

Article

A Robust and Energy-Efficient Weighted Clustering Algorithm on Mobile Ad Hoc Sensor Networks [†]

Huamei Qi ¹, Fengqi Liu ¹, Tailong Xiao ^{1,*} and Jiang Su ²

¹ School of Information Science and Engineering, Central South University, Changsha 410000, China; qhm@csu.edu.cn (H.Q.); liufengqi@csu.edu.cn (F.L.)

² Information Technology Department, China Life Ecommerce Company Limited Changsha Regional Branch, Changsha 410000, China; jiangsu_cs_clec@chinalife.com.cn

* Correspondence: 0905140216@csu.edu.cn

[†] This paper is an extended version of our paper published in International Conference on 11th Annual International Conference on Combinatorial Optimization and Applications (COCO'A'17), Shanghai, China, 16–18 July 2017; pp. 182–195.

Received: 12 May 2018; Accepted: 25 July 2018; Published: 1 August 2018



Abstract: In an Ad hoc sensor network, nodes have characteristics of limited battery energy, self-organization and low mobility. Due to the mobility and heterogeneity of the energy consumption in the hierarchical network, the cluster head and topology are changed dynamically. Therefore, topology control and energy consumption are growing to be critical in enhancing the stability and prolonging the lifetime of the network. In order to improve the survivability of Ad hoc network effectively, this paper proposes a new algorithm named the robust, energy-efficient weighted clustering algorithm (RE²WCA). For the homogeneous of the energy consumption; the proposed clustering algorithm takes the residual energy and group mobility into consideration by restricting minimum iteration times. In addition, a distributed fault detection algorithm and cluster head backup mechanism are presented to achieve the periodic and real-time topology maintenance to enhance the robustness of the network. The network is analyzed and the simulations are performed to compare the performance of this new clustering algorithm with the similar algorithms in terms of cluster characteristics, lifetime, throughput and energy consumption of the network. The result shows that the proposed algorithm provides better performance than others.

Keywords: mobile Ad hoc sensor networks; energy-efficient; robustness; fault detection

1. Introduction

A collection of mobile nodes can make up mobile Ad hoc sensor networks, which are not intervened on by any centralized access point [1,2]. In order to assist other nodes in the network to build information links, nodes can be used as router or terminal or both of them [3,4]. Mobile Ad hoc sensor network can be utilized in these scenarios, such as military and habitat monitoring [5,6], target tracking network [7,8]. For example, in the animal tracking system, sensor nodes are loaded on the body of animals in order to observing animal behavior. Generally, the animals (such as zebras, elephants etc.) have the low group mobility. Additionally, the sensor nodes cannot be charged in this situation. Hence, the energy consumption is a vital problem for the limitation of the energy. Moreover, since the animals live in a relatively bad environment, the fault node will be a necessary part needed to be considered. Therefore, our proposed protocol is motivated by this kind of applications.

It has been proved that clustering in hierarchical structure is an effective scheme to improve the network survivability [9,10]. In [9], it is introduced that the clustering method of fuzzy logic theory in the ECPF protocol. At the same time, in order to solve the problem of clustering non-uniformity,

LEACH-C algorithm is proposed in [10] and the residual energy and location of the cluster node are decided by the base station. However, how to elect the optimal cluster heads (CHs) and how can the optimal number of nodes be assigned become the bottleneck [11–13]. The structure of cluster is especially more significant in large scale network. Instead of using gateways to exchange data packet, nodes in the network are distributed and confined within each cluster. The cluster members only need to collect data and transmit packet to CHs.

Though the process of clustering will consume a little of energy, the overhead of routing control can be reduced a lot. Apart from above, the measurement will cause some difficulties due to the high mobility of nodes. In this paper, a high density network is considered, so we adopt the specific group mobility model [14–16]. In order to reduce the re-clustering times and save more energy, the novel mobility measurement is adopted. Nodes with similar motion pattern will form into a group or a cluster. The linear displacement of nodes is based on the record of history movement. The displacement index of each node will be calculated only when their displacement goes beyond the cluster range.

Consequently, robustness of network is becoming a hotspot because the network has characteristics of dynamic topology, limited energy and mobility. Currently, a number of researches on robustness have been done in the mobile Ad hoc network. Reasonable fault management framework can enhance work efficiency of nodes, which can improve the robustness [17,18]. A strong robust network will be utilized in different scenarios.

Currently, the proposed algorithms have taken fault tolerance and energy-efficiency into consideration and make some progress on fault detection and CH election. However, to some degree, only consideration of fault detection and energy-efficiency is unilateral to network, which neglected the mobility of nodes. Therefore, in order to render the energy consumption more homogenized, a robust and energy-efficient weighted clustering algorithm is adopted, which considers the group mobility and the residual energy. Meanwhile, for the purpose of enhancing the robustness of the network, a topology maintenance algorithm is proposed to control when the topology need be maintained. Our algorithm, A Robust, Energy-efficient Weighted Clustering Algorithm (RE²WCA), achieves the tradeoff of energy-efficiency and robustness and the following goals of our algorithm is demonstrated as below:

- (1) For purpose of achieving load balance of cluster head, the clustering algorithm we proposed takes group mobility and residual energy into account to adapt to the change of network. Clustering times will be dramatically reduced due to the group mobility.
- (2) Design a periodic fault detection protocol to exclude the fault node otherwise leading to paralysis of the network. The protocol is based on a clustering structure which integrates the advantages of centralized and distributed scheme.
- (3) Combine topology protocol and clustering algorithm systematically to enhance lifespan and robustness of mobile Ad hoc network.

The structure of this paper is organized as followed: An overview of related work is given in Section 2. A fault detection algorithm and RE²WCA algorithm are described in detail in Section 3. The topology maintenance process is demonstrated in Section 4.1 and the inter-cluster and intra-cluster communication is demonstrated in Section 4.2; Simulation results and discussion are provided in Section 5; Section 6 concludes the paper.

2. Related Work

There are various clustering algorithms for mobile Ad hoc network. Among them, lowest relative mobility clustering algorithm (MOBIC) based on mobility is proposed [19]. The algorithm regards the relative mobility between nodes as the criterion of cluster header selection. WCA and DWCA are put forward in terms of weight clustering [20,21]. Although both of them take the mobility, the limited energy and the degree of nodes into consideration, it is also not involved in how to enhance the robustness and anti-attacks ability. Based on energy-efficiency of network, HEED forms

clusters through a distributed scheme to render the energy consumption of network communication minimized [22]. The algorithm elects the cluster heads based on the residual energy of nodes, which means that the nodes with high residual energy are more possible to be elected as cluster head than other nodes. Based on fuzzy c-means clustering, an energy efficient CH selection is put forward [22].

The cluster structure is optimized to consume less energy. However, in order to make a right decision, the fuzzy c-means protocol has to collect more information, which actually needs more energy and shortens the lifespan of network. So, the performance of the algorithm is not better due to the optimized structure. In [23], based on residual energy and trust value, a cluster head selection is put forward. The trust value according to the interactive history and residual energy of each node are calculated to be used as the criterion of selecting cluster head.

In [24], a distributed fault tolerant topology control scheme based on cooperative communication is proposed. In the literature, the author first defined the k -connectivity and constructed a t -spanner with k -connectivity of an arbitrary communication network. The scheme enhances the fault tolerance ability and also takes the advantages of the cooperative communication to reduce the power consumption. Other algorithms also have their concentration on how to construct redundant backbone and k -connected graph network [25–27]. In addition, in order to overcome the malicious attack, trust model through the weighted method with direct trust and recommendation trust is proposed [28]. In the literature, author builds a trust model to measure whether a node is a malicious node or not. However, since the multi-hop recommendation messages forwarding, each node consumed much energy although the security of the network and fault tolerance is enhanced. Especially when the route is long, the transmission delay and throughput of the whole network are seriously decreased.

In the literature [29], the author adopts a voting scheme in sparse sensor network to obtain each statue of fault node. However, this scheme based on crowd-sourcing will increase the complexity of algorithm and decrease the network lifetime. Additionally, the algorithm is based on a flat structure and the routing overhead will be amplified. Besides many high complexity algorithms are presented to solve the problem of fault node with a flat structure instead of clustering [30–32], however, it also results in energy depletion of the sensors nodes. In [33], a full 2-connectivity restoration (F2CRA) algorithm and partial 3-connectivity restoration algorithm (P3CRA) algorithm for connectivity restoration are put forward. The topology quality and network coverage are taken into consideration when enhancing the fault tolerance and restoration. Unfortunately, the algorithm is executed in a flat structure so that when the network scale becomes large, the complexity of the calculation is increasing dramatically. Consequently, the P3CRA algorithm is not suitable for the large scale wireless sensor networks. In summary, these above algorithms sacrifice the validity for the sake of reliability. Therefore, a method and protocol which does not increase the connectivity of the network but enhances the fault tolerance ability with a hierarchical structure are motivated.

3. Description of RE²WCA Algorithm

3.1. Energy Consumption Model

The battery power of each nodes in network are assigned identically. Equations (1) is adopted to calculate the energy consumption of sending data and Equation (2) for receiving data [15]:

$$\begin{cases} E_t(d, l) = lE_{elec} + l\epsilon_{fs}d^2 & \text{if } d < d_0 \\ E_t(d, l) = lE_{elec} + l\epsilon_{amp}d^4 & \text{if } d > d_0 \end{cases} \quad (1)$$

$$E_r(l) = lE_{elec} \quad (2)$$

In Equations (1) and (2), E_{elec} is required energy for activating the electronic circuits; if the transmission distance is less than the threshold d_0 , the consumption of energy amplification adopts the free space model. If the transmission range is greater than the threshold d_0 , the energy amplification

adopts the multipath attenuation model. ε_{fs} and ε_{amp} are required energy for amplification of transmitted signal to transmit one bit in open space and multipath models respectively; l denotes the number of bits of data; d is the distance between two nodes; $E_t(d, l)$ is the energy consumption of the sending end; $E_r(l)$ is the energy consumption of the receiving end. The residual energy [6] of each node is calculated by Equation (3):

$$E_i^{residual} = E_i^{initial} - E_t(d, l) - E_r(l) \quad (3)$$

In Equation (3), $E_i^{initial}$ is the initial or max energy of each node i ; $E_i^{residual}$ demonstrates the residual energy of each node i after transmitting data.

When each node begins to send data packets, the summation of energy in each cluster can be calculated. In this paper, one of the significant threshold is topology maintenance energy threshold value. When the average of CHs' energy lower than this value, the re-clustering process will be implemented.

Taking the difference of energy of each cluster into account, the average energy within cluster is implied as the inner cluster average energy. Only when the residual energy of cluster head is lower than the value, the process of new cluster head selection will start. Therefore, Equation (4) is adopted to calculate the inner cluster average energy. In Equation (4), C_k represents each cluster and if a node belongs to one of the clusters, then $j \in C_k$.

$$E_k^{ave} = \frac{\sum_{j \in C_k} E_j^{residual}}{\sum_{j \in C_k} i_j} \quad (4)$$

3.2. Mobility Measurement

MOBIC [19] is a mobility-aware clustering scheme. Instead of using the node's ID, MOBIC adopts the relative metric to measure the mobility of the nodes. However, if mobile nodes move in an arbitrary manner, the diversity with other nodes is dominant. Even if some nodes have similar motion but a few others move differently. It still results in dramatic increase in the variance. It shows the variance is greatly influenced by the diverse motion.

In the paper, a group mobility protocol is adopted to measure whether a node has possibility to be elected as cluster head. The higher group mobility of one node implies it has the similar mobility pattern with the majority of its neighbors [14]. Thus, it suits to be elected as CH. Figure 1 shows the derivation process and motion model of the nodes.

Based on record of last movement, the linear displacement of each node can be calculated in real time. The linear displacement of each node will be figured out as soon as nodes move in the network.

Then, the relative direction and speed ratio are considered to calculate the spatial dependency between nodes. In addition, the history cache can be updated.

As is shown in Figure 1, the displacement of node i is lower than D_{thr} and consequently this record will not be updated into the history cache. From moment T to t' , the displacement D' of node j is greater than D_{thr} and consequently the history cache is updated. In addition, from moment t' to t , movement of node j D is still greater than the D_{thr} , hence the record will be entered into the history cache.

Hence, only when node moves, the calculation process will be implement and not all nodes need to save their record into cache due to the cluster radius threshold. Nodes which have the frequent and great displacement will consume more energy for their mobility. Therefore, it is small probability of these nodes to be elected as CHs. Within a cluster or a group, the CH needs to be rotated for the homogenization of energy. Here, we define the spatial dependency with CH as CHSD which demonstrates that each round cluster members need to calculate the spatial dependency with current

CH. In order to describe the spatial dependency with CH (CHSD), the following measurements will be demonstrated.

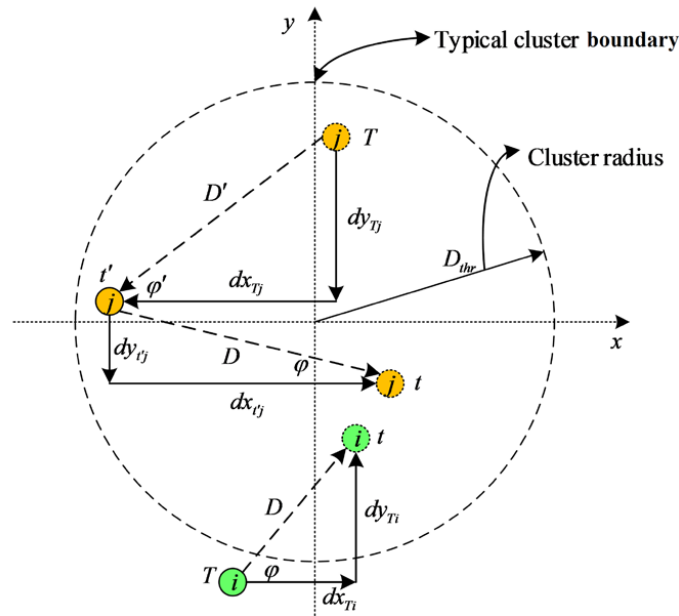


Figure 1. Nodes displacement schematic.

Table 1 gives the detailed explanation of parameters used in Figure 1.

Table 1. Parameters Explanation.

Parameters	Explanation
D_{thr}	Cluster radius
D	Distance of displacement
φ	Angel of direction
dx	Lateral displacement
dy	Longitudinal displacement
$\hat{S}(t)$	Speed of node at one moment t
$\hat{\theta}(t)$	Direction of node at one moment t

According to Algorithm 1, relative direction (RD) and speed ratio (SR) at one moment t between one node i and cluster head are measured respectively as

Algorithm 1 Group mobility computing process

$$dx_{Ti} = x_{ti} - x_{Ti} \quad dy_{Ti} = y_{ti} - y_{Ti}$$

where t is the current time and $x_{ti}, x_{Ti}, y_{ti}, y_{Ti}$ are the coordinates of the node i from time T to t , respectively.

Hence the displacement and angle can be calculated by:

$$D = \sqrt{(dx_{Ti})^2 + (dy_{Ti})^2} \quad \tan \varphi = \left| \frac{dy_{Ti}}{dx_{Ti}} \right|$$

If $D \geq D_{thr}$, the most recent record of speed and direction that will be saved into the history cache are:

$$\hat{S}_i = \frac{D}{t - \hat{T}_i} \quad \hat{\theta}_i = \begin{cases} \varphi \cdot \text{sgn}(dy_{Ti}) & dx_{Ti} > 0 \\ \pi/2 \cdot \text{sgn}(dy_{Ti}) & dx_{Ti} = 0 \\ (\pi - \varphi) \cdot \text{sgn}(dy_{Ti}) & dx_{Ti} < 0 \end{cases}$$

where $T = \hat{T}_i$ be the moment when the last record was saved for node i and $\hat{\theta}_i \in (-\pi, \pi)$. The time \hat{T}_i ,

the speed \hat{S}_i , the direction $\hat{\theta}_i$ and the present location of node i will be recorded into the history cache with

$$x_{Ti} = x_{ti}, y_{Ti} = y_{ti}, \hat{T}_i = t.$$

$$RD(i, ch, t) = \cos(\hat{\theta}_i(t) - \hat{\theta}_{ch}(t)) \quad (5)$$

$$SR(i, ch, t) = \frac{\min(\hat{S}_i(t), \hat{S}_{ch}(t))}{\max(\hat{S}_i(t), \hat{S}_{ch}(t))} \quad (6)$$

Hence, the *CHSD* can be calculated by:

$$CHSD(i, ch, t) = RD(i, ch, t) \times SR(i, ch, t) \quad (7)$$

The range of *CHSD* is between $[-1, 1]$. If *CHSD* is very close to 1, it indicates the nearly same motion with *CH* otherwise the less similarity with *CH*. *CH* has the feature of maximum similarity and lower mobility and it is elected in the last round. However, in the current round, *CH* is a very ordinary node and it still needs to calculate the speed, the angle. But nodes now will not compare with each other, instead, with *CH*. It is the improvement compared with original measurement. It dramatically decreases the complexity of calculation and reduces great energy consumption. Additionally, the maximum of *CHSD* will be appropriate to be elected as *CH*. Therefore, in order to describe the stability of one cluster, it only needs to calculate the summation of *CHSD*.

3.3. Weight Model and Hybrid Clustering Algorithm

After the initialization of the network, one node within cluster announces the information of its cluster head. The cluster head needs to be rotated to prolong the survivability and lifespan of the network. Therefore, we put forward the weight probability average. The high weight value of node implies it has more possibility to be selected as cluster head. Hence, a weight model is proposed and its measurement is given by Equation (8):

$$weight(i) = C_{prob} \times \left[a \times \frac{E_i^{residual}}{E_i^{initial}} + b \times CHSD_{map}(i) \right] \quad (8)$$

where a and b are weighted coefficient index of residual energy and group mobility feature respectively; $CHSD_{map}(i)$ is the spatial dependency with *CH* at one moment after linear mapping from $[-1, 1]$ to $[0, 1]$; $weight(i)$ denotes one node probability to be *CH*; C_{prob} is an initial iteration probability.

It is obvious that the range of $weight(i)$ value is limited with $(0, 1)$. Additionally, the summation of a and b is set as 1. Generally, $a > b$ is rendered. The adequate residual energy of the node implies that it can manage a cluster better than other nodes. In addition, the high group mobility of *CHSD* demonstrates that one of node within cluster has a high correlation with its cluster head. Hence, in order to make the times of re-clustering due to nodes' mobility, the node with a high *CHSD* value is elected as the cluster head in the group. In addition, we use C_{prob} to limit the announcement of cluster head. Through calculating the weight value above, it is concluded that the nodes with high group mobility and residual energy is more suitable to be the cluster head.

Through limiting iteration of cluster head, a *CH* selection protocol is put forward. In order to prolong the lifetime of the network, we regard *average minimum reachability power (AMRP)* as the cost of election. Compared with the other types of cost, such as *node degree* or *1/node degree*, the *AMRP* is a compromise between network density and load balancing. It has been proved that HEED has fewer iterations than General Cluster (GC) Scheme [34]. In this paper, we consider a greater minimum iteration threshold p_{thr} . Due to the mobility, the radius of cluster or group will be greater. However, with the increase of the radius, the iterations of GC are dramatically increased but HEED has to stop iterating for the p_{thr} . Considering mobility of nodes, increasing p_{thr} appropriately will reduce the iterations so that save energy. Equation (9) demonstrates the calculation:

$$CH_{prob} = \max(weight(i), p_{min}) \quad (9)$$

where CH_{prob} represents the probability of each node to be CH. Through a limited iteration, CH will be elected. In every iteration, the CH_{prob} will be doubled and until the cluster formation accomplished or finished beyond the threshold. The iteration times can be calculated:

$$N_{iter} \leq \left\lceil \log_2 \frac{1}{p_{min}} \right\rceil + 1 \quad (10)$$

Therefore, $N_{iter} \approx O(1)$. If p_{thr} sets to 10^{-4} , there is 15 iterations to end up. Actually, the greater number of iterations will consume much energy, thus it is necessary to select an appropriate iteration threshold. We define the ratio of residual energy of CH and average energy of cluster is ER (energy ratio) as Equation (11) described:

$$ER = \frac{E_{ch}^{residual}}{E_k^{ave}} \quad (11)$$

In addition, $p_{min} = \max\{ER, p_{thr}\}$ is chosen. For Equation (11), if residual energy of cluster head is equal to the average energy of cluster approximately, ER is about 1. Therefore, the times of iteration is limited within 2. Considering an extreme case, if when the residual energy of CH is close to 10^{-3} J (a totally unsuitable residual energy) and cluster average energy is 1 J, then $ER = 10^{-3}$. Thus, if we set initial $p_{thr} = 10^{-2}$, then the iteration times will be limited within 7 rather than 10. Therefore, if initial p_{thr} is greater than ER, p_{thr} will be chosen as the iteration threshold. Considering energy efficiency, if the residual energy of cluster head is enough, the iteration times of cluster head can be reduced. On the contrary, if the energy of cluster head is not adequate, the network will need more iteration times to update the perfect cluster head. In conclusion, according to Algorithm 2, the process of cluster head election is demonstrated as follows:

Algorithm 2 CH election algorithm (each round for all nodes within the network)

Function: $weight(i) = C_{prob} \times \left[a \times \frac{E_i^{residual}}{E_i^{initial}} + b \times CHSD_{map}(i) \right]$; $CH_{prob} \leftarrow \max(weight(i), p_{min})$;
 $p_{min} \leftarrow \max\{ER, p_{thr}\}$;
 Input: $a, b \leftarrow weight(i)$ factors satisfy $a > b$ and $a + b = 1$; $C_{prob} \leftarrow$ assuming value;
 $d \leftarrow$ distance between sending end and receiving end;

1. Begin
2. The sending end radio energy consumption is E_t if $d < d_0$, the free space model is used, otherwise, multi-path attenuation model is used
3. The receiving end radio energy consumption is $E_r(l) = lE_{elec}$
4. $E_{residual} \leftarrow$ the difference between initial energy and the summation of sending and receiving energy
5. $\hat{\theta}_i, \hat{\theta}_{ch}, \hat{S}_i, \hat{S}_{ch} \leftarrow$ group mobility process. $RD(i, ch, t) = \cos(\hat{\theta}_i(t) - \hat{\theta}_{ch}(t))$; $SR(i, ch, t) = \frac{\min(\hat{S}_i(t), \hat{S}_{ch}(t))}{\max(\hat{S}_i(t), \hat{S}_{ch}(t))}$
6. $CHSD(i, ch) \leftarrow RD(i, ch) \times SD(i, ch)$ at one moment t ; then a linear mapping from CHSD to $CHSD_{map}$.
7. Start CH selection protocol to implement CH election.
8. End.

3.4. The Process of Cluster Establishment

In mobile Ad hoc sensor network, this stage constitutes the final step of our novel weight algorithm and represents the construction of the cluster members' set. Each cluster defines its neighbors at one hop maximum, which forms the member of the cluster. Firstly, all nodes are distributed in a network range and assigned with typically identical energy. In the network, each node broadcasts Hello messages packets which will consume a little of energy. Nodes will move towards different direction with different velocity, which will also cost various energy. When nodes are stable in first round, it is time to cluster through calculating the weight value of each node. Those nodes with large weight value will be elected as CH. Since the cluster is only restricted to manage its one hop neighbor, the process of

cluster establishment will be accomplished quickly. Eventually, when all CHs connect with each other, the network is initialized. In order to display the process clearly, we give an explanatory example to clarify the clustering process.

Firstly, we give some relevant parameters. For example, we render $a = 0.7$, $b = 0.3$. It implies that the residual energy attaches more significance than mobility. Actually, the mobility will not very large in sensor network. It is only a small range of displacement; therefore, the history cache operations will not be very frequent. More clearly and directly, only those displacements of nodes beyond D_{thr} , the record will be entered into cache and is used to calculate the next moment displacement. Additionally, the CH is typically identical with cluster members. Hence, it also needs same computing with cluster members. Then, $CHSD$ of one node will be calculated by Equations (5)–(7). This process actually need no data transmission except for the computing of $CHSD$. The data transmission needs only n times (if the number of cluster members is n). Of course, the weighted value can be adjusted according to different needs in application. For example, if the protocol is used in VANET, b can be adjusted larger since the energy is not a vital factor for sensor network. When the initialization of the network, the ratio of residual energy to average energy is equal and speed, directions are same, hence the weighted summation of $CHSD$ and ratio of energy is 1. By Equation (8), we assume C_{prob} is an appropriate value that can reduce the iterations as much as possible. Then all nodes start to announce to form a cluster. Since we consider the cost of AMRP, therefore, each node need to broadcast a packet {ID, CH_{prob} , cost (AMRP)}. According to the cost of broadcasting, the left nodes join in a cluster according to the minimum cost.

In Figure 2a, nodes (A, B, C, D) announce their tentative CH and display their connection with nodes in the first iteration. In Figure 2b, tentative CHs (B, C) quit the job of CH and join in cluster because the cost (AMRP) or CH_{prob} is not suitable. Tentative CH (A, D) is still keeping their job for the group mobility feature ($CHSD$) is relatively greater. In Figure 2c, more tentative CHs are announced. Actually, AMRP will control the hops minimized in order to energy efficiency. After the first iteration, the CH_{prob} will double. Therefore, if CH_{prob} of one tentative CH is greater than 1 or greater, this CH will be the formal CH and does not participate the iteration process but allows another node to join in. This example shows three iterations in CH selection. Actually in the next round, the process will keep longer and will even trigger p_{thr} to finish the iteration forcibly.

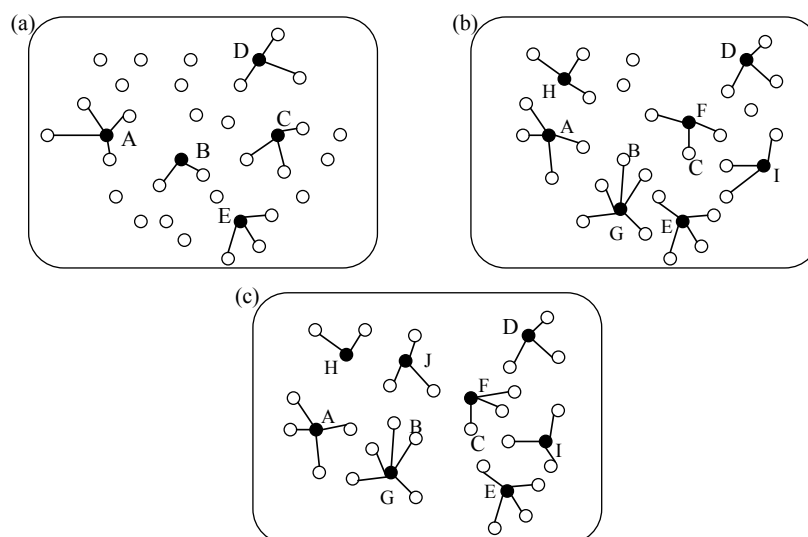


Figure 2. The CH selection process in RE²WCA. (a) Tentative CHs (A, B, C, D, E) form. There are some nodes fail to join a cluster. (b) The tentative CHs (A, D, E, F, G, H, I) are updated and increased, so only a few nodes are at free state. (c) All nodes have their only cluster and tentative CHs become formal CHs (A, D, E, F, G, H, I, J).

3.5. Fault Detection Algorithm

Currently, mobile Ad hoc sensor network is widely utilized and obtains extensive applications for its rapid development, which results in the large quantities of problems emerged for the bad working condition [35–37]. For example, nodes in the network are likely to suffer from battery out of charge for the increasing power consuming rate and the link failure in the network. In Figure 3, the network is established when the clustering algorithm finished. All nodes have its unique cluster and CH. CH is responsible for the intra-cluster communication and inter-cluster packet transmission. Currently, there are mainly two kinds of methods in fault detection for wireless sensor network, which are centralized method and distributed method. In centralized fault detection approach, the central node, commonly sink node, is responsible for the faulty nodes in network. Apparently, these approaches are not suitable for large scale network because huge message forwarding will consume considerable energy and this will surely shorten the lifetime of the network. Therefore, distributed approach is proposed to overcome the problem of centralized approach. In distributed approach, faulty nodes are identified by local decision-making. In our network, the distributed fault detection managers are CHs and fault detection responsibility is subsequently distributed among each individual cluster. Further, if a failure is detected locally, the failure can be propagated to all other clusters. Hence, we propose a distributed approach to construct a fault detection framework to render the network more robust and fault tolerant.

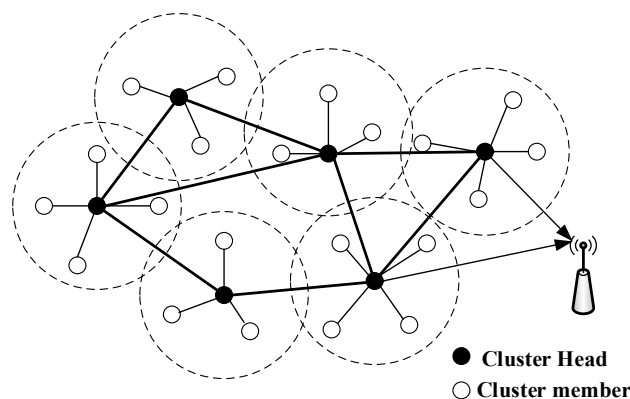


Figure 3. A clustering wireless sensor network model.

Through this hierarchical structure, the management framework is more efficient. CHs need to find a path to forward packets to the sink node. Thus, the optimal path is also vital for the sake of energy efficiency. Consequently, in order to reduce the cases of network paralysis caused by the nodes failures or CH, we propose a fault-detection and CH backup mechanism for the network. A periodic fault detection protocol is considered, which can detect the failures of nodes and CHs efficiently. Additionally, most topology control strategy is centralized [38–40], which consumes amounts of energy and increases the overhead for broadcasting global messages. Therefore, we adopt a distributed fault tolerance control method to avoid the drawbacks of centralized strategy. When the process of cluster establishment is accomplished, the proposed fault detection algorithm is as followed:

The following example figures the process of this algorithm within each cluster under different cases. Among them, Figure 4 gives the practical example of fault node detection. Meanwhile, Figure 5 shows the practical process of Backup CH mechanism. We consider a low group mobility since actually most nodes have the similar movement characteristics.

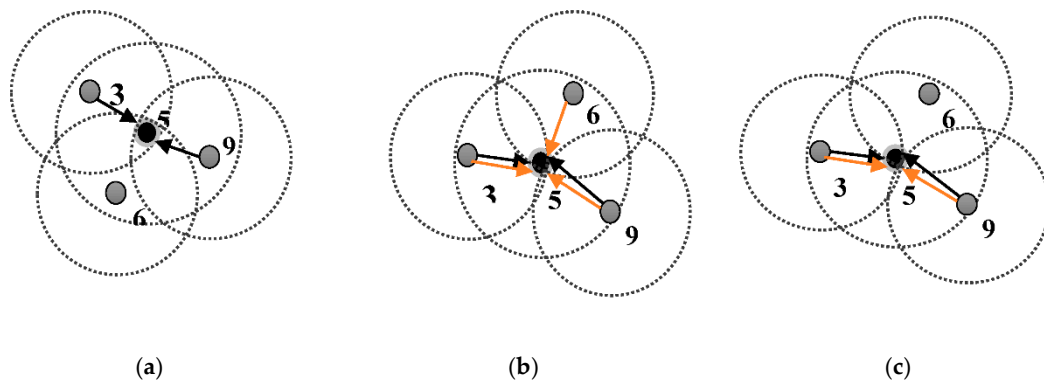


Figure 4. (a–c) is the practical example of fault node detection process. (a) Cluster members (3, 9) send message to CH. Node 6 fails sending, hence node 6 is to be probable fault node. (b) Nodes (3, 6, 9) send message to CH respectively. Node 6 is normal. (c) Nodes (3, 9) send message to CH respectively. Node 6 still fails to send messages.

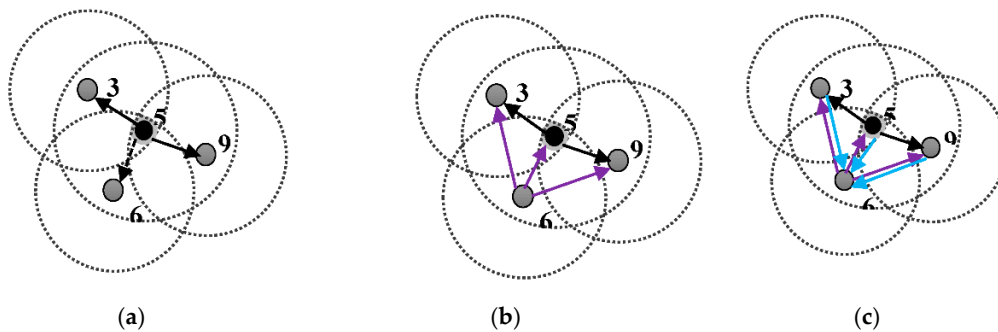


Figure 5. (a–c) is the practical example of Backup CH mechanism process. (a) In the period of $time_fn2$, nodes (3, 9) receive the message from CH (5) but node 6 not. Node 6 marks CH possibly failed. (b) Backup CH launch the status enquiry message for cluster. (c) Backup CH receive the messages from inner-cluster nodes and judge CH is faulted or not.

The above two processes are periodic. The period of detecting the fault node is $time_fn1$. Actually, the period is equal with the time when the last cluster member finish sending packets for the purpose of not influencing the normal communication of nodes. Thus, every time the last cluster member nodes finish sending packets, the network will run the fault detection algorithm. Similarly, the period of CH detection $time_fn2$ is equal with the time which consists of $time_fn1$ and the data processing time. Backup CH broadcasts sensing messages to cluster member and neighbor cluster nodes periodically. If finds CH cannot work, then execute the algorithm of Backup CH to make the network work. When the network cannot work normally, it needs to trigger the topology maintenance. In Algorithm 3, the detailed description of fault detection algorithm is shown.

Algorithm 3 Fault detection algorithm description (run between inner-cluster and intra-cluster)

After the packet transmission is finished, the fault detection begins. *CH* is elected and the degree (dv_j) of each *CH* is ensured.

Initialization: $k_i = 0$, $\sum i = N$, $\sum j = H$, $S_{\text{fault}} = \phi$, $S_{\text{prob_fault}} = \phi$, $S_{\text{prob_fault}}(j) = \phi$.

#Round m (period time $_{fn1}$)

```

1.  Initialization( $k_i$ ,  $S$ );
2.  whi( $v_i \in \text{Cluster}_j$  and  $j \leq H$  and  $i = N$ )
3.  for  $i = 1$  to  $dv_j$ 
4.       $\text{bool flag} = \text{Send}(v_i \rightarrow \text{ID}, v_i)$ 
5.      if ( $\text{flag} == \text{false}$ )
6.           $\text{Append}((v_i \rightarrow \text{ID}), S_{\text{prob\_fault}}(j))$ 
7.           $k_i++$ ;  $i++$ ;
8.      end;
9.  end;
10.  $j++$ ;
11. end;
12. if ( $k_i \geq 2$ )
13.     $\text{Append}((v_i \rightarrow \text{ID}), S_{\text{fault}})$ 
14. else
15.     $S_{\text{prob\_fault}} = \cup_{j=1}^H S_{\text{prob\_fault}}(j)$ ;
16. return  $k_i$ ,  $S_{\text{prob\_fault}}$  and  $S_{\text{fault}}$ ;

```

#Round m (period time $_{fn2}$)

```

1.  Input ( $\text{Cluster}_j$ ,  $\text{weight}$ );
2.  Select_BackupCH ( $v_i \rightarrow \text{weight} < \text{CH} \rightarrow \text{weight}$ )
3.  while ( $v_i \in \text{Cluster}_j$  and  $j \leq H$  and  $i = N$ )
4.      for  $j = 1$  to  $\text{Cluster}_j$ 
5.           $\text{bool flag} = \text{Send}(\text{Cluster}_j \rightarrow \text{ID}, v_i)$ 
6.          if ( $\text{flag} == \text{false}$ )
7.               $\text{mark}(v_i \rightarrow \text{ID})$  for each cluster
8.               $\text{Append}(\text{Cluster}_j \rightarrow \text{ID}, S_{\text{fault\_prob}})$ 
9.          end;
10.     end;
11.     for  $j = 1$  to  $\text{Cluster}_j$ 
12.          $\text{Send}(v_i \rightarrow \text{ID}, \text{Cluster}_j \text{ and } v_i)$ 
13.          $\text{bool flag} = \text{Receive}((\text{Cluster}_j \text{ and } v_i) \rightarrow \text{ID}, v_i)$ 
14.         if ( $\text{flag} == \text{false}$ )
15.              $\text{Append}((\text{Cluster}_j \rightarrow \text{ID}), S_{\text{fault}})$ 
16.         end;
17.     end;
18. end;

```

3.6. Complexity

The time complexity of *CH* election algorithm is $O(MN)$, in which M is the rounds number and N is the total number of nodes. When each round goes on, every node within the network will calculate its value of *CHSD* in order to compete with other nodes to be the next *CH*. So, when the times of rotation is M , the time complexity is $O(MN)$. The most complex operation in the Fault detection algorithm is the rotation computation and it runs in $O(N \log(N))$. This operation is performed for each cluster, which gives $O(KN \log(N))$, in which K is the number of clusters. Then, the time complexity of our proposed algorithm is $O(MN) + O(KN \log(N))$.

4. Communication and Topology Maintenance

4.1. Topology Maintenance

Topology maintenance is based on the fault nodes detection and *CHs* backup mechanism. Once finds fault nodes or failure *CH* nodes, then it needs to cluster again. From this perspective, the network of clustering is not periodic but on demand. However, because of node's mobility, if the network is normal after a certain period of time, the topology maintenance process is automatically conducted. The timing sequence of our protocol is presented as Figure 6 shows.

In set up phrase, cluster formation process is to be conducted, that is, our proposed protocol is executed at this period. Actually, the time consumed is relied on the moment that the cluster formation accomplishes. If the energy of the whole network is extremely little and all nodes are going to die, the clustering formation will be longer for the greater times of iteration. In steady phrase, nodes are occupied with collecting data and transmission. Apparently, in this time phrase *CH* will consume much energy for the routing of nodes' message. Due to the mobility of nodes, the routing is a hard problem. However, after the cluster formation considering group mobility, the

topology will be relatively stable. Therefore, the overhead of routing will be reduced. During the period of *time_fn1*, fault detection algorithm will be implemented. In *time_fn2*, topology maintenance is executed. The time overhead is not stable which is dependent on the different application. If network is deployed in a well environment, the lifetime of each node will be longer than those malicious or bad environment. The fault detection algorithm can exclude “dead” node from the network. Here, “dead” nodes are relatively fewer. We mainly consider how to reduce the network partition and nodes failure. This consideration is also a way of topology maintenance. A stable topology will benefit decreasing overhead of routing and clustering. Of course, our proposed clustering algorithm mainly focuses on the clustering algorithm, however, in order to demonstrate routing problem with our protocol, we also need to clarify the reasonability of the proposed algorithm in routing finding aspects.

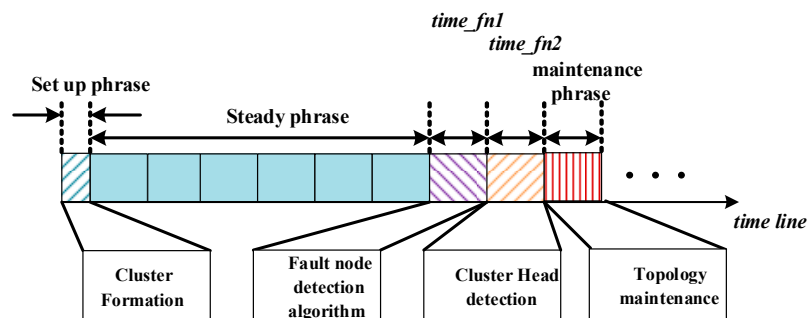


Figure 6. Round of network set up.

Algorithm 4 gives the detailed process description of cluster maintenance algorithm. Through topology maintenance process, the cluster topology is relatively stable and the performance of the whole network can be guaranteed, such as throughput, lifespan and so forth. The network does not paralyze for the CH fault or nodes failure. The QoS of the network can be enhanced through a robust CH and stable topology. Actually, the fault detection can extend into a secure field for the similarity of the trust value of nodes. Currently, the trust value computing is a common method to estimate whether a node is good or bad. The realization of this kind of trust estimation, which can enhance the security of the network, is implemented through sensing among nodes or interception [24,28,41].

Algorithm 4 Cluster Maintenance Algorithm

Step 1: When cluster members finish sending packets, network fault detection process begins, if cannot detect any problems, then go to step 5; otherwise execute sequentially.

Step 2: If finds the CH failure, then go to step 4; otherwise execute sequentially.

Step 3: The RE²WCA algorithm is conducted according to the information of all cluster members. If fault node is backup CH, backup CH is obliged to be elected again according to weight value among cluster members. Then jump to step 5.

Step 4: Backup CH takes over the duty of CH firstly and broadcasts messages overall the network. If CH is permanent fault node, the ID will be deleted from the network; otherwise to be cluster member.

Then, the RE²WCA algorithm is conducted to make sure the network re-cluster again. Then, jump to step 5.

Step 5: Withdraw from topology maintenance process.

The robustness of the network is enhanced when the fault detection and backup CH mechanism are introduced. These two algorithms are not confronted within a special circumstance. As long as it detects the fault node, the network can be fixed up through the topology maintenance algorithm. In RE²WCA, based on nodes' residual energy and group mobility, the CHs are elected and rotated more reasonably which renders the load balanced and the lifetime prolonged. On the other hand, in a totally distributed clustering approach, nodes do not need to broadcast global messages to reach the target of local topology maintenance. It certainly decreases the overhead so that can adapt to the network well.

4.2. Inter-Cluster and Intra-Cluster Communication

In the description of RE²WCA, we assume single-hop communication among CHs and their registered cluster members. This is desirable in source-driven networks, where reports are periodically transmitted by nodes. Hence, a Time Discrete Multiple Address (TDMA) frame may be constructed at each CH to eliminate interference within a cluster. Obviously, constructing TDMA frames need well synchronization and in a lightly-loaded network, adopting TDMA frames may waste frequent resources. Therefore, a better way is to use channel contention. Multipath routing can increase channel capacity in this case. In intra-cluster communication, in order to reduce the collisions of data, each CH will create a TDMA scheme for its nodes and broadcasts it among cluster members. Every node sends its data to its CH according to specified TDMA schedule. In order to avoid the interference caused by signals from a nearby cluster, CH can set the Code Discrete Multiple Address (CDMA) code of all nodes within the cluster. The CDMA code, along with the TDMA, are sent at scheduled times [2,3,19,34]. However, in intra-cluster communication, for example if two nodes from different cluster need to communication, then the multi-path routing may work better. Clearly, the collisions caused by data forwarding are nearly unavoidable, therefore the capacity and throughput will decrease dramatically. Additionally, if it adopts a way of saving stable routing table, the great communication overhead will degrade the network performance. In order to solve this problem, some routing protocols are proposed, such as Directed Diffusion or Dynamic Sourcing Routing (DSR) [31,36]. The protocols are mainly to reduce communication overhead through increasing the computing cost since computing cost is relatively much less compared with communication overhead. As is shown in Figure 7, TDMA schemes are adopted for the intra-cluster communication. Meanwhile, as is shown in Figure 8, the inter-cluster communication adopts the CDMA schemes.

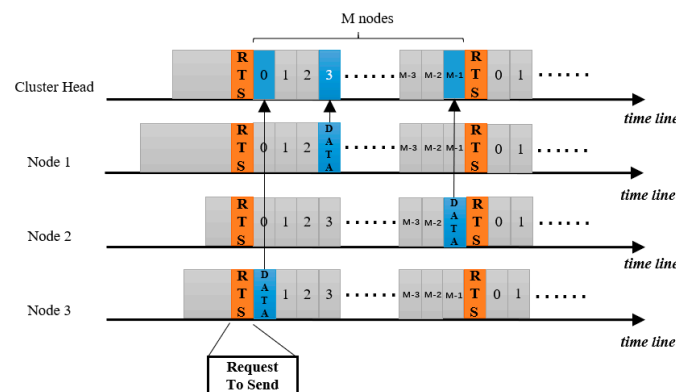


Figure 7. TDMA schemes adopted for the intra-cluster communication.

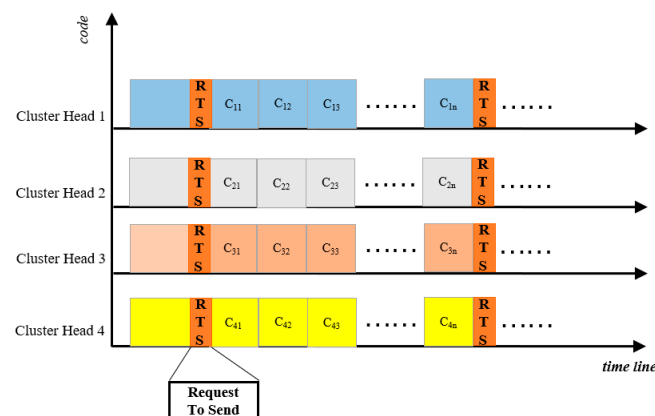


Figure 8. CDMA schemes adopted for the inter-cluster communication.

In RE²WCA, the proposed two fault tolerant algorithms which are fault detection algorithm and CH backup algorithm respectively need to consider the synchronization. When CH find fault node, it needs to transmit data to sink node through inter-cluster communication. If some clusters failed to accomplish the process of fault detection, then the data forwarding cannot succeed. In Figure 6, only when all nodes finish the detection process during *time_fn2*, the topology maintenance will be triggered. Hence, a suitable way is to send Hello message packets among all CHs to ensure every node is in a waiting state (in fact has nothing to do). Actually, the waiting of other CHs will degrade the instantaneity of the network because if just one CH has not implemented the fault detection, all nodes need to wait. But if a cognitive routing algorithm is adopted, the synchronization will be solved. The CH can find a best dynamic route to transmit the data. However, whatever the routing protocol has been adopted, the synchronization should not be a hard problem. On the one hand, the time waiting is not worthy of paying much attention because when CH are in waiting state, the energy consumption is nearly limited to zero. Actually, this is a tradeoff between instantaneity and reliability. On the other hand, the great difference among clusters is not common in real situations. After sink node receiving the fault data CHs forward, the topology maintenance process will be triggered. Sink node collects the fault information of nodes so that upload to higher class network. Meanwhile, sink node will broadcast packets to relevant clusters to tell them re-clustering.

5. Simulation Results and Discussion

Firstly, the performance of RE²WCA algorithm is evaluated. For purpose of generality, it is assumed that one thousand nodes are placed into a field which ranges 2000 m × 2000 m. In addition, the iteration threshold is set to 0.0005 [10,11,13]. Equation (10) implies that the max iteration times of RE²WCA protocol is 12. In the process of initialization, $CH_{prob} = C_{prob} = 5\%$ is set for all nodes in the network, where only in sixth iteration, nodes with high energy will exit RE²WCA protocol. It means that if the residual energy of nodes is higher, they will terminate RE²WCA more earlier. The performance of DWCA and EDWCA are evaluated to demonstrate more straightforward and convincing. Therefore, from the aspects [3,10] listed as follows, RE²WCA is compared with EMDWCA and DWCA protocols. (1) Ratio of the number of clusters to the number of all nodes; (2) Ratio of non-single node clusters to the number of clusters; (3) Standard deviation of the number of nodes in a cluster and maximum number of nodes in a cluster; (4) Average residual energy of CHs elected in each cluster.

Figure 9a shows that both in three algorithms, when the radius of cluster increases, the percentage of cluster heads decrease respectively. When the radius of cluster reaches 400 m, it is not surprised that only one cluster head exists in the mobile network. With cluster radius increasing, the percentage of cluster heads of RE²WCA is larger than DWCA and EDWCA algorithms. Because group mobility is considered where the nodes with similar motion pattern can maintain the original structure of cluster when the network topology changes. Thus, the number of cluster heads can be decreased a little slowly in RE²WCA protocol. In Figure 9b, RE²WCA protocol does not perform very well compared with DWCA and EMDWCA. When the radius of cluster reaches 25 m, the average residual energy of three protocols are nearly same. Figure 9b implies that with the radius of cluster increasing, the average residual energy increases. However, the residual energy of DWCA and EDWCA protocols is higher than RE²WCA. The difference between RE²WCA and EMDWCA is about 13%. Compared with DWCA is approximately 10%. It demonstrates that the CH residual energy selected by RE²WCA is high and is not far lower than the EMDWCA and DWCA.

Application requirement meets different types of cost function when selecting CH. If the protocol needs to be utilized in high node density network, the standard deviation will larger. If considering load balance, the standard deviation will be much smaller. In EMDWCA, the protocol considers the *1/node degree*. Clearly, the maximum degree cost will increase the standard deviation. However, DWCA considers the minimum degree cost, that is, *node degree*, which demonstrates the standard

deviation is relatively smaller. In RE²WCA, we consider a compromise between two extremes which is load balancing and cluster density respectively.

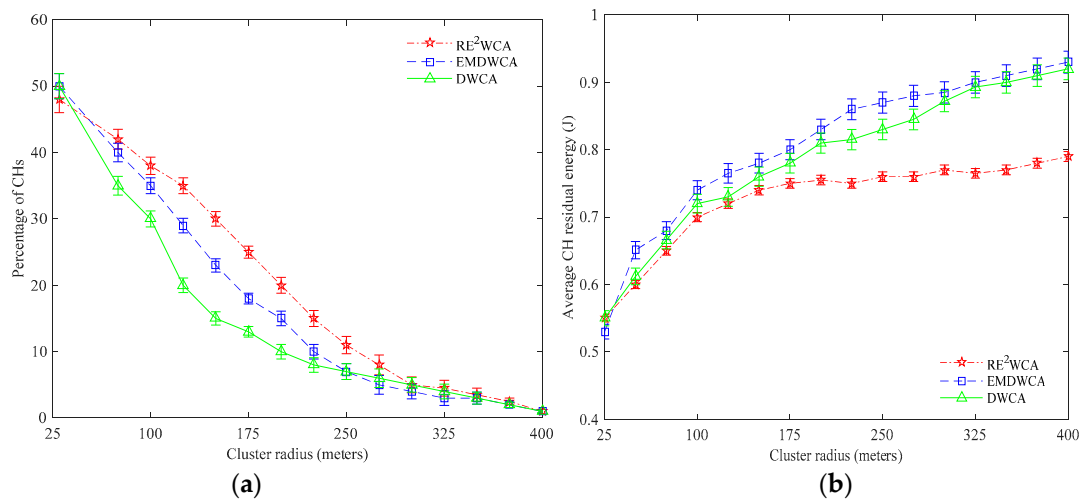


Figure 9. Features of elected CHs. (a) Percentage of CHs. (b) Average residual energy per CH.

Another vital property of cluster is percentage of non-single node clusters. It demonstrates the feature of minimize the node into a cluster. In Figure 10b, the percentage of non-single nodes of three protocols increase with the radius of cluster becomes larger. Theoretically, if the radius of cluster is large enough, the percentage of non-single nodes nearly reaches to 100% with all nodes forming into one cluster. It also implies that the percentage non-single nodes of RE²WCA is much higher compared with two other protocols. From Figure 1c, we can see the ratio of maximum number of nodes RE²WCA to EMDWCA is lower than other two types of cost. The number of nodes will surely larger than minimum node degree and AMRP. In fact, this figure shows similar facts with Figure 10a. They all demonstrates the different types of cost will form into various clusters. More requirements need considering in the actual application. Figure 10d illustrates that when the node speed increases, the rate of CH change goes high since the topology of network changes dramatically. Considering group mobility of network, the performance of RE²WCA is better than DWCA and EDWCA protocols. The rate of CH changes in RE²WCA is nearly lower 7.5% than DWCA, 2.5% than EDWCA.

In addition, our protocol can be used into a cluster structure routing protocols, in which higher tier nodes should have more residual energy. Our approach can also be effective for sensor applications requiring efficient data aggregation and prolonged network lifetime. In the simulations, most of parameters are similar to those in [2,19,20,34]. Additionally, the mobility speed of nodes is 10m/s which is also similar with Ref [14,42,43]. The parameters are listed in Table 2. The energy consumption computing of sensor nodes is on the basis of Equations (1)–(3). Except for the above parameters, without loss of generality, the application parameters are as followed: data packet size of 100 bytes, broadcast packet size of 25 bytes and packet header size of 25 bytes [3,10,34]. Additionally, we assume $a = 0.7$, $b = 0.3$ as the weighted average index. The reasonability has been discussed before. Firstly, all network has to solve the problem of throughput. For the purpose of increasing the throughput of the network, clustering method for nodes is adopted.

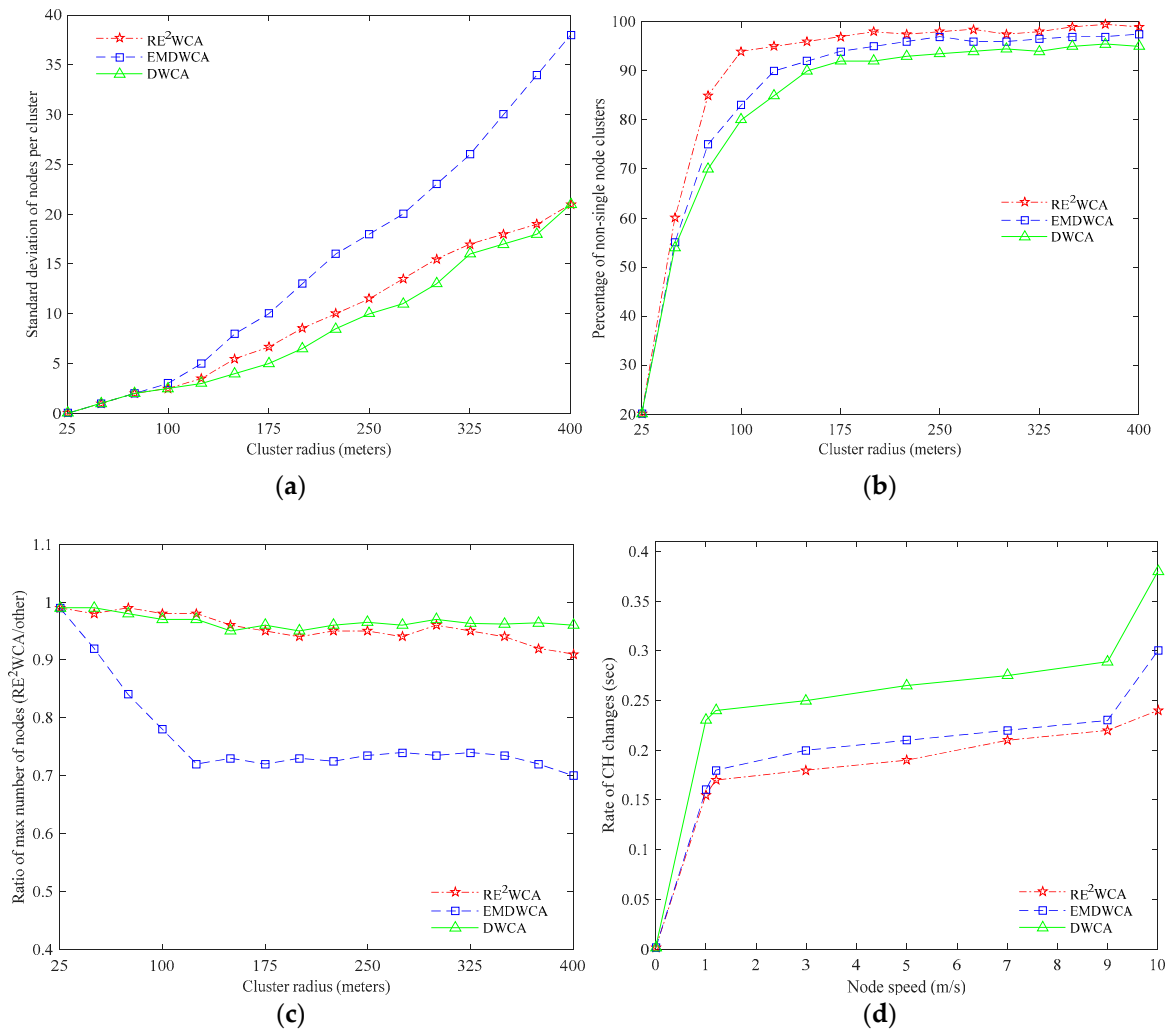


Figure 10. Features of clusters. (a) Standard deviation of the number of nodes/cluster. (b) Percentage of non-single clusters. (c) Ratio of maximum number of nodes in RE²WCA to other two types protocol. (d) Rate of CH changes with node speed varying.

Table 2. Simulation Parameters.

Parameters	Significance	Value
N	Total numbers of nodes	50
$m \times n$	Network size	100 m \times 100 m
$MaxSpeed$	Maximum movement speed	10 m/s
$time_{fn1}$	Fault node detection period	20 min
$time_{fn2}$	CH fault detection period	30 min
$Round$	data transmission frequency	5 TDM frames
T	Simulating time	800 min
$E_{initial}$	Initial energy	2 J
E_{elec}	Activating energy	50 nJ/bit
E_{fusion}	Energy consumption of data fusion	5 nJ/bit/signal
ε_{amp}	Amplification coefficient ($d > d_0$)	0.0013 pJ/bit/m ⁴
ε_{fs}	Amplification coefficient ($d < d_0$)	10 pJ/bit/m ²
d_0	Threshold distance (d_0)	75 m
D_{thr}	Cluster radius	25 m

In Figure 11, the first node of the DWCA algorithm failed at 196 rounds and the first node of the EMDWCA algorithm failed at 331 rounds, while the first node of the RE²WCA algorithm failed at 398 rounds; the nodes that survive with the RE²WCA algorithm is 103% higher than that of the DWCA algorithm and 20% higher than that of the EMDWCA algorithm. We also know that, when using the DWCA algorithm, all nodes are dead after 721 rounds of cluster head election. By applying the EMDWCA algorithm the life cycle of the node is 1270 rounds, however, by using RE²WCA algorithm the surviving nodes remain to exist after 1410 rounds. This is because the algorithm proposed in this paper not only considers the residual energy and the mobility of the node but also takes the maximal capability of cluster head handling into account, which makes the nodes' energy consumption more homogeneous and protects the cluster heads from excessive consumption. So, the RE²WCA algorithm improves the stability of the network topology and the survivability of the overall network.

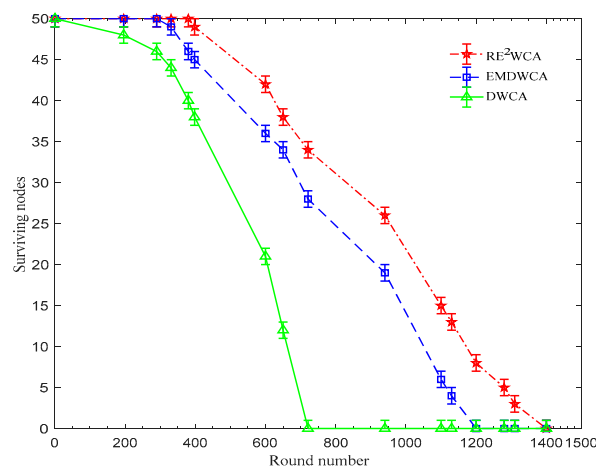


Figure 11. Surviving nodes vary with Round number.

Figure 12 demonstrates the average network throughput varies with the distance of nodes. Within a short distance, the RE²WCA is similar to the DWCA and EMDWCA, however, in the long distance, the throughput of RE²WCA is obviously greater than the DWCA and EMDWCA since the proposed algorithm can elect the optimal CH, which can accelerate the message forwarding. The throughput is nearly 10% greater than the EMDWCA and nearly 15% greater than DWCA respectively.

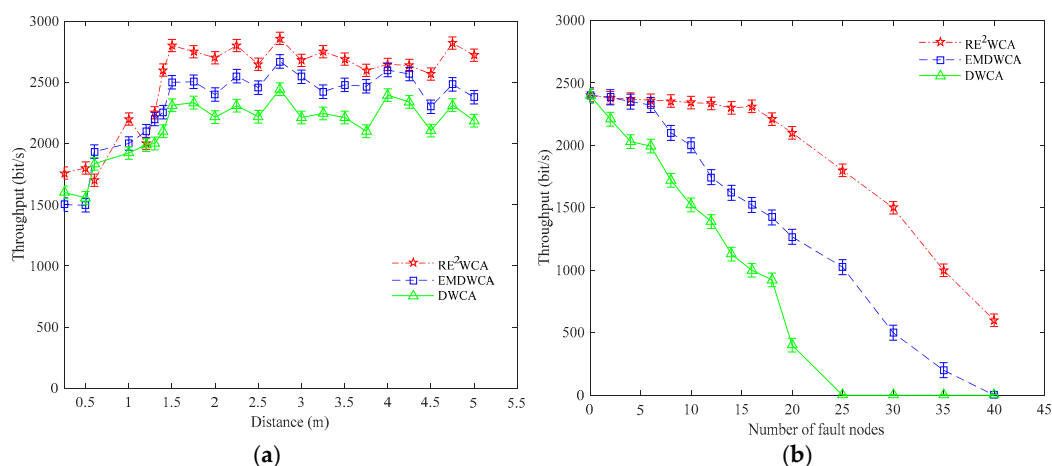


Figure 12. Throughput of the network. (a) Network throughput varies with the distance. (b) Network throughput under certain fault nodes.

In Figure 12, with the number of fault nodes increasing, the throughput of DWCA is decreased rapidly. On the meanwhile, the throughput of EMDWCA and RE²WCA decreased slowly since in order to increase robustness of network, both of them introduce fault nodes detection algorithm. In RE²WCA, the redundancy of the network is not increased but a mechanism of fault detection is introduced. The EMDWCA increases the redundant links of the nodes, which increases the routing overhead and consume more energy to construct links. Therefore, the throughput will be degraded. When the proportion of fault nodes reaches to 16%, the throughput of three types protocols decreased dramatically since many clusters cannot work normally. Actually, the proportion of fault node cannot reach so highly, thus, RE²WCA remains to maintain a better throughput than DWCA and EDWCA protocols.

Figure 13 displays the effect of mobility on the packet drop rate. With increasing in the mobility of nodes, link failures occur causing more packets to be dropped on way. This decrease is less in RE²WCA algorithm because of the use of mobility thresholds as well as group mobility.

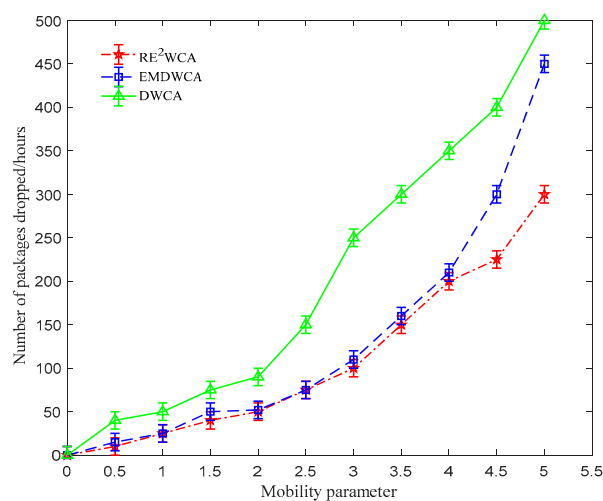


Figure 13. Number of packages dropped varies with Mobility parameter.

Figure 14 shows the energy consumption of the whole network. We can conclude that DWCA consumes energy fastest, EMDWCA second and RE²WCA slowest. The energy consumption is vital in sensor network, however, DWCA fails to control the energy consumption reasonably, therefore when the round number is nearly 700 rounds, the network is paralyzed. In EMDWCA, the lifetime of the network is relatively longer because the CH election considers the residual energy. Unfortunately, since the increased routing overhead and constructed redundancy link, the energy still consumes fast, therefore when the round number is nearly 1100 rounds, the network is paralyzed. However, the proposed algorithm can reach 1220 rounds, because we not only consider the residual energy but also exclude the fault nodes so that network can proceed the topology maintenance which can save amount of energy caused by the times of failed routing. The energy consumption of network which is conducted in RE²WCA protocol is 50% less than DWCA and 15% for EMDWCA respectively. In conclusion, we discuss the cluster application from different aspects which are throughput and network energy consumption. The throughput actually reflects the lifespan of the network and data transmission ability. Network energy consumption however, demonstrates the energy-efficiency, which indirectly implies the lifetime of network. From the simulations results, we can conclude that our protocol, RE²WCA, is superior to original two protocols, EMDWCA and DWCA respectively.

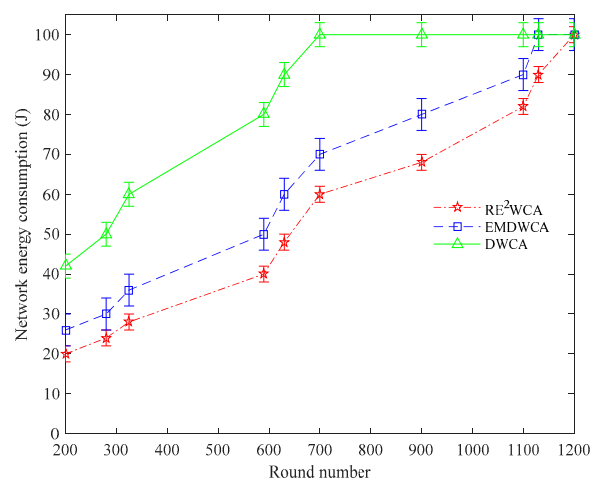


Figure 14. Network energy consumption with the network round number increases.

6. Conclusions

A robust and energy-efficient weighted clustering algorithm on mobile Ad hoc sensor networks is put forward in this paper. Taking group mobility and residual energy into account, the algorithm focuses on the weight of nodes. In addition, the iteration times are decreased dramatically in the protocol, which can guarantee the energy-efficiency in the mobile networks. This protocol renders the nodes consuming energy more homogeneously and prolongs the network lifetime. In addition, we propose a periodic fault detection algorithm to enhance the fault tolerance and robustness of network. The topology maintenance is executed when fault nodes are detected or the average energy of all nodes is below a predetermined threshold value. Thus, the fault tolerance ability is enhanced and the throughput of network is more stable. The results of simulation also illustrate that the performance of throughput, lifetime and robustness of the network is much better than the other protocols.

Our future work will be focusing on the robustness and trust estimation of the network. Currently, the security of the network is a hotspot. However, most of security authentications are trust value estimation through a frequent data aggregation. But clearly the energy consumption of data transmission is greater than the data computing. Therefore, we will consider a kind of trust estimation through minimum data transmission. We believe this field and direction will be made a big progress.

Author Contributions: Conceptualization, H.Q. and F.L.; Methodology, H.Q.; Software, T.X.; Formal Analysis, T.X.; Investigation, F.L.; Resources, S.J.; Data Curation, S.J.; Writing-Original Draft Preparation, T.X.; Writing-Review & Editing, F.L.; Project Administration, H.Q.; Funding Acquisition, T.X.

Funding: This research has been sponsored by Natural Science Foundation of China (project number: 61572528).

Acknowledgments: The work is also supported by Central South University of College students' free exploration project (project number: 201810533168) and Central South University of College students' research studying and innovative experiment project (project number: YC20180624).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kafle, V.P.; Fukushima, Y.; Harai, H. Design and implementation of dynamic mobile sensor network platform. *IEEE Commun. Mag.* **2015**, *53*, 48–57. [\[CrossRef\]](#)
2. Qi, H.; Xiao, T.; Liu, A.; Jiang, S. Toward Energy-Efficient and Robust Clustering Algorithm on Mobile Ad Hoc Sensor Networks. In Proceedings of the International Conference on COCOA 2017, Shanghai, China, 16–18 December 2017; pp. 182–195.
3. Zhang, W.; Han, G.; Feng, Y.; Lloret, J.; Shu, L. A survivability clustering algorithm for ad hoc network based on a small-world model. *Wirel. Person. Commun.* **2015**, *84*, 1835–1854. [\[CrossRef\]](#)

4. Ari, A.A.A.; Damakoa, I.; Gueroui, A.; Titouna, C.; Labraoui, N.; Kaladzavi, G.; Yenke, B.O. Bacterial foraging optimization scheme for mobile sensing in wireless sensor networks. *Int. J. Wirel. Inf. Netw.* **2017**, *24*, 254–267. [[CrossRef](#)]
5. Fadel, E.; Gungor, V.; Nassef, L.; Akkari, N.; Maik, M.A.; Almasri, S.; Akyildiz, I.F. A survey on wireless sensor networks for smart grid. *Comput. Commun.* **2015**, *71*, 22–33. [[CrossRef](#)]
6. Capella, J.V.; Campelo, J.C.; Bonastre, A.; Ors, R. A reference model for monitoring iot wsn-based applications. *Sensors* **2016**, *16*, 1816. [[CrossRef](#)] [[PubMed](#)]
7. Meng, T.; Li, X.; Zhang, S.; Zhao, Y. A hybrid secure scheme for wireless sensor networks against timing attacks using continuous-time markov chain and queueing model. *Sensors* **2016**, *16*, 1606. [[CrossRef](#)] [[PubMed](#)]
8. Asad, M.; Nianmin, Y.; Aslam, M. Spiral Mobility Based on Optimized Clustering for Optimal Data Extraction in WSNs. *Technologies* **2018**, *6*, 35. [[CrossRef](#)]
9. Taheri, H.; Neamatollahi, P.; Younis, O.M.; Naghibzadeh, S.; Yaghmaee, M.H. An energy-Aware distributed clustering protocol in wireless sensor networks using fuzzy logic. *Ad Hoc Netw.* **2012**, *10*, 1469–1481. [[CrossRef](#)]
10. Heinzelman, W.; Chandrakasan, A.; Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel.* **2002**, *1*, 660–670. [[CrossRef](#)]
11. Abboud, K.; Zhuang, W. Stochastic modeling of single-hop cluster stability in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 226–240. [[CrossRef](#)]
12. Zhang, D.; Chen, Z.; Zhou, H.; Chen, L.; Shen, X.S. Energy-balanced cooperative transmission based on relay selection and power control in energy harvesting wireless sensor network. *Comput. Netw.* **2016**, *104*, 189–197. [[CrossRef](#)]
13. Chatterjee, M.; Das, S.K.; Turgut, D. Wca: A weighted clustering algorithm for mobile ad hoc networks. *Clust. Comput.* **2002**, *5*, 193–204. [[CrossRef](#)]
14. Zhang, Y.; Ng, J.M.; Low, C.P. A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks. *Comput. Commun.* **2009**, *32*, 189–202. [[CrossRef](#)]
15. Misra, S.; Singh, S.; Khatua, M.; Obaidat, M.S. Extracting mobility pattern from target trajectory in wireless sensor networks. *Int. J. Commun. Syst.* **2015**, *28*, 213–230. [[CrossRef](#)]
16. Jain, D.; Payal, A.; Singh, U. Sensor nodes based group mobility model (sn-gm) for manet. *Int. J. Sci. Eng. Res.* **2013**, *4*, 823–830.
17. Gherbi, C.; Aliouat, Z.; Benmohammed, M. An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks. *Energy* **2016**, *114*, 647–662. [[CrossRef](#)]
18. Bentaleb, A.; Boubetra, A.; Harous, S. Survey of clustering schemes in mobile ad hoc networks. *Commun. Netw.* **2013**, *5*, 8. [[CrossRef](#)]
19. Dhamodharavadhani, S. A survey on clustering based routing protocols in mobile ad hoc networks. In Proceedings of the 2015 International Conference on Soft-Computing and Networks Security (ICSNS), Coimbatore, India, 25–27 February 2015; pp. 1–6.
20. Gomathi, K.; Parvathavarthini, B. An enhanced distributed weighted clustering routing protocol for key management. *Indian J. Sci. Technol.* **2015**, *8*, 342. [[CrossRef](#)]
21. Bentaleb, A.; Harous, S.; Boubetra, A. A weight based clustering scheme for mobile ad hoc networks. In Proceedings of International Conference on Advances in Mobile Computing & Multimedia; ACM: Vienna, Austria, 2013; pp. 161–166.
22. Bhatti, D.M.S.; Saeed, N.; Nam, H. Fuzzy C-Means Clustering and Energy Efficient Cluster Head Selection for Cooperative Sensor Network. *Sensors* **2016**, *16*, 1459. [[CrossRef](#)] [[PubMed](#)]
23. Ma, S.Q.; Guo, Y.C.; Lei, M.; Yang, Y.; Cheng, M.Z. A cluster head selection framework in wireless sensor networks considering trust and residual energy. *Ad Hoc Sens. Wirel. Netw.* **2015**, *25*, 147–164.
24. Guo, J.; Liu, X.; Jiang, C.; Cao, J.; Ren, Y. Distributed fault-tolerant topology control in cooperative wireless ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 2699–2710. [[CrossRef](#)]
25. Torkestani, J.A. Mobility-based backbone formation in wireless mobile ad-hoc networks. *Wirel. Person. Commun.* **2013**, *71*, 2563–2586. [[CrossRef](#)]
26. He, J.S.; Ji, S.; Pan, Y.; Cai, Z. Approximation algorithms for load-balanced virtual backbone construction in wireless sensor networks. *Theor. Comput. Sci.* **2013**, *507*, 2–16. [[CrossRef](#)]

27. Younis, M.; Senturk, I.F.; Akkaya, K.; Lee, S.; Senel, F. Topology management techniques for tolerating node failures in wireless sensor networks: A survey. *Comput. Netw.* **2014**, *58*, 254–283. [\[CrossRef\]](#)
28. Jiang, J.; Han, G.; Wang, F.; Shu, L.; Guizani, M. An efficient distributed trust model for wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1228–1237. [\[CrossRef\]](#)
29. Panda, M.; Khilar, P.M. Distributed byzantine fault detection technique in wireless sensor networks based on hypothesis testing. *Comput. Electr. Eng.* **2015**, *48*, 270–285. [\[CrossRef\]](#)
30. Singh, K.; Sharma, T.P. Fdr: Fault detection and recovery scheme for wireless sensor networks using virtual grid. *Int. J. Parallel Emerg. Distrib. Syst.* **2017**, *32*, 617–631. [\[CrossRef\]](#)
31. Rahat, A.A.M.; Everson, R.M.; Fieldsend, J.E. Evolutionary multi-path routing for network lifetime and robustness in wireless sensor networks. *Ad Hoc Netw.* **2016**, *52*, 130–145. [\[CrossRef\]](#)
32. Zeng, Y.; Xu, L.; Chen, Z. Fault-Tolerant Algorithms for Connectivity Restoration in Wireless Sensor Networks. *Sensors* **2016**, *16*, 3. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Liu, Q.; Yang, Y.; Xue-song, Q. A metric-correlation-based fault detection approach using clustering analysis in wireless sensor networks. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 526–531.
34. Younis, O.; Fahmy, S. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans. Mob. Comput.* **2004**, *3*, 366–379. [\[CrossRef\]](#)
35. Lin, H.; Bai, D.; Gao, D.; Liu, Y. Maximum Data Collection Rate Routing Protocol Based on Topology Control for Rechargeable Wireless Sensor Networks. *Sensors* **2016**, *16*, 1201. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Zhang, X.M.; Zhang, Y.; Yan, F.; Vasilakos, A.V. Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Trans. Mob. Comput.* **2015**, *14*, 742–754. [\[CrossRef\]](#)
37. Shi, B.; Wei, W.; Wang, Y.; Shu, W. A Novel Energy Efficient Topology Control Scheme Based on a Coverage-Preserving and Sleep Scheduling Model for Sensor Networks. *Sensors* **2016**, *16*, 1702. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Bagci, H.; Korpeoglu, I.; Yaz, A. A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 914–923. [\[CrossRef\]](#)
39. Deniz, F.; Bagci, H.; Korpeoglu, I.; Yazıcı, A. An adaptive, energy-aware and distributed fault-tolerant topology-control algorithm for heterogeneous wireless sensor networks. *Ad Hoc Netw.* **2016**, *44*, 104–117. [\[CrossRef\]](#)
40. Gui, J.; Zhou, K.; Xiong, N. A Cluster-Based Dual-Adaptive Topology Control Approach in Wireless Sensor Networks. *Sensors* **2016**, *16*, 1576. [\[CrossRef\]](#) [\[PubMed\]](#)
41. Das, T.; Roy, S. Employing cooperative group mobility model for mobile target tracking in mwsn. In Proceedings of the 2015 Applications and Innovations in Mobile Computing (AIMoC), Kolkata, India, 12–14 February 2015; pp. 55–61.
42. Das, T.; Roy, S. Energy efficient and event driven mobility model in mobile wsn. In Proceedings of the 2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Kolkata, India, 15–18 December 2015; pp. 1–6.
43. Camp, T.; Boleng, J.; Davies, V. A survey of mobility models for ad hoc network research. *Wirel. Commun. Mob. Comput.* **2002**, *2*, 483–502. [\[CrossRef\]](#)

