

An Energy-Efficient Adaptive Overlapping Clustering Method for Dynamic Continuous Monitoring in WSNs

Yuan Hu, Yugang Niu, James Lam, *Fellow, IEEE*, and Zhan Shu, *Member, IEEE*

Abstract—Clustering is a key technique to improve energy efficiency in wireless sensor networks (WSNs). In continuous monitoring applications, the clusters should be formed dynamically according to the event development for energy-efficient data gathering. In this paper, an energy-efficient adaptive overlapping clustering method (EEAOC) is proposed in WSNs for continuous monitoring applications. In EEAOC, a 2-logical-coverage overlapping clustering topology is established such that the adjacent sensors in the event area can be grouped into the same cluster for data fusion and the cluster migration operation can be processed without changing the overlapping structure among clusters. Moreover, to further reduce energy consumption, a hybrid data reporting strategy that switches between time-driven and event-driven schemes is introduced based on the QoS requirements in continuous monitoring applications. Simulation results show that EEAOC achieves a longer network lifetime cycle.

Index Terms—Clustering algorithm, event-driven clustering, event tracking, wireless sensor networks

I. INTRODUCTION

WIRELESS sensor networks (WSNs) consist of a large amount of sensors to perform distributed sensing tasks, which are widely used in health-care monitoring, wildlife tracking, environmental monitoring and industrial detection, for instance [1]–[4]. As sensors in WSNs are powered by battery and are usually deployed over a large unpopulated or sometimes hostile region, it is impossible to recharge or replace them in most cases [1]–[6]. In addition, the characteristics of multi-hop and many-to-one communication mode often result in unbalanced energy depletion [6] [7]. Nodes at hotspots (near the sink) tend to exhaust the battery energy quickly, leading to the 'Funnel Effect' [8]. No data can be transmitted to the sink

even 90% of total initial energy of the network is left [9]. Hence, it is a major concern of how to reduce and balance energy consumption in WSNs [6]–[11].

A. Clustering in WSNs

Clustering is an efficient approach to prolong the network lifetime for large-scale WSNs [12]. In a cluster-based WSN, the cluster head (CH) is responsible for collecting data from the cluster members (CMs) in each cluster. Then the CH aggregates the collected data and sent it to the sink directly or via relay nodes. The clustering methods can prolong network lifetime, balance energy consumption and provide scalability of the network [12].

In general, the operations in clustering protocols are divided into four steps: information collection, CH selection, cluster formation and data transmission [12] [13]. CHs consume much more energy than ordinary nodes due to extra data collection and fusion tasks. Hence, many criteria related to, for example, residual energy, node density, distance and location have been proposed for selecting proper CHs to balance the energy consumption [14].

B. Types of Application in WSNs

WSNs are application-oriented networks, and their typical applications can be derived according to the monitored physical process including event-based applications, continuous monitoring applications, query-driven applications, and hybrid applications [15]. In event-based applications, the reading in a particular area will be used only when an event is detected. In query-driven applications, each node transmits its gathering data to the sink when a query message is received from the sink. In event-based and query-driven applications, the network load is light most of the time, and will become heavy only when the data reporting condition is triggered [16] [17]. On the contrary, in continuous monitoring applications, sensors send data to the sink periodically [18]. In hybrid applications, burst data may be generated in addition to periodically sensed data.

C. Motivations

In some continuous monitoring applications, the detected event may spread to a larger region or/and may change its physical properties [19]. For instance, in the forest fire alarm system, fire may spread from one region to another, or extend to a larger region. In a safety supervision system of a chemical factory, the leaking gas cloud may dispense to the atmosphere.

This work is supported by National Natural Science Foundation (NNSF) from China (61273073, 61374107), HKU CRCG 201411159139, and Cheung Kong Scholars Programme.

Yugang Niu and Yuan Hu are with Key Laboratory of Advanced Control and Optimization for Chemical Process of Ministry of Education, East China University of Science and Technology, Shanghai, China (e-mail: acniuyl@ecust.edu.cn; yuanhu@mail.ecust.edu.cn).

James Lam is with Department of Mechanical Engineering, University of Hong Kong, Hong Kong (e-mail: james.lam@hku.hk).

Zhan Shu is with Electro-Mechanical Engineering Group, Faculty of Engineering and the Environment, University of Southampton, UK (e-mail: z.shu@soton.ac.uk).

In a rainstorm and flood warning system, the flooded area may change according to seasonal and climate variations. In those cases, the detected events change gradually with their size and location.

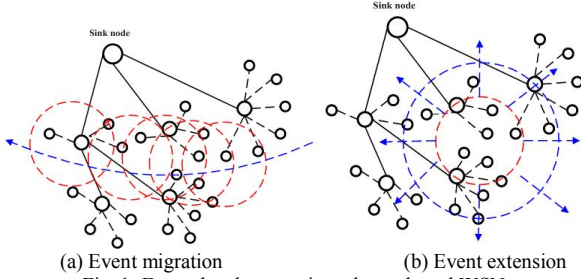


Fig. 1: Event development in a cluster-based WSN

In traditional clustering schemes, clusters are formed without considering event development. Consequently, adjacent nodes in the same event area may belong to different clusters, while nodes outside the event area may be forced to send data to CHs. As shown in Fig. 1(a)–(b), in a cluster-based WSN, an event occurs in a small region and the event area moves or extends gradually, the event area will be covered by different group of sensors as time progresses. In some data gathering rounds, improper cluster formation may appear due to event development. As an example shown in Fig. 2, the relevant data sensed from the same event area is transmitted to three different CHs for fusion, and nodes that do not detect the event in these three clusters also transmit their sensed data to CHs.

Furthermore, most of the existing clustering schemes dismiss all clusters to re-format new ones after certain transmission rounds. This periodic global cluster re-adjusting scheme leads to huge communication overhead. Moreover, most event-driven clustering schemes take the whole event area as one cluster for data gathering. The assumption is not practical when the event is detected in a large area. Therefore, how to design more energy-efficient data reporting schemes in cluster-based WSNs for dynamic continuous monitoring applications still needs to be further investigated.

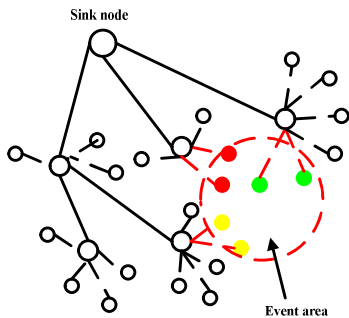


Fig. 2: Improper cluster formation

Motivated by above discussion, an energy-efficient adaptive overlapping clustering method (EEAOC) is proposed in this paper, which focuses on continuous monitoring with possibly event fluctuation. The key features of EEAOC algorithm are summarized as follows:

(1) An energy-efficient 2-logical-coverage overlapping clustering is designed to adapt to event fluctuation, which ensures that data collected by adjacent sensors in the event area

are sent to the same CHs for fusion. Once the event spreads to neighboring regions, a collaborative CH re-adjustment and cluster migration technique is used to ensure effective cluster formation associated with a 2-logical overlapping clustering. To balance the intra-cluster energy consumption, CH rotation is operated without changing the overlapping structure among clusters when the CHs in the event area have depleted their energy.

(2) Aiming at prolonging the network lifetime of WSNs for continuous monitoring applications under some guaranteed monitoring accuracy, a hybrid data reporting strategy that switches between time-driven and event-driven schemes is introduced.

The rest of this paper is organized as follows. In Section 2, related works are introduced, and critical problems in these schemes are described. In Section 3, we present the network model and some definitions. The details of our protocol are proposed in Section 4. In Section 5, we analyze the performance of EEAOC and simulate it in comparison with existing protocols. Finally, the conclusion is given in Section 6.

II. RELATED WORKS

In this section, we classify the main commonly used clustering schemes based on intra-cluster topology (single-hop, multi-hop) and data reporting schemes (time-driven, event-driven).

A. Single-hop and Multi-hop Clustering Schemes

According to the intra-cluster topology, clustering techniques can be classified into single-hop and multi-hop approaches [13]. Single-hop clustering schemes as in [20]–[22] assign CMs to communicate with their corresponding CHs directly. LEACH [20] is a classic single-hop clustering protocol that assigns CHs according to a predefined probability ensuring that all nodes in the network become CHs with the same chance. Each CM sends data to its CH periodically after cluster formation. In DSBCA [21], the network is divided into unequal single-hop clusters for load-balancing based on the node density and the distance from the sink. A dynamic single-hop clustering DECS [22] was proposed for heterogeneous WSNs, wherein CHs are selected through the forecasted network residual energy.

In large-scale WSNs, multi-hop intra-cluster communication model is adopted to avoid long distance transmission. In multi-hop clustering schemes as in [23]–[25], there are relay nodes in each cluster to maintain multi-hop sensors-to-CHs connectivity. The hierarchical cluster-based routing algorithm in [23] constructs a multi-hop routing tree for inter-cluster communication, in which multiple criteria are used to select CHs and an adjustment degree is set to modify intra-cluster and inter-cluster energy consumption. HEHC [24] is a heterogeneous-aware multi-level clustering scheme, which adds extra cluster layers and super CHs for obtaining better energy utilization. The objective of DEECIC [25] is to preserve the coverage and achieve efficient energy consumption by adopting 2-hop intra-cluster communication mode. Generally,

multi-hop clustering schemes are more complicated for building and maintaining the network topology.

However, both single-hop and multi-hop clusters are formed independent of event developments, which are not suitable for dynamic continuous monitoring applications. As shown in Fig. 3, an event is detected in a small region in the beginning shown as Area 1. All the sensors in the event area can be grouped into one single-hop cluster. Gradually, the event may spread to a larger area shown as Area 2. The sensors in Area 2 can be grouped into one multi-hop cluster. Finally, the event may spread to an even larger region shown as Area 3. It is inappropriate to group the sensors in Area 3 into just one cluster. As a result, the previous one multi-hop cluster formation should be dismissed and the sensors in Area 3 should be grouped to multiple clusters. Hence, a dynamic clustering scheme should be designed to adapt to the event development.

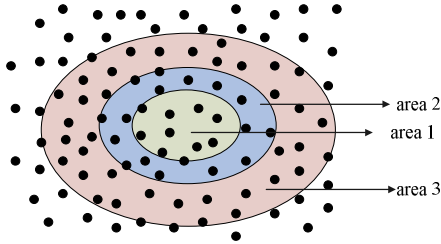


Fig. 3 Event area development

B. Time-driven and Event-driven Clustering Schemes

The aforementioned clustering schemes in [20]–[25] are based on time-driven data reporting schemes. The nodes sense and send data to the sink periodically. However, in some applications, the event may not be detected in the whole area. Event-driven clustering protocols are designed to further improve energy efficiency, in which data collection depends on the occurrence of events [13]. The TEEN [26] and APTEEN [27] protocols utilize two thresholds (Hard Threshold and Soft Threshold) to reduce the number of transmission messages. Hard Threshold is a sensed attribute for the node to send data to its CH while Soft Threshold is a small change of the sensed attribute that triggers the node to switch on the transmission mode. By introducing the inverse path for event notifications and the shortest path for the delivery of events, the HPEQ protocol [28] speeds up new subscriptions in the sensor region under low transmission latency. However, to the best of the authors' knowledge, none of the event-driven clustering schemes consider how to group sensors in the event area into clusters. Most of them assumed that the sensors in event area can be organized into one cluster or the detected phenomenon occurs intermittently in the fully monitored area. Both of the assumptions are not suitable for continuous monitoring applications when the detected event changes gradually with their size and location.

Many target tracking or estimation algorithms as reported in [29] [30] were proposed for monitoring or tracking moving events. Event-driven clustering and event-based sampling are effective approaches used to balance the energy efficiency and desired monitoring accuracy in the target tracking or estimation algorithms [31]–[34]. However, in most of cluster-based target

tracking algorithms, only one cluster is active for tracking the moving event at each instant, leading to poor performance when the moving event lies in several clusters [33]. Moreover, most target tracking algorithms consider how to track the moving target without considering the development of the event area, and the target motion is not considered in the event-based state estimation algorithms.

In most continuous monitoring applications, QoS requirement varies according to the changes of monitoring process. For example, in a temperature monitoring application for vegetable shed automatic control system, when temperature is within the normal range (10°C – 30°C , for instance), the vegetables are growing under an appropriate growth condition. Farmers care more about the trend of the temperature variation rather than the exact value of temperature. When the temperature exceeds the normal range, the values at each sampling instant should be sent to users. To further reduce the energy consumption, some trade-off can be made between improving the energy efficiency and guaranteeing the detection performance [18]. Therefore, in the beginning, only a few data should be sent to CH to judge the event development tendency, the event-driven data reporting scheme can be used to save energy. Once an abnormal phenomenon is detected, data reporting switches to the time-driven scheme for accurate measurement. After the abnormal phenomenon disappears, data transmission will switch back to the event-driven scheme.

In EEAO, a novel 2-logical-coverage cluster formation scheme and cluster migration scheme are designed to form and activate proper clusters according to event development. The overlapping cluster formation operation is illustrated in Fig. 4. Firstly, non-overlap clusters are formed similar to traditional cluster formation schemes, as in Fig. 4(a). Secondly, each node chooses the next-closest CH to form the 2-logical-coverage overlapping clusters, as in Fig. 4(b).

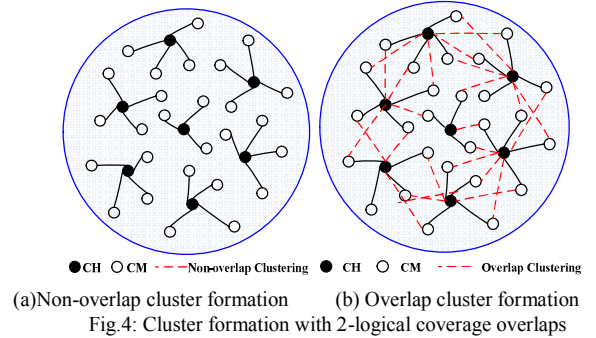


Fig.4: Cluster formation with 2-logical coverage overlaps

III. PRELIMINARIES

A. Network Model

We consider a network with N sensor nodes uniformly distributed in a $M \times M$ square monitoring area with high density, which satisfies the following assumptions:

- (1) There is a sink located in the middle of the monitoring area. The sensors and the sink are stationary after deployment.
- (2) The sensors are initially charged with identical energy, and not equipped with any GPS-like hardware. In addition, the sink does not have energy constraints.

(3) The links are symmetric (two nodes can communicate with each other using the same power). Nodes are time synchronous with the help of MAC protocols.

(4) Power control is allowed to vary the transmission power based on the distance between two nodes. The conflict control and channel selection would be dealt by using MAC protocols.

(5) Each node is assigned with a unique ID via ID assignment algorithms for cluster-based networks as in [25].

By using the radio model in [20], both the free space and multi-path channel energy consumption models are used according to the distance between transmitters and receivers, as shown in Fig. 5.

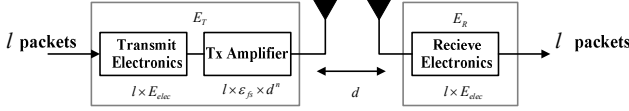


Fig.5: Energy Consumption Model

The energy dissipation for transmitting l packets over distance d is

$$E_T = \begin{cases} l(E_{elec} + \epsilon_{fs} \times d^2), & d < d_0 \\ l(E_{elec} + \epsilon_{mp} \times d^4), & d \geq d_0 \end{cases} \quad (1)$$

The energy dissipation for receiving l packets is

$$E_R = l \times E_{elec}, \quad (2)$$

where E_{elec} is the energy spent per bit to run the circuitry of each node, ϵ_{fs} and ϵ_{mp} are the energy required by amplifier in two transmission energy modes, $d_0 = \sqrt{\epsilon_{mp} / \epsilon_{fs}}$ is the threshold distance for two energy models. E_{DA} denotes the energy consumption for aggregating one packet, and $E_{DA} \ll E_T$. The notations and definition in EEAOC are listed in Table 1.

TABLE 1
NOTATION AND DEFINITION IN THE EEAOC PROTOCOL

Notation	Definition
N	Total number of sensors in the network
$S(i)$	Set of sensors in cluster i
n_i	Number of sensors in cluster i
CH_i	CH in cluster i
k_i	Number of overlapping regions in cluster i
R_s	Communication radius of sensors
R_C	CH competing radius
$E_{re}(i)$	Residual energy of sensor i
K_{CH}	Total number of CHs in the network
m_0	Number of sensors in the CH competing area
n_0	Number of sensors in each cluster
f	Number of data frames sent from a CH to the sink in a round
m_i^u	Number of sensors in the u^{th} overlapping region for cluster i
$r_p(t)$	Reading sensed by the p^{th} sensor in the network at instant t

B. Definitions

Definition 1 (Boundary nodes). A node x is a boundary node, if x is located in the overlapping area between cluster i and cluster j . The set of boundary nodes in the overlapping area between the two clusters i and j is

$$BN(i, j) = \{x \mid x \in S(i) \ \& \ x \in S(j)\}, \quad (3)$$

where $S(i)$ and $S(j)$ are the sets of sensors in cluster i and cluster j , respectively.

To detect event development, the whole network is divided into 2-logical-coverage subnets. That is, each node belongs to two adjacent clusters simultaneously. The role of boundary nodes includes two aspects: as 1-hop forwarding nodes when the single hop clusters are combined into 2-hop clusters; as beacon nodes to indicate the moving direction of the event.

Definition 2 (Overlapping degree). The overlap degree for cluster i is defined as

$$OD(i) = n_i / k_i, \quad (4)$$

where n_i is the total number of nodes in cluster i , k_i is the number of overlapping areas in cluster i .

In the EEAOC protocol, clusters are divided into the same number of overlapping regions, which means that k_i is a constant (denoted as k_0). The value of k_0 is dependent on the moving speed of the detected event. When an event moves at a high speed, the cluster migration should be operated at high frequency to avoid improper cluster formation. The value of k_0 should be greater such that the cluster migration is more energy-efficient and flexible to adapt to the high event development rate.

Definition 3 (Cluster size). The cluster size is defined as the total number of sensors in the cluster.

IV. THE EEAOC PROTOCOL

EEAOC operates in several epochs, and each epoch contains the cluster set-up phase and the steady data transmission phase. An epoch of EEAOC includes five steps (information collection, CH selection, cluster formation, data transmission, CH re-adjustment and cluster migration).

A. Information Collection Step

In this step, local and global information of the network will be obtained. First, the sink floods a ‘Hello message’ to the network. Based on the Received Signal Strength Indication (RSSI), the Euclidean distance between the ordinary nodes and the sink can be estimated [21]. Second, each node broadcasts a ‘Hand-shaking message’ to its neighbors within R_s which contains the information of node ID and residual energy.

B. Cluster Head Selection Step

For load balancing, CHs should be distributed uniformly in the monitoring area. A competing radius R_C is set to limit the broadcasting range of ‘CH competing message.’ In this work, we set $2R_C \leq R_s$ for ensuring network connectivity, and set $2R_C \leq d_0$ for avoiding long distance intra-cluster communication. The exact value of R_C is determined by the QoS requirements of users and the features of sensors in the network.

To save the competing overhead and avoid collision, a broadcast delay related to nodes’ residual energy is introduced. The broadcast delay of node i is given as

$$T_i = e^{1/E_{re}(i)}, \quad (5)$$

where $E_{re}(i)$ is the residual energy of node i . If a node receives ‘competing messages’ before its waiting time expires, it will give up the competition operation. Otherwise, it will broadcast the ‘competing message’ to announce itself as a CH in R_C . Due to the limited competing radius and broadcast delay, nodes with higher residual energy have higher chance to serve as CHs, and the selected CHs are uniformly distributed in the network.

Suppose that there are K_{CH} CHs selected in the network. In the ideal situation of CH distribution, the sensors in the network are covered by K_{CH} non-overlapping circles with radius R_C . Let $\lceil u \rceil$ denotes the smallest integer greater than u . The number of sensors in the circles (denoted as m_0) can be estimated as

$$m_0 = \left\lceil N \cdot \frac{\pi R_C^2}{M^2} \right\rceil. \quad (6)$$

As a result, the optimal number of CHs can be estimated as

$$K_{CH}^{opt} = N / m_0 \approx \left\lceil \frac{M^2}{\pi R_C^2} \right\rceil. \quad (7)$$

To format the 2-logical-coverage clusters, each node belongs to two CHs simultaneously. Since each cluster has the same cluster size, the cluster size under ideal CH distribution is a constant (denoted as n_0), which can be estimated as

$$n_0 = \left\lceil \frac{2N}{K_{CH}^{opt}} \right\rceil \approx 2 \left\lceil N \cdot \frac{\pi R_C^2}{M^2} \right\rceil \approx 2m_0. \quad (8)$$

Substituting (8) into (4), we obtain

$$OD(i) = \frac{n_i}{k_i} = \frac{n_0}{k_0} \approx 2 \left\lceil \frac{\pi N R_C^2}{k_0 M^2} \right\rceil. \quad (9)$$

It is seen from (9) that, for a given k_0 , the overlapping degree is a constant (denoted as OD_0) and approximately proportional to the square of the competing radius.

Once more than K_{CH}^{opt} ($K_{CH} > K_{CH}^{opt}$) nodes are selected as CHs, $K_{CH} - K_{CH}^{opt}$ redundant CHs should be eliminated. An ordinary node will broadcast a ‘cluster member message’ in its communication radius R_S . Each CH hears the message and counts the total number of received messages, $K_{CH} - K_{CH}^{opt}$ CHs receiving the least number of ‘cluster member messages’ will be eliminated. Finally, K_{CH}^{opt} CHs will broadcast ‘cluster head messages’ in their communication radius R_S to declare as the final CHs.

C. Cluster Formation Step

The 2-logical overlapping cluster formation is operated in three time intervals.

In the first time interval $[0, T_1]$, non-overlapping clusters are formed. Each node will connect to its closest CH (master CH) by sending a ‘cluster-join message’. If a node has more than one nearest CHs, it will choose the one with the smallest ID.

Remark 1: Under uniform distribution, the cluster size for each cluster is no more than m_0 after the first cluster formation round. We will prove the result by analyzing three cases:

Case I: the Euclidean distance between any pair of closest CHs is less than $2R_C$, and the cluster radius for each non-overlapping cluster is less than R_C .

As the number of nodes is m_0 when the cluster radius is R_C , the number of sensors in each cluster is less than m_0 in Case I.

Case II: the Euclidean distance between any pair of closest CHs is farther than $2R_C$ and there exists an ordinary node with distance from both CHs greater than R_C .

Since the node belongs to neither of the two clusters, it will announce itself as a CH. Therefore, the two CHs are not the closest pair, which contradicts the above assumption.

Case III: the Euclidean distance between any pair of two closest CHs is farther than $2R_C$. No nodes are with distance from both CHs greater than R_C .

In this case, the maximum Euclidean distance from a node to its closest CH is less than R_C , the number of cluster members for each cluster is less than m_0 .

Therefore, the cluster size for each cluster is less than m_0 .

In the second time interval $[T_1, T_2]$, overlapping regions are formed among clusters. All nodes send ‘overlapping cluster-join messages’ to their next-closest CHs (slave CHs) which contain less than n_0 CMs. The slave-CHs will broadcast ‘overlapping reply messages’ to confirm the overlapping cluster formation. In the end, each cluster contains n_0 CMs. The number of boundary nodes in every overlapping area is obtained by the corresponding CHs, and the adjacent CHs are connected by the boundary nodes.

Remark 2: In some cases, the total number of sensors N may not be a multiple of n_0 , which means that the network cannot be divided into clusters with equal cluster size. Anyhow, $K_{CH} - 1$ overlapping clusters will be grouped with cluster size n_0 and the last cluster will be grouped with cluster size $N - (K_{CH} - 1)n_0$.

In the last time interval $[T_2, T_3]$, equal-sized overlapping regions are divided. If an overlapping region has more than OD_0 boundary nodes, its slave-CH will send an ‘overlapping dismissal message’ to the redundant boundary nodes (with the farthest distance between themselves and the slave-CH). After receiving the dismissal messages, all the redundant boundary nodes broadcast ‘overlapping tuning messages’ in R_C . A CH will reply ‘overlapping join messages’ to the redundant boundary nodes if the following conditions are satisfied: (1) its corresponding cluster is overlapped with the cluster which the boundary nodes belongs to; (2) the boundary nodes in the overlapping region is less than OD_0 .

Each CH contains an overlapping cluster table with the information of which overlapping regions its members belong to, while each CM establishes a boundary table containing all IDs of its neighbors in the same overlapping region.

Remark 3: Although the cluster formation schemes is more complicated than non-overlapping cluster formation schemes, its overhead still can be reduced because it only needs to execute once during the whole working time.

D. Data Transmission Step

In continuous monitoring applications, the detection accuracy and energy efficiency can be traded according to the event development (the event is defined as the value of detected variable exceeds a pre-set threshold r_0). Two data reporting schemes are used to balance the requirements of detection

accuracy and energy efficiency. As shown in Fig. 6, if the measured variable is in the normal range, the sensors transmit to CHs only when they sense a change (the difference between two adjacent sampling instant is greater than the reference value r_{th}). Nodes will communicate at a low frequency for saving energy while guaranteeing network connectivity. Once the event is detected, sensors in the event area send data to CHs periodically at a higher frequency, all the freshly sensed values at each sampling instant will be sent to users for improving the monitoring accuracy.

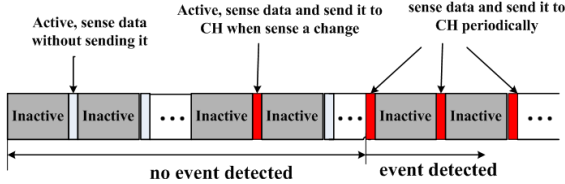


Fig. 6: Data transmission in EEOC

As the monitoring area is overlapped by 2-logical-coverage clusters and each CM only needs to send the data to one of its belonging CHs, $\lceil K_{CH}/2 \rceil$ CHs will be activated to play the data fusion role (active CHs) and the rest work as ordinary nodes (inactive CHs) in the data transmission step. The active CHs should be distributed uniformly in the network. First, the CH with the highest ID will send a ‘cluster initialization message’ to its neighbor CH with the least overlapping regions to be an active CH. And the neighbor CHs will repeat this initialization operation until $\lceil K_{CH}/2 \rceil$ CHs are selected. Then, the active CHs will send ‘connectivity-request message’ to their members to inform them which CHs the sensed data should be sent to. Once a node detects the event, it will send a ‘query message’ to its CH at first, and the CH will broadcast a ‘reply message’ to inform its members to switch the data reporting frequency to a high level.

Each CH creates a TDMA schedule for time slots allocation. To reduce the interference among clusters, each cluster communicates through different channels. In the inter-cluster communication slot, each CH sends the aggregated data to the sink directly.

E. CHs Re-adjustment and Cluster Migration Step

As CHs consume more energy than ordinary nodes, CHs rotation is essential to balance the energy consumption. However, frequent updating of CHs results in additional energy consumption [22]. We use the total energy depletion of CHs in the last round as the threshold for re-adjusting CHs. The energy depletion of CH_i during the last round is

$$E_{CH}(i) = (n_0 I E_{elec} + n_0 (l+1) E_{DA} + l \varepsilon_{mp} \cdot d_{CH}^2) \cdot f, \quad (10)$$

where d_{CH} is the distance from CH_i to the sink, and there are f frames sent from CH_i to the sink in the last round.

When the residual energy for CH_i is less than $E_{CH}(i)$, the nearest CM with residual energy more than $E_{CH}(i)$ in the active cluster is selected to be CH in the next round. The current CH will send a ‘CH-switching message’ to its members.

When the event spreads to another region, a demand-driven cluster migration strategy is utilized to ensure that the adjacent active nodes are divided into the same cluster as far as possible.

Boundary nodes in the overlapping areas are used to indicate the direction of event development. Every boundary node maintains a boundary node table to store the information about the IDs of its master-CH, slave-CH and neighbor boundary nodes (BNNs) in the same overlapping region. According to the average reading of the boundary nodes in the same overlapping region, the direction of the event development can be estimated and the proper adjacent CHs are activated for event monitoring.

For cluster i , suppose the number of boundary nodes in its u_{th} ($u = 1, 2, \dots, k_0$) overlapping area is m_i^u , the reading of the p^{th} sensor at the sampling instant t is $r_p(t)$. Hence, the difference of average reading at the adjacent sampling instant t and $(t+1)$ in the u_{th} overlapping area (denoted as $\Delta r_u(t+1)$) is

$$\Delta r_u(t+1) = \frac{1}{m_i^u} \sum_{p=1}^{m_i^u} (r_p(t+1) - r_p(t)). \quad (11)$$

The difference of average reading at sampling instant t between the u_{th} and v_{th} overlapping areas in cluster i (denoted as $\Delta q_{uv}(t)$) is

$$\Delta q_{uv}(t) = \frac{1}{m_i^u} \sum_{p=1}^{m_i^u} r_p^u(t) - \frac{1}{m_i^v} \sum_{p=1}^{m_i^v} r_p^v(t). \quad (12)$$

The parameter Δr_u indicates whether the detected event will expand. When the reading of a sensor is above r_0 , an event is detected in the network. If Δr_u decreases continuously, the event may disappear sooner. Most of the sensors in these areas will go into the sleep mode while a few of them transmit the sensed data at a low frequency for network connectivity. The parameter Δq_{uv} indicates the motion direction of the event. If Δq_{uv} decreases continuously, it means that the event will move to the adjacent cluster sharing the overlapping region v . As shown in Fig. 7, an event is detected in a small region covered by Cluster B. The CH of Cluster B judges the direction of event development based on the boundary node tables. When the average readings of *node-a* and *node-b* decrease while the average readings of *node-c* and *node-d* increase, the event will spread to the area covered by Cluster C. Therefore, the CH of Cluster B sends an ‘event moving’ message to the CH of Cluster C to activate all CMs in Cluster C for data gathering in the next round.

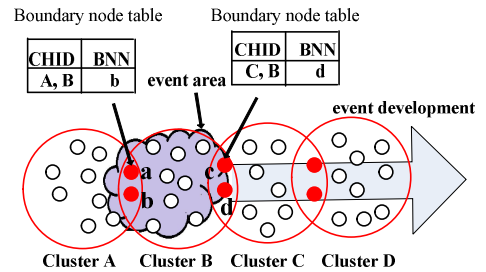


Fig. 7: Cluster migration with event development

To prevent misjudgment of event development, a judgment window is introduced to control the aggressiveness level of cluster migration. The window contains a sequence of F number of Δr_j and L number of Δq_{jff} in consecutive time intervals. If the sequence of Δr_j strictly monotonically

increases and the sequence of Δq_{if} are strictly non-increasing, the cluster migration can be manipulated.

When the event spreads to a larger area, the adjacent 1-hop clusters are combined into a 2-hop cluster via the boundary nodes. As shown in Fig. 8, at the beginning, the event area is covered by the blue chain line cluster. Once the event spreads to a larger area, the three green dotted line clusters will combine with the blue one to form a 2-hop cluster. The nodes can communicate with CH directly or through the nearest boundary nodes. The boundary nodes aggregate the forwarding data and send it to CHs. If the event spreads to an even larger region, more than one 2-hop clusters are combined to ensure all the event development is detected.

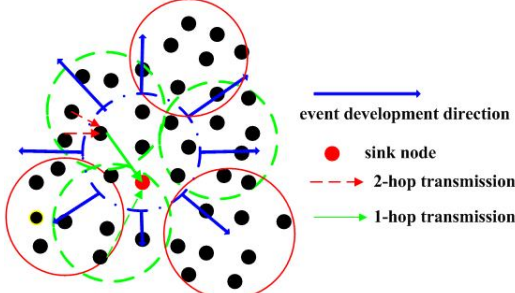


Fig. 8: Cluster migration with event expansion

Algorithm 1: Cluster migration and CH re-adjustment algorithm

```

1: for each CH do
2:   if  $E_{re} < E_{th}$  then
3:     Send 'CH-switching message' & 'Cluster-join messages'
4:   end
5:   if  $\Delta r > r_{th}$  &  $\Delta q > q_{th}$  then
6:     send 'event-moving message' to the next active CH;
7:     If  $k_0$  'event-moving message' should be sent then
8:       send 'cluster extension messages' to neighbor CHs;
9:     end
10:  end
11:  if receive 'cluster extension messages' then
12:    broadcast 'cluster extension messages' to its CMs;
13:    be 1-hop or 2-hop CMs;
14:  end
15: end
16: for each CMs do
17:   if receive 'CH-switching messages' then
18:     if CM_ID=CH_swich_ID then
19:       be a CH in the coming round;
20:     else
21:       send 'cluster-join message' to the new CH;
22:     end
23:   end
24:   if receive 'event-moving message' then
25:     send data to the specified CH;
26:   end
27:   If receive 'cluster extension message' then
28:     send data to the 1-hop node;
29:   end
30: end

```

Algorithm 1 provides the pseudo-code for cluster migration and CH re-adjustment of EEAOC protocol. From Lines 1 to 5, the CH re-adjustment operation starts by current CHs with residual energy lower than the threshold. From Lines 6 to 11, the cluster migration operation starts by active CHs when the event development condition is met. From Lines 12 to 16, the cluster extension operation starts when all directions detected

the event development. From Lines 17 to the end of the code, each CM will decide its role in the coming round after receiving the corresponding control messages.

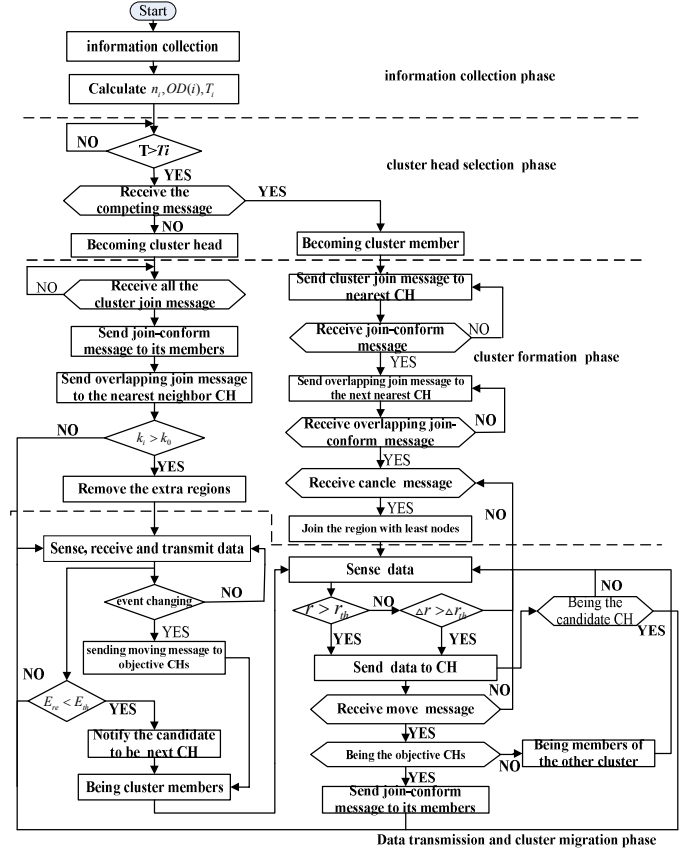


Fig. 9: Flow chart of EEAOC

The flow chart of EEAOC protocol is shown in Fig. 9. It can be seen that the information collection step, cluster head selection step and cluster formation step only need to be operated once after the sensors are deployed over the monitoring area, which can reduce the complexity of the EEAOC protocol. The data transmission step will be rotated over time while the CH re-adjustment and cluster migration steps will be operated only when the CH re-adjustment and cluster migration conditions are satisfied.

V. PROTOCOL ANALYSIS AND SIMULATIONS

A. Protocol Analysis

(1) There is only one CH in each CH competing radius.

In the CH selection step, the node with the highest residual energy will broadcast competing message within the competing radius. All the nodes in its competing radius will give up the competing operation. As a result, the node with the highest residual energy is the only CH in its competing radius.

(2) All nodes in the network are mapped to the cluster topology by EEAOC.

In EEAOC, on one hand, all the nodes receiving the CH competing messages can be connected by at least one CH. On the other hand, all the nodes which do not receive the CH competing messages will announce themselves as CH, and can

be connected by neighboring CHs. The isolate points are avoided in the network.

(3) The message complexity of EEAO is $O(N)$.

The details of control messages during the cluster-setup steps are summarized in Table 2.

TABLE 2
CONTROL MESSAGES DURING THE CLUSTER SET-UP STEPS

Cluster setup steps	Message types	Number of messages	Message Producers
Information Collection	hello	1	sink node
	hand-shaking	N	ordinary nodes
CH Selection	CH competing	$K_{CH}(K_{CH} < N)$	ordinary nodes
	cluster member	$N - K_{CH}$	ordinary nodes
	cluster head	$K_{CH}^{opt} (K_{CH}^{opt} < K_{CH})$	CH
Cluster formation	cluster-join	$N - K_{CH}^{opt}$	CM
	overlapping cluster-join	N	all nodes
	overlapping reply	K_{CH}^{opt}	CH
	overlapping -dismissal	$K_{CH}^{*} (K_{CH}^{*} < K_{CH}^{opt})$	slave-CH
	overlapping -tuning	$N^{*} (N^{*} < N)$	boundary nodes
	overlapping -join	$K^{*} (K^{*} < K_{CH}^{opt})$	slave-CH
CH readjustment & cluster migration	CH switching	$0 \sim K_{CH}^{opt}$	current CH
	event moving	$0 \sim K_{CH}^{opt}$	current CH
	cluster extension	$0 \sim K_{CH}^{opt}$	current CH

Remark 4: In the CH re-adjustment and cluster migration step, the minimum number of control messages is 0 when the current CHs continue to be the CHs in the next round. At most K_{CH}^{opt} ‘CH-switching messages’, K_{CH}^{opt} ‘event moving messages’ and K_{CH}^{opt} ‘Cluster extension messages’ should be sent when all CHs need to be adjusted.

Based on Table 2, the message complexity of EEAO is O_{Total} satisfies:

$$O_{Total} \geq 4N + K^{*} + N^{*} + K_{CH}^{*} + K_{CH}^{opt} + 1 > 4N + 1 \quad (13)$$

$$O_{Total} \leq 4N + K^{*} + N^{*} + K_{CH}^{*} + 4K_{CH}^{opt} + 1 < 10N + 1 \quad (14)$$

Combining inequalities (13) and (14), we obtain

$$4N + 1 < O_{Total} < 10N + 1. \quad (15)$$

Thus, the overall communication overhead complexity for the cluster set-up steps in the network is $O(N)$.

B. Simulations

In this section, we evaluate the performance of EEAO via MATLAB. A comparison among LEACH (time-driven, single hop) and DEECIC (time-driven, multi-hop) and TEEN (event-driven) is performed in two simulation scenarios under different event scalabilities and changing rates. In this simulation, we will take the same configuration parameters in LEACH. The pseudo-codes of LEACH are publicly available in <http://csr.bu.edu/sep/LEACH.m> by Gergios Smaragdakis. In the simulations for TEEN, the cluster formation process is the

same as the process in LEACH protocol. Moreover, we set hard threshold $HT=1$ and soft threshold $ST=0.01$. We assume that the reading of the nodes in the event area is $2+\sin(0.1r)$ and the reading of the nodes outside the event area is 0. The DEECIC protocol is a multi-hop time-driven clustering protocol. Like the LEACH protocol, all the nodes in the monitoring area send data to the sink node periodically. As a result, the data reporting process is independent of the event area.

The process of event development is modeled as a circular region moving with uniform rectilinear speed. Ten simulation runs are performed and statistics are averaged over these 10 runs. The specific parameters are set as in Table 3.

TABLE 3
NETWORK PARAMETERS

Parameter	Value
Network size (two cases)	$(100 \times 100) \text{ m}^2$ with 100 nodes
	$(200 \times 200) \text{ m}^2$ with 200 nodes
BS position	center of the monitoring area
Initial energy	0.5J
l	4000 bits
R_c	20 m

We assume that the detected event occurs in a circular region and the event development is modeled by an array $[x_e, y_e, r_e, v_e, v_r]$, where (x_e, y_e) is the coordinates for center point of the event area, r_e is the radius of event area, v_e is the movement speed of the event area, and v_r is the expansion speed of the event size. v_e and v_r are constant such that the detected event fluctuates with uniform rectilinear speed. As a result, the analytical expression of the event area is

$$(x - x_e)^2 + (y - y_e)^2 \leq r_e^2, \quad 0 \leq x_e, y_e \leq M \quad (16)$$

Four event development cases are utilized to evaluate the performance of the EEAO protocol. The specific parameters of the event development are set as in Table 4.

TABLE 4
PARAMETER SETTINGS OF EVENT DEVELOPMENT

Parameter	Value		
	Case 1	Case 2	Case 3
(x_0, y_0)	(25,25)	(50,50)	(25,25)
r_0	20m	10m	10m
v_e	0.03m/round	0	0.03m/round
v_r	0	0.02m/round	0.03m/round

- **Case 1:** the event moves while maintaining its shape. In this situation, y_e and r_e are constant and x_e is a variable set as $x_e = x_0 + v_e \times t$, where x_0 is the initial value of x_e . The schematic diagram is shown in Fig. 10(a).
- **Case 2:** the event expands to a larger area while keeping its centre. In this situation, x_e and y_e are constant values while r_e is a variable set as $r_e = r_0 + v_r \times t$, where r_0 is the initial value of r_e . The schematic diagram is shown in Fig. 10(b).
- **Case 3:** the event moves while extending its area, and it will spread to the whole network eventually. The schematic diagram is shown in Fig. 10(c).
- **Case 4:** All sensors in the monitoring area detect the event and send the gathered data to the sink periodically.

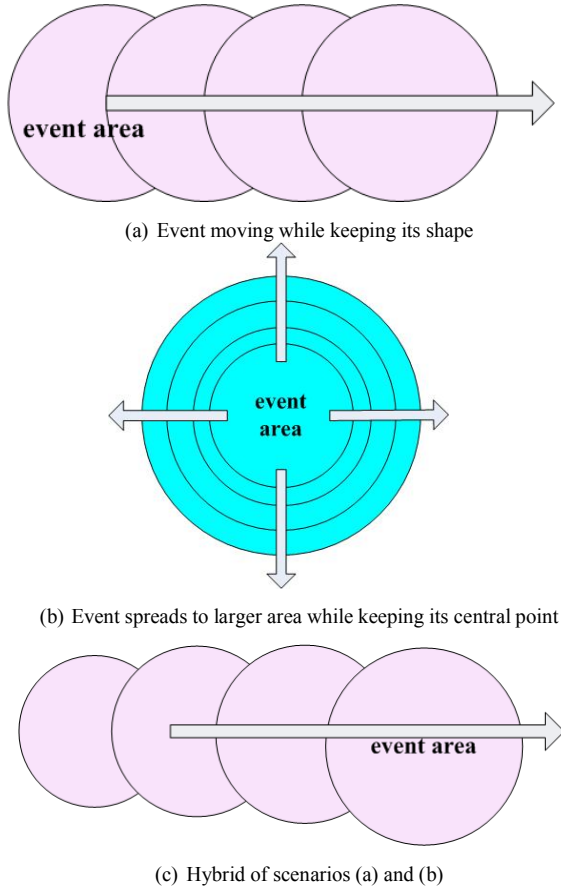
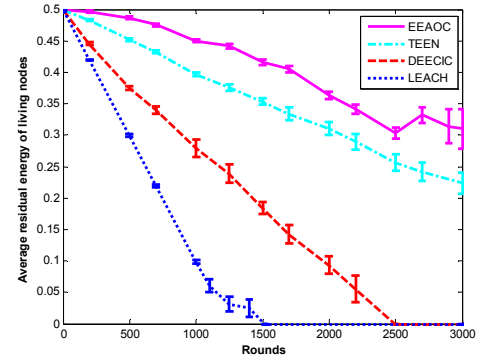


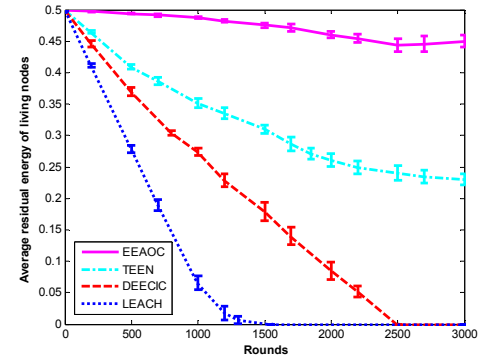
Fig. 10: Event area development

When a node runs out of its energy, it is taken as dead. In this work, we take the time when 10% nodes die (TND) to evaluate the network lifetime because the death of 10% nodes has a great influence on the connectivity of network [9]. Moreover, we assume that all packets received by a CH are well aggregated into a constant length.

First, we evaluate the EEAOC protocol with two network sizes under Case 3 to investigate its adaptability to the network size. The results in Fig. 11 show that EEAOC retains most residual energy compared with other protocols in both network sizes. The results in Fig. 12 indicate that most of the nodes are still alive under the network size with 200 nodes and the second most of the nodes are still alive under the network size with 100 nodes. Comparing the results in Fig. 11 and Fig. 12, we can see that LEACH performs better in small-scale networks due to the single hop intra-cluster communication feature and DEECIC performs better in large-scale networks because of the 2-hop intra-cluster communication feature. Taking the advantage of event-driven data reporting schemes, the TEEN protocol has more living nodes than LEACH. Since EEAOC combines the virtues of event-driven data reporting scheme and event-based cluster formation mechanism, it can decide whether or not to transmit to the CHs continuously based on the data attributes, and form clusters to single hop or 2-hop according to the event development. Hence, EEAOC can prolong the network lifetime in both network sizes.

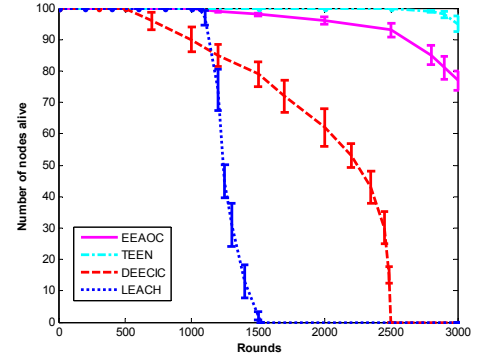


(a) Network with 100 nodes

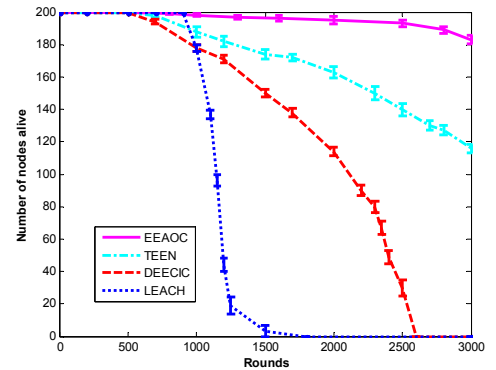


(b) Network with 200 nodes

Fig. 11: Average residual energy of living nodes over rounds



(a) Network with 100 nodes



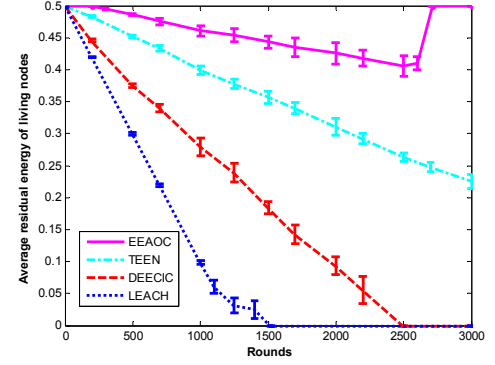
(b) Network with 200 nodes

Fig. 12: Number of nodes alive over rounds

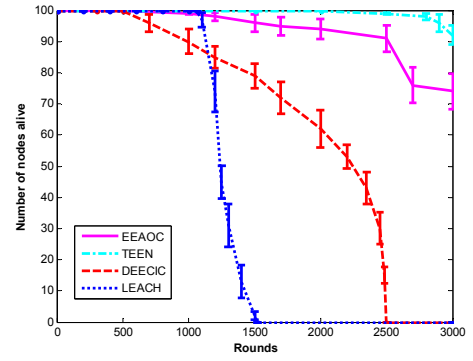
Second, we simulate EEAOC under the four event development cases mentioned above. Figs. 13–16 show the number of living nodes and average residual energy of the living nodes for EEAOC, TEEN, DEECIC and LEACH under four event development cases. The results in Figs. 13 (a), 14 (a) and 15 (a) indicate that the EEAOC protocol retains most average energy under Case 1–Case 3. In Case 1–Case 3, the event occurs in a local area and changes dynamically with its location and size. The sensors in the event area send data to the sink periodically while a few sensors outside the event area will communicate with the sink at a low frequency. By considering the residual energy during the CH selection phase, DEECIC retains more energy compared with LEACH. As event-driven reporting scheme is used, no data will be transmitted most of the time for TEEN. As a result, TEEN achieves better energy efficiency compared with LEACH and DEECIC. Compared with TEEN, more communication traffic is required by the EEAOC protocol due to the QoS requirement of continuous monitoring. However, since EEAOC can organize sensors into more reasonable clusters, it achieves even higher energy efficiency compared with the TEEN protocol. Since the vast majority of nodes exhaust their energy after 2500 rounds while a few nodes maintain their initial energy as they are always located outside the event area, there is a significant increase of the average residual energy for living nodes after the nodes with low residual energy die. In some cases, a few nodes are located outside the event area all the time. After all other nodes die, the average residual energy of living nodes is equal to the initial energy (for example, in Case 1).

The results in Figs. 13 (b), 14(b) and 15 (b) show that although the first node die time of EEAOC (1040th round) is almost the same to LEACH (1039th round) in some specific cases (Case 1), EEAOC achieves a prolonged network lifetime because TND is much longer in all cases, compared with LEACH and DEECIC. We also discover that, in a small monitoring area with sensors randomly distributed, the 2-hop intra-cluster clustering protocol (DEECIC) sometimes cannot achieve better result than the single-hop inter-cluster clustering protocol. Due to the flexible cluster formation scheme, EEAOC achieves higher energy efficiency than LEACH and DEECIC. The results in Fig. 16 show that our method achieves better performance even in worst case (Case 4, all the sensors should be actively sending data at every round). As TEEN transmits data to the sink only when the nodes sense a change in the network, the TND of TEEN is a little longer than EEAOC. However, since no information is obtained within a certain period of time, data distortion will be inevitable, as shown in Fig.17, when an event is detected in the network. In LEACH and DEECIC, all readings are sent to the sink to achieve high accuracy regardless of the requirements of users. In TEEN, no information will be obtained within a certain period of time which is not suitable for continuous monitoring applications. In EEAOC, the sensed data at every sampling instant are sent to the sink to achieve high monitoring accuracy when the event is detected while fewer data are sent to the sink to detect the tendency of event development. The monitoring accuracy and

energy efficiency are well balanced, making it suitable for continuous monitoring applications.

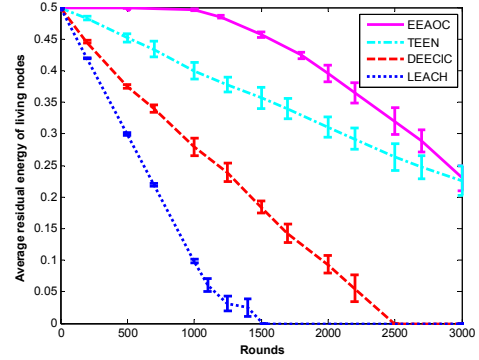


(a) Average residual energy of living nodes

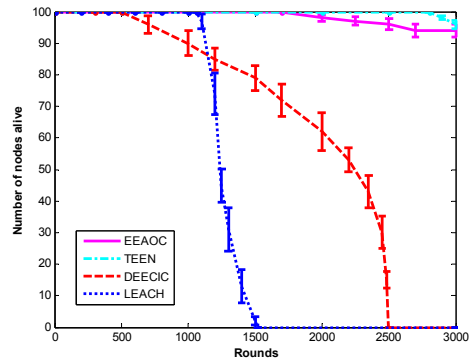


(b) Number of nodes alive

Fig.13: Living nodes and average residual energy under Case 1

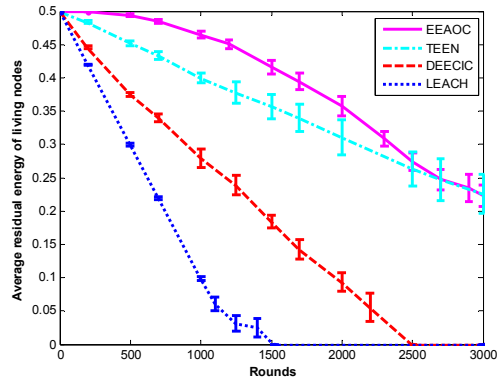


(a) Average residual energy of living nodes

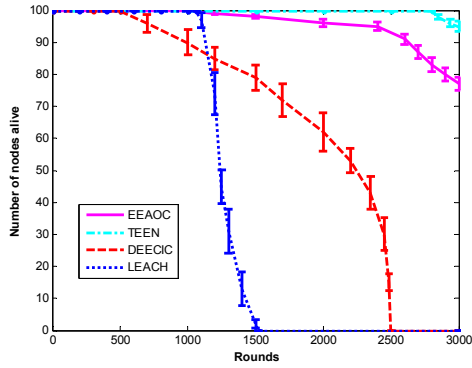


(b) Number of nodes alive

Fig.14 Living nodes and average residual energy under Case 2

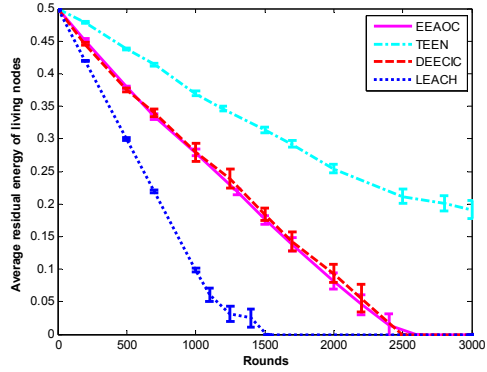


(a) Number of nodes alive

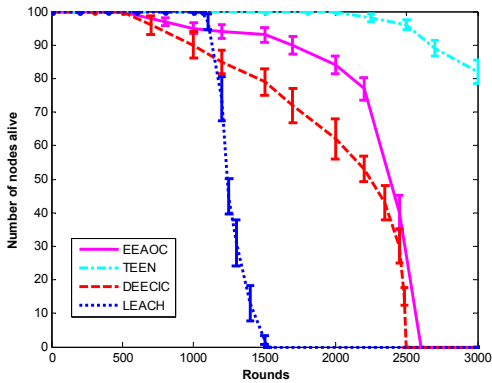


(b) Average residual energy of living nodes

Fig.15: Living nodes and average residual energy under Case 3



(a) Average residual energy of living nodes



(b) Number of nodes alive

Fig.16 Living nodes and average residual energy under Case 4

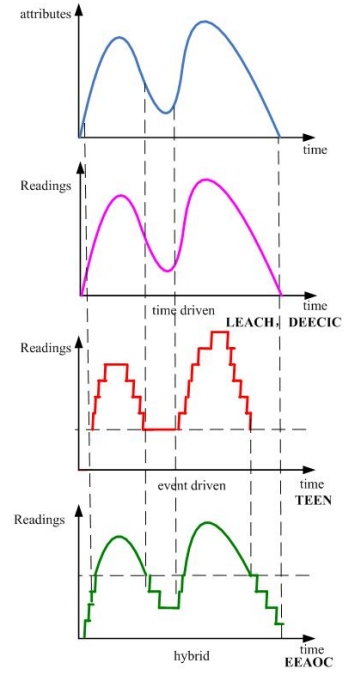


Fig. 17: Information obtained with different protocols

The comparison of TND among EEAOC and other protocols is depicted in Fig. 18 (mean, standard deviation, 95% confidence interval). It is shown that EEAOC can obtain better performance when the detected event occurs in a large event area (the TND of our EEAOC protocol can be prolonged nearly by 1401% and 210% under Case 1, 200% and 290% under Case 2, 140% and 220% under Case 3, 50% and 100% when compared with LEACH and DEECIC respectively). To meet the continuous monitoring requirement, certain amount of data should be sent to sink even if no event is detected in the network. As a result, EEAOC obtains a little shorter TND than the event-driven TEEN protocol (the TND of our EEAOC protocol is nearly 20%, 5%, 20% and 20% shorter compared with TEEN protocol under Case 1–Case 4, respectively). The comparison results have illustrated the high energy efficiency of the proposed EEAOC protocol in continuous monitoring applications.

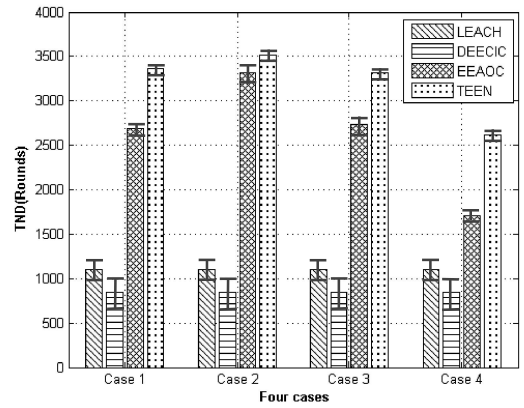


Fig.18: Network lifetime under the four cases

Finally, to evaluate the influence of the cluster migration scheme in EEAOC protocol, we simulate EEAOC without considering the cluster migration scheme under Case 3, and compare with the full version of EEAOC protocol. The results in Fig. 19 indicate that the cluster migration scheme makes a great contribution in prolonging the network lifetime. The cluster migration scheme can improve the energy efficiency since it is adaptive to the event development. Associated with our cluster migration scheme, the nodes in the event area are grouped into proper clusters for data gathering, which can reduce unnecessary data transmission from irrelevant nodes.

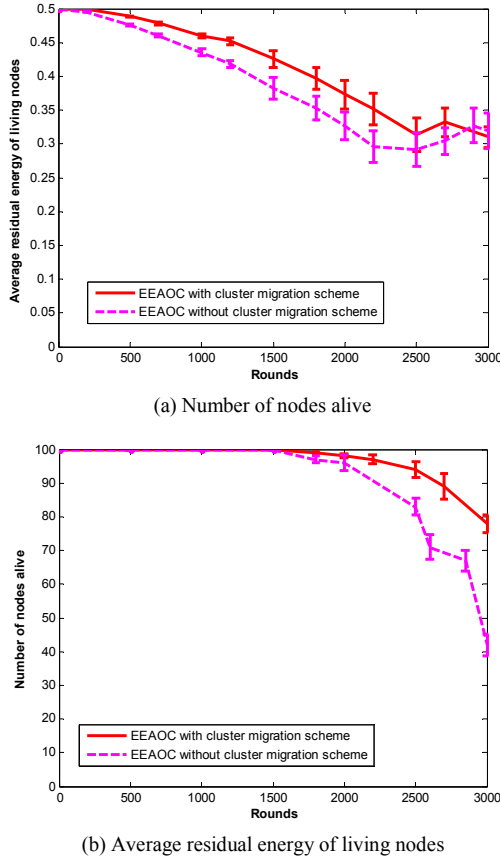


Fig.19: Living nodes and average residual energy under Case 3

VI. CONCLUSION

In this work, we have presented a dynamic clustering method EEAOC for dynamic continuous monitoring applications. In the proposed EEAOC, the collaborative CH re-adjustment and cluster migration technique is used to activate appropriate clusters for continuous monitoring via the 2-logical-overlapping clustering scheme. As a result, the cluster topology can be switched according to the event development with low communication overhead. The hybrid data reporting scheme is operated subsequently to balance the energy efficiency and detection accuracy. By doing so, an energy-efficient operation can be achieved by EEAOC. The simulation results have confirmed that EEAOC obtains better energy efficiency than time-driven clustering protocols and better detection accuracy than event-driven protocols. It is thus concluded that the EEAOC protocol is more suitable for the

dynamic continuous monitoring applications compared with existing clustering protocols.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey, *Computer Networks*, 38(4): 393–422, 2002.
- [2] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, *Computer Networks*, 52(12): 2292–2330, 2008.
- [3] Q. Liu, Z. Wang, X. He and D. H. Zhou, Event-based recursive distributed filtering over wireless sensor networks, *IEEE Transactions on Automatic Control*, 60(9): 2470–2475, 2015.
- [4] Z. Wang and Y. Niu (editors), Special Issue on “Distributed Estimation and Filtering for Sensor Networks”, *International Journal of Systems Science*, 42(9): 1421–1425, 2011.
- [5] V. Potdar, A. Sharif, E. Chang, Wireless sensor networks: a survey, *In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops*, pp.636–641, 2009.
- [6] C. F. Wang, J. D. Shih, B. H. Pan, and T. Y. Wu, A network lifetime enhancement method for sink relocation and its analysis in wireless sensor networks, *IEEE Sensors Journal*, 14(6): 1932–1943, 2014.
- [7] Y. Hu, Y. Niu, F. H. Gan, Congestion control for wireless sensor networks: a survey, *In Proceedings of the 26th Chinese Control and Decision Conference*, pp. 4853–4858, 2014.
- [8] A. F. Liu, P. H. Zhang, Z. G. Chen, Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks, *Journal of Parallel and Distributed Computing*, 71(10): 1327–1355, 2011.
- [9] A. F. Liu, X. Y. Wu, Z. G. Chen, W. H. Gui, Research on the energy hole problem based on unequal cluster-radius for wireless sensor networks, *Computer Communications*, 33(3): 302–321, 2010.
- [10] M. E. Keskin, İ. K. Altınel, N. Aras, C. Ersoy, Wireless sensor network lifetime maximization by optimal sensor deployment, activity scheduling, data routing and sink mobility, *Ad Hoc Networks*, 17(6): 18–36, 2014.
- [11] K. Malav, D. Gupta, V. Murray, Energy efficient routing in zigbee wireless sensor network- a review. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(4): 336–343, 2015.
- [12] M. M. Afsar, M. Tayarani-N, Clustering in sensor networks: a literature survey. *Journal of Network and Computer Applications*, 46:198–226, 2014.
- [13] A. Abbasi, M. Younis, A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14–15): 2826–2841, 2007.
- [14] N. C. Rathore, P. Gour, B. S. Meena, Improved clustering protocol for delay minimization, *International Journal of Computer Networking, Wireless and Mobile Communications*, 3(5): 41–48, 2013.
- [15] M. A. Kafi, D. Djenouri, J. B. Othman, N. Badache, Congestion control protocols in Wireless Sensor Networks: a survey, *IEEE Communications Surveys & Tutorials*, 16(3): 1369–1390, 2014.
- [16] B. D. Lee, K. H. Lim, An energy-efficient hybrid data-gathering protocol based on the dynamic switching of reporting schemes in Wireless Sensor Networks, *IEEE Systems Journal*, 6(3): 378–387, 2012.
- [17] A. R. Wilson, Event triggered analog data acquisition using the exponential moving average, *IEEE Sensors Journal*, 14(6): 2048–2055, 2014.
- [18] N. Bouabdallah, M. E. Rivero-Angeles, B. Sericola, Continuous monitoring using event-driven reporting for cluster-based wireless sensor

- networks, *IEEE Trans. on Vehicular Technology*, 58(7): 3460–3479, 2009.
- [19] G. Amato, S. Chessa, C. Gennaro, C. Vairo, Querying moving events in Wireless Sensor Networks, *Pervasive and Mobile Computing*, 16: 51–75, 2015.
- [20] A. P. Chandrakasan, A. C. Smith, W. B. Heinzelman, An application specific protocol architecture for wireless micro sensor networks, *IEEE Transactions on Wireless Communications*, 1(4): 660–669, 2004.
- [21] Y. Liao, H. Qi, W. Q. Li, Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks, *IEEE Sensors Journal*, 13(5): 1498–1506, 2013.
- [22] H. Zhen, Y. Li, G. J. Zhang, Efficient and dynamic clustering scheme for heterogeneous multi-level Wireless Sensor Networks, *Acta Automatica Sinica*, 39(4): 454–460, 2013.
- [23] M. Sabet, H. R. Naji, A decentralized energy efficient hierarchical cluster-based routing algorithm for wireless sensor networks, *International Journal of Electronics and Communications*, 69(5): 790–799, 2015.
- [24] M. A. Mamun, Y. Koi, N. Nakaya, Y. Hagihara, G. Chakraborty, HEHC: Heterogeneous-aware enhanced hierarchical clustered scheme for wireless sensor networks, *In Proceedings of SICE Annual Conference*, pp.1517–1522, 2011.
- [25] Z. X. Liu, Q. C. Zhang, L. X., X. P. Guan, A distributed energy-efficient clustering algorithm with improved coverage in Wireless Sensor Networks, *Future Generation Computer Systems*, 28(5): 780–790, 2012.
- [26] A. Manjeshwar, D. P. Agrawal, TEEN: A routing protocol for enhanced efficiency in wireless sensor networks, *In Proceedings on Parallel and Distributed Processing Symposium*, pp.2009–2015, 2001.
- [27] A. Manjeshwar, D. P. Agrawal, APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, *In Proceedings on Parallel and Distributed Processing Symposium*, 2002.
- [28] B. A. Pazzi, R. W. N. Araujo, R. B. HPEQ: A hierarchical periodic, event-driven and query-based wireless sensor network Protocol, *In Proceedings of IEEE Conference on Local Computer Networks*, pp. 560–567, 2005.
- [29] C. Huang, D. W. C. Ho, J. Q. Lu, Partial-Information-Based distributed filtering in two-targets tracking sensor networks, *IEEE Transactions on Circuits and Systems I*, 59(4): 820–832, 2012.
- [30] S. Chen, D. W. C. Ho, C. Huang, Fault reconstruction and state estimator design for distributed sensor networks in multi-targets tracking, *IEEE Transactions on Industrial Electronics*, 62(11): 7091–7102, 2015.
- [31] J. Yan, C. Chen, X. Luo, H. Liang, X. Guan, X. Yang, Topology optimisation-based distributed estimation in relay assisted wireless sensor networks, *IET Control Theory & Applications*, 8(18): 2219–2228, 2014.
- [32] W. Zhou, J. Mou, T. Wang, T. Ji, J. Fang, Target-synchronization of the distributed wireless sensor networks under the same sleeping-awaking method, *Journal of the Franklin Institute-Engineering & Applied Mathematics*, 349(6): 2004–2018, 2012.
- [33] T. Jing, H. Snoussi, C. Richard, Prediction-based cluster management for target tracking in wireless sensor networks, *Wireless Communications and Mobile Computing*, 12: 797–812, 2012.
- [34] S. Lee, I. Hwang, Event-based state estimation for stochastic hybrid systems, *IET Control Theory & Applications*, 9(13): 1973–1981, 2015.