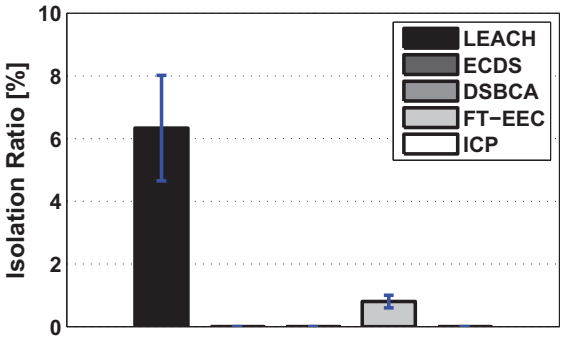


Fig. 11. Experiment of isolation ratio.

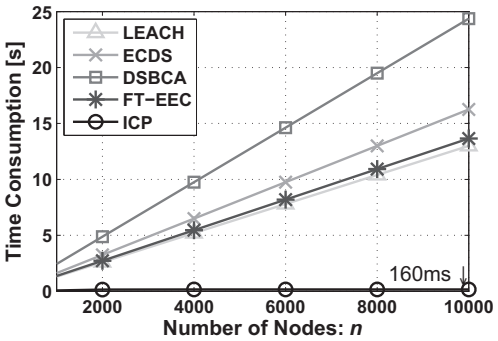


Fig. 12. Simulation of time consumption.

$m = 7$ or 8 CHs, can obtain the opportunity to transmit one message per CH during the clustering process. On the contrary, every sensor node transmits at least one message, either an ID message from CH or an ACK message from CM/GW, in LEACH and FT-EEC. Their number of transmissions is around 63 with some unavoidable retransmissions due to CSMA/CA. ECDS requires nearly 2x transmission amount of LEACH because every sensor node in ECDS needs to transmit at least 2 messages, one for voting and another for ID/ACK. Furthermore, DSBCA incurs the highest transmission amount because its clustering process requires the participation of all sensor nodes, each of which needs to send multiple messages for voting and iteration.

In Fig 11, we show the average isolation ratio of five clustering methods. We find that ICP, ECDS, and DSBCA are always zero on this metric, so all sensor nodes are connected to the network using these three methods. Benefitting from the design of compensation mechanism, ICP achieves this zero isolation with low consumption. On the contrary, although ECDS and DSBCA achieves the zero isolation, they require a larger amount of transmissions as shown in Fig. 10. We find in Fig 11 that LEACH and FT-EEC yield an average of 6.5% and 0.8% nodes not connecting to any cluster. The reason is not only because the pre-assigned CHs are unevenly distributed, but also because the concurrent ACKs significantly increases the probability of transmission collisions. As a result, some nodes in LEACH and FT-EEC have to stay isolated.

Experiment results from Figs. 9–11 demonstrate the efficiency and efficacy of ICP. Compared with the classic and state-of-the-art methods, ICP leverages an time-efficient and energy-efficient design requiring neither CH voting nor ACK. Therefore, ICP is significantly advantageous over other clustering methods in terms of time consumption, transmission amount, and isolation ratio.

7. Simulation

Experiments in the NetEye testbed are limited by its scale. In order to study the performance of ICP in large-scale WSNs, as a supplement to our experiments, we conduct extensive simulations to further evaluate ICP in this section.

In addition, the simulation settings include that $n = 1,000$ to 10,000 sensor nodes are stochastically distributed in an 800×800 square area and the transmission range of each node varies from $d = 120$ to $l = 200$ using the quasi disk model [23]. In addition, the number of time slots of ICP is set as the theoretical results in Section 4.5, so $x = T + \Delta = 16$. The redundancy coefficient $\beta = 3$ as the analysis in Section 4.4.4. The seed $\omega()$ discussed in Section 5 is set to follow the uniform distribution. All simulation results are the average of 1000 runs.

7.1. Simulation results

As the comparison in experiments, our simulations also compare the performance of ICP with LEACH [14], ECDS [4], DSBCA [25], and FT-EEC [18].

For understanding the consumption clearly, we set the units for time and energy in the simulations. First, one time slot is set as $t = 10$ms. Since the bit rate of IEEE 802.15.4 protocol for WSN transmissions is 250kbps and the ID/ACK message is usually $<$ 100 Bytes, 10ms is enough to transmit the ID/ACK message. Second, we set that transmitting one message in one time slot consumes 0.21mW, and receiving in one time slot consumes 0.15mW. The reason is because the typical Mica node consumes 21mW on transmission and 15mW on reception [15] per second.

We study the time consumption T_{tot} of five clustering methods in Fig. 12, where the number of total sensor nodes n varies from 1,000 to 10,000. It can be found that ICP keeps a constant time consumption, which is $T_{tot} = xt = 16 \times 10\text{ms} = 160\text{ms}$ independent to the vary of n. However, the durations of all the other methods linearly increase with n. For instance, when $n = 10,000$, LEACH, FT-EEC and ECDS require about 15s, and DSBCA requires up to 25s. ICP can maintain such a constant duration because only m CHs need to transmit, where m is at $O(\log n)$ level. In our simulation scenario, the number of CHs ranges from m=16 to 28 (we will discuss about it in detail in Fig. 15), so 160ms is enough for these m CHs to contend. Although the other four methods can also cluster in a parallel manner, the number of nodes in one cluster (*i.e.*, density) is increased with n. All these intra-cluster nodes need to transmit ID/ACK messages, and thus leads to the increase of time consumption.

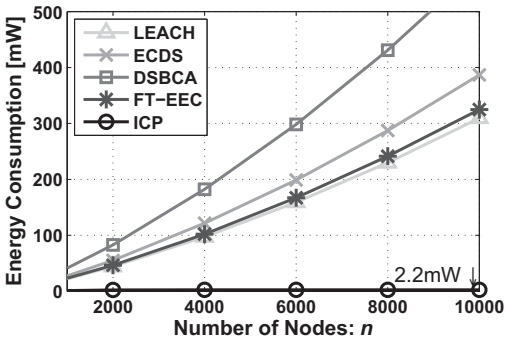


Fig. 13. Simulation of transmission amount.

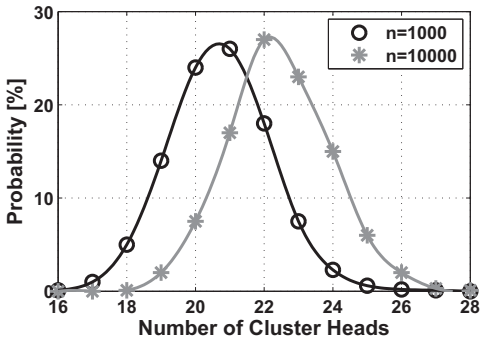


Fig. 15. PDF of CHs in different n.

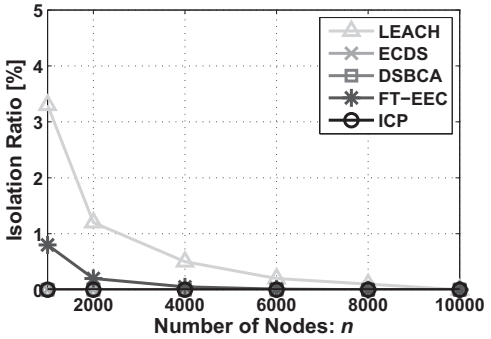


Fig. 14. Simulation of isolation ratio.

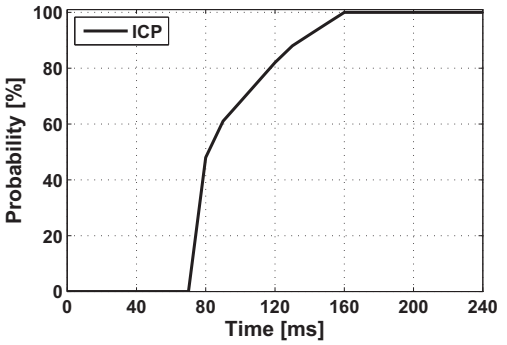


Fig. 16. CDF of time consumption.

The simulation results of energy consumption E_{tot} in different methods are shown in Fig. 13. First, we find that the trends of energy consumption curves in Fig. 13 are close to the time consumption lines in Fig. 12. This observation indicates that the energy consumption is mainly dominated by the time consumption. Second, the energy consumption curves are not perfectly linear but slight upward. The reason is that LEACH, ECDS, DSBCA, and FT-EEC consume extra energy on retransmission due to mass of intra-cluster collisions in high density. Third, ICP consumes only 2.2mW energy when $n = 10,000$, definitely outperforming the other methods, which consumes from 300 to 580mW energy.

We show the comparison of isolation ratio in Fig. 14. The simulation results of ICP, ECDS, and DSBCA on this metric are always zero, so all nodes are successfully connected into the clusters, which are the same as the experiment results. Then, we find that the isolation ratios of LEACH and FT-EEC are reduced with the increase of n. The reason is that the effect of uneven distribution of CHs is mitigated when the pre-assigned CH candidates become dense in the network. As shown in Fig. 14, if there are more than 8,000 sensor nodes scattered in the 800 $\times$ 800 area, the isolation ratios of all clustering methods approach zero.

In summary, not only in experiments but also in simulations, ICP achieves an instantaneous clustering with very low energy consumption, significantly outperforming existing clustering methods.

7.2. Design insights

In order to further understand the advantages of ICP design, we conduct the following simulations.

First, we measure the impact of the total number of sensor nodes n on the number of CHs m. Fig. 15 is a probability distribution figure (PDF) showing the number of CHs generated by ICP, when $n = 1,000$ and 10,000 respectively. We find that both probability curves nearly follow the Guassian distribution. Their median values are closed to each other, *i.e.*, 21 and 22 respectively, which fits the result that m is $O(\log n)$. Moreover, the $n = 1,000$ curve ranges from 16 to 26 and the $n = 10,000$ curve ranges from 18 to 28, which can be considered as bounded by C_1 and C_2. This result is also in consistent with the theoretical result given in Section 4.4.

Then, we investigate the impact of time slot setting. Fig. 16 is a cumulative distribution figure (CDF), which shows the probability distribution of time consumption in ICP. From the statistical results in our simulation, in about 45% cases, the clustering is accomplished within 80ms; and 100% cases, the clustering is completed within 160ms. Since every time slot is $t = 10$ms long, 160/10=16 time slots are adequate for ICP. Because of this empirical result, we set the number of time slots to be $x = T + \Delta = 8 + 8 = 16$. This setting is also verified by our experiments.

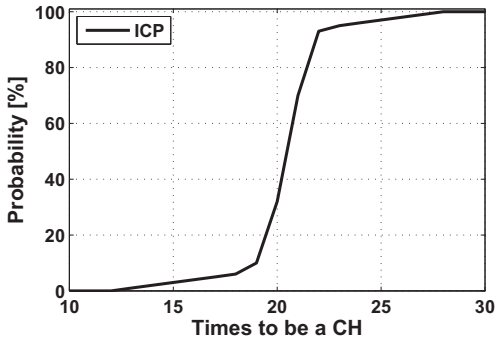


Fig. 17. Simulation of load balance.

Finally, we study the load balance feature of ICP. Since CHs usually consume more energy than CM/GWs, to prolong the lifetime of a WSN, a load-balanced clustering is desired that every sensor node serves as a CH with the same probability. We conduct the simulation with $n = 1,000$ nodes and re-cluster these nodes 1,000 times. The CDF of times to be a CH is shown in Fig. 17. We find that less than 10% nodes serve as CHs less than 19 times; and less than 91% nodes serve as CHs less than 22 times. This result indicates that about 81% nodes serve as CHs with 20, 21, or 22 times. We summarize that most nodes have the same probability to be CHs, and thus ICP can provide a load-balanced clustering.

8. Conclusion

Time and energy consumption are two fundamental metrics to evaluate the clustering in WSNs. In this paper, we present a parallel clustering method, namely ICP, to reduce both the time and energy consumption. The proposed ICP benefits from two key principles. First, the cluster heads are locally determined by the pre-assigned probability instead of voting. Second, the transmission load and the duration of clustering are minimized as long as the connectivity could be achieved. In this way, retransmissions and ACKs are removed. Moreover, ICP is a lightweight and fully distributed method. We implement ICP in the NetEye testbed using 64 TelosB nodes and conduct extensive simulations for large-scale WSNs. Results from experiments and simulations demonstrate that ICP significantly outperforms existing methods in terms of time and energy while ensuring the connectivity, load balance, and fault tolerance. ICP is promising in practical WSN applications, especially for mission-critical applications.

Acknowledgment

This research was supported in part by the NSERC Discovery Grant 341823, NSERC Collaborative Research and Development Grant CRDPJ418713, Canada Foundation for Innovation (CFI)'s John R. Evans Leaders Fund 23090, NSFC Grant 61373155, 61303202, and China Postdoctoral Science Foundation Grant 2014M560334, 2015T80433.

References

[1] A.A. Abbasi, M. Younis, A survey on clustering algorithms for wireless sensor networks, Elsevier Comput. Commun. 30 (14) (2007) 2826–2841.
[2] M.M. Afsar, M.H. Tayarani-N, Clustering in sensor networks: A literature survey, Elsevier J. Netw. Comput. Appl. 198 (2014) C226.
[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Elsevier Comput. Netw. 38 (4) (2002) 393–422.
[4] J. Albath, M. Thakur, S. Madria, Energy constraint clustering algorithms for wireless sensor networks, Elsevier Ad Hoc Netw. 11 (8) (2013) 2512–2525.
[5] S. Bandyopadhyay, E.J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings of the IEEE INFOCOM, 2003.
[6] M. Chatterjee, S.K. Das, D. Turgut, WCA: A weighted clustering algorithm for mobile ad hoc networks, Springer Clust. Comput. 5 (2) (2002) 193–204.
[7] A. Chen, S. Kumar, T.H. Lai, Local barrier coverage in wireless sensor networks, IEEE Trans. Mob. Comput. 9 (4) (2010) 491–504.
[8] M. Di Ventra, Y.V. Pershin, The parallel approach, Nat. Phys. 9 (4) (2013) 200–202.
[9] J. Elson, K. Römer, Wireless sensor networks: A new regime for time synchronization, ACM SIGCOMM Comput. Commun. Rev. 33 (1) (2003) 149–154.
[10] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, A. Woo, The collection tree protocol, TinyOS TEP 123 (2006) 2.
[11] Y. Gao, W. Dong, L. Deng, C. Chen, J. Bu, Cope: Improving energy efficiency with coded preambles in low power sensor networks, Trans. Ind. Inf. (2015).
[12] P. Gupta, P.R. Kumar, Critical power for asymptotic connectivity in wireless networks, in: Stochastic Analysis, Control, Optimization and Applications, 1999.
[13] L. He, Z. Yang, J. Pan, L. Cai, J. Xu, Y. Gu, Evaluating service disciplines for on-demand mobile data collection in sensor networks, IEEE Trans. Mob. Comput. 13 (4) (2014) 797–810.
[14] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wirel. Commun. 1 (4) (2002) 660–670.
[15] J.L. Hill, D.E. Culler, Mica: A wireless platform for deeply embedded networks, IEEE Micro 22 (6) (2002) 12–24.
[16] Y.-K. Huang, A.-C. Pang, P.-C. Hsiu, W. Zhuang, P. Liu, Distributed throughput optimization for zigbee cluster-tree networks, IEEE Trans. Parallel Distrib. Syst. 23 (3) (2012) 513–520.
[17] X. Ju, H. Zhang, D. Sakamuri, NetEye: a user-centered wireless sensor network testbed for high-fidelity, robust experimentation, 25, Wiley Int. J. Commun. Syst., 2012, pp. 1213–1229.
[18] L. Karim, N. Nasser, T. Sheltami, A fault-tolerant energy-efficient clustering protocol of a wireless sensor network, Wiley Wirel. Commun. Mob. Comput. 14 (2) (2014) 175–185.
[19] M. Kodialam, T. Nandagopal, Fast and reliable estimation schemes in RFID systems, in: Proceedings of the ACM MOBICOM, 2006.
[20] L. Kong, M. Xia, X.-Y. Liu, G. Chen, Y. Gu, M.-Y. Wu, X. Liu, Data loss and reconstruction in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 25 (11) (2014a) 2818–2828.
[21] L. Kong, M. Zhao, X.-Y. Liu, J. Lu, Y. Liu, M.-Y. Wu, W. Shu, Surface coverage in sensor networks, IEEE Trans. Parallel Distrib. Syst. 25 (1) (2014b) 234–243.
[22] F. Kuhn, T. Moscibroda, R. Wattenhofer, Initializing newly deployed ad hoc and sensor networks, in: Proceedings of the ACM MOBICOM, 2004.
[23] F. Kuhn, R. Wattenhofer, A. Zollinger, Ad hoc networks beyond unit disk graphs, Wirel. Netw. 14 (5) (2008) 715–729.
[24] P. Kuila, P.K. Jana, Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach, Eng. Appl. Artif. Intell. 33 (2014) 127–140.
[25] Y. Liao, H. Qi, W. Li, Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks, IEEE Sens. J. 13 (5) (2013) 1498–1506.
[26] C.R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE J. Sel. Areas Commun. 15 (7) (1997) 1265–1275.
[27] X.-Y. Liu, S. Aeron, V. Aggarwal, X. Wang, M.-Y. Wu, Adaptive sampling of rf fingerprints for fine-grained indoor localization, Trans. Mob. Comput. (2015).to appear in

[28] X.-Y. Liu, K.-L. Wu, Y. Zhu, L. Kong, M.-Y. Wu, Mobility increases the surface coverage of distributed sensor networks, Elsevier Comput. Netw. 57 (11) (2013) 2348–2363.
[29] Y. Liu, Q. Zhang, L.M. Ni, Opportunity-based topology control in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 21 (3) (2010) 405–416.
[30] J. Niu, L. Cheng, Y. Gu, L. Shu, S.K. Das, R3e: Reliable reactive routing enhancement for wireless sensor networks, IEEE Trans. Ind. Inf. 10 (1) (2014) 784–794.
[31] Y.K. Tan, T.P. Huynh, Z. Wang, Smart personal sensor network control for energy saving in dc grid powered led lighting system, IEEE Trans. Smart Grid 4 (2) (2013) 669–676.
[32] P.-J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, in: Proceedings of the IEEE INFOCOM, 2002.
[33] Q. Xiang, H. Zhang, J. Wang, G. Xing, S. Lin, X. Liu, On optimal diversity in network-coding-based routing in wireless networks, in: Proceedings of the IEEE INFOCOM, 2015.
[34] Q. Xiang, H. Zhang, J. Xu, X. Liu, L.J. Rittle, When in-network processing meets time: Complexity and effects of joint optimization in wireless sensor networks, IEEE Trans. Mob. Comput. 10 (10) (2011) 1488–1502.
[35] H. Xiong, D. Zhang, L. Wang, J.P. Gibson, J. Zhu, Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants, ACM Trans. Intell. Syst. Technol. 6 (3) (2015) 39.
[36] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: Proceedings of the IEEE INFOCOM, 2002.
[37] O. Younis, S. Fahmy, Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Trans. Mob. Comput. 3 (4) (2004) 366–379.
[38] J.Y. Yu, P.H.J. Chong, A survey of clustering schemes for mobile ad hoc networks, IEEE Commun. Surv. Tutor. 7 (1–4) (2005) 32–48.
[39] H. Zhang, A. Arora, Gs3: scalable self-configuration and self-healing in wireless sensor networks, Elsevier Comput. Netw. 43 (4) (2003) 459–480.
[40] J. Zhao, R. Govindan, Understanding packet delivery performance in dense wireless sensor networks, in: Proceedings of the ACM SenSys, 2003.

Linghe Kong is currently a tenure-track assistant professor at Shanghai Jiao Tong University. From 2014 to 2015, he was a Postdoctoral Fellow in the School of Computer Science of McGill University. He received his Ph.D. degree in Computer Science from Shanghai Jiao Tong University 2012, Dipl. Ing. degree in Telecommunication from TELECOM SudParis 2007, and B. E. degree in Automation from Xidian University 2005. His research interests include wireless sensor networks, mobile computing, and RFID.

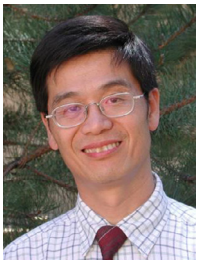
Qiao Xiang is a postdoctoral associate with Department of Computer Science of Yale University and Tongji University. From 2014 to 2015, he was a Postdoctoral Fellow in the School of Computer Science of McGill University. He received his Master and Ph.D. Degrees in Computer Science from Wayne State University in 2012 and 2014, respectively. Before that, he received his Bachelor Degree in Engineering and Bachelor Degree in Economics from Nankai University, Tianjin, China, in 2007. His research interests include wireless cyber physical systems, vehicular networks, wireless sensor networks, smart grid and network economics.

Xue Liu received the B.S. degree in mathematics and the MS degree in automatic control both from Tsinghua University, China, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 2006. He is an associate professor in the School of Computer Science at McGill University. His research interests include computer networks and communications, smart grid, real-time and embedded systems, cyber-physical systems, data centers, and software reliability.

Xiao-Yang Liu received his B.E. Degree in computer science and technology from the Huazhong University of Science and Technology, Wuhan, China, in 2010. He is now a Ph.D. candidate in the Department of Computer at the Shanghai Jiao Tong University. His research interests include Internet of Things, Wireless Networks and Cybersecurity.

Xiaofeng Gao received the B.S. degree in Mathematics from Nankai Univ., China, the M.S. degree in Operations Research from Tsinghua Univ., China, and the Ph.D. degree from Univ. of Texas at Dallas, USA. She is currently an associate professor at Department of Computer Science and Engineering, Shanghai Jiao Tong Univ., China. Her research interests include data engineering, data center, and combinatorial optimization in networks.

Guihai Chen earned his B.S. degree from Nanjing University in 1984, M.E. degree from Southeast University in 1987, and Ph.D. degree from the University of Hong Kong in 1997. He is a distinguished professor of Shanghai Jiaotong University, China. He had been invited as a visiting professor by many universities including Kyushu Institute of Technology, Japan in 1998, University of Queensland, Australia in 2000, and Wayne State University, USA during September 2001 to August 2003. He has a wide range of research interests with focus on sensor network, peer-to-peer computing, high performance computer architecture and combinatorics.



Min-You Wu received the Ph.D. degree from Santa Clara University, Santa Clara, CA in 1984. He is a professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University and a research professor with the University of New Mexico. His research interests include grid computing, wireless networks, sensor networks, multimedia networking, parallel and distributed systems, and compilers for parallel computers.