Fuzzy Controller Based Unequal Clustering Algorithm with Fault Tolerance

https://doi.org/10.3991/ijoe.v13i12.7263

Chuhang Wang
Changchun Normal University, Changchun, China

Huangshui Hu(✉)
Changchun University of Technology, Changchun, China
huhuangshui@ccut.edu.cn

Abstract—In order to solve the hot spots problem in equal wireless sensor networks, a distributed Fuzzy controller based Unequal Clustering algorithm with Fault Tolerance (FUCFT) is proposed. Fuzzy controller in FUCFT uses the residual energy, node centrality, node degree and distance to base station as input variables and outputs chance and size based on the IF-THEN rules, so as to make the optimal nodes be elected as cluster heads with appropriate cluster size. Moreover, TDMA scheme is adopted to transmit data among the clusters and tolerate the faults for the cluster heads and member nodes. Reclustering on demand is used to reduce the energy dissipation of the consecutive clustering in rounds. The simulation results indicate that FUCFT ensures load balance among clusters, and reduces the energy consumption as well as improves the network lifetime compared with its counterparts.

Keywords—wireless sensor networks, unequal clustering, fuzzy logic, fault tolerance, load balance

1 Introduction

Recently, Wireless Sensor Networks (WSNs) which consist of numerous nodes with the integration of sensing and communication abilities have been widely investigated and deployed in a variety of environments to support healthcare, emergency response, environmental monitoring, and space exploration et al, [1-2]. The impact from environments together with energy constraints of the nodes makes high energy efficiency as well as long lifetime a challenging task. Clustering is one significant way to achieve this task by organizing the sensor nodes with near locations into disjoint groups called clusters [3-4], where each cluster has a cluster head (CH) with the purpose of gathering data from the cluster members (CMs) , aggregating the gathered data and sending it to the base station (BS). At the same time, the CMs gather data from the environment and transmit it to their CHs [5-8]. By this way, the amounts of transferred data and inference as well as overheads in communication are significantly reduced. Consequently, the network scalability, energy efficiency, real-time performance and lifetime are improved greatly [2,3].

Usually, equal clustering mechanism is used to form the clusters with relatively equal size, which means that each cluster includes nearly equal number of nodes to ensure that CHs are distributed evenly over the network and each cluster has almost the same coverage area [8,9]. However, there is a major problem for this type of clustering that the traffic load is not evenly distributed among all the nodes because the nodes in the vicinity of the BS have to relay more data than farther ones, then the energy of them is dropped in a faster rate, which is typically known as the hot spot [3]. To solve this problem, unequal clustering is provided to balance the load, which makes the closer cluster to the BS be smaller in size[4,10,11]. It is obvious that the smaller the number of cluster members, the less the rate of intra-cluster energy consumption. Thus, such CHs could save more energy for relaying the data received from farther clusters. Moreover, most of the existing methods maintain the created clusters by reclustering in rounds, which increase the energy dissipation of the network due to the overhead of consecutive clustering. So the on demand reclustering methods have provided to deal with this problem [8], which triggers the reclustering process when a certain parameter such as residual energy becomes less than the predefined threshold so as to save the energy consumed by the continuous clustering phases. In this case of reclustering, the nodes cannot collect data. Moreover, the CHs and CMs with failure arise if not predicted. Then the CMs keep sending their data to the failed CHs and undoubtedly the data will be lost. So fault-tolerant clustering approaches have investigated to solve this problem, in these approaches backup CHs (BCHs) [13-15] are used to take over the responsibility once the defined CHs notice their eligibilities of being CHs are at a certain level or disordered.

This paper proposes an unequal clustering algorithm FUCFT with fault tolerance ability based on fuzzy logic principle. FUCFT elects an optimal node as CH based on its residual energy, node centrality, distance to BS, and assigns maximum limit of number of members for the CH, Moreover, fault-tolerance mechanism is adopted to maintain the clusters when their CHs or CMs are at a certain level or disordered due to temporary or permanent faults. Reclustering on demand is performed in order to reduce the energy dissipation due to the overhead of consecutive cluster formation phases. Because of the restriction in number of members and the cluster maintaining mechanism for a CH, even load, energy reduction and fault tolerance are achieved while the hot spot problem is solved in FUCFT.

2 Related Works

A number of clustering algorithms have been proposed for WSNs. Low-Energy Adaptive Clustering Hierarchy (LEACH) [5] is one of the earliest and well-known clustering algorithms, which uses a probabilistic approach for selecting the CHs and assures that all the nodes in the network get selected as CH for at least once in a certain round. Moreover, the CHs newly formed communicate with their CMs by a TDMA mechanism so as to make CMs forward their sensed data in allotted timeslots, then aggregate the collected data and send it to the BS directly. Simplicity, distribution, balanced load, low overhead and configurable number of CHs LEACH is, however, direct communication between CHs and BS makes the farther CHs deplete energy at a faster rate, and thus the network cannot be implemented in large scales. More importantly, electing CHs randomly and not considering their energy lead to distribute CHs unevenly and elect the nodes with low energy as CHs. In order to improve the performance of LEACH, several further studies are presented in [6-8]. By taking the node degree and node centrality into account, EADC-FL [8] overcomes the shortcoming of unevenly distributed cluster head as enjoyed by LEACH. And the residual energy is used as the primary parameter to elect the candidate cluster heads so as to make the sensor nodes with the higher energy have more chance to be elected as the candidate CHs. Subsequently, each node calculates the value of its cost using fuzzy logic with inputs of node degree and node centrality. Thus, the candidate CH whose cost is the least will be elected as the final CH and finally clusters in equal size are constructed to make the energy consumption of cluster members balanced. Moreover, a routing tree is created for inter-cluster communication as in [9] in order to balance the energy consumption among the CHs. However, communication in multi-hop fashion among CHs inevitably makes the CHs in the vicinity of the BS take more data relay tasks and lose more energy compared to the father ones, which causes hot spot problem [3].

Consequently, many unequal clustering algorithms are proposed to obtain even energy consumption and solve the hot spot problem [3,4,10], moreover, the unequal clustering has proved to be better than equal clustering in most forms of deployments[2,3,4]. DSBCA[10] considers the residual energy, connection density and times of being elected as the parameters to elect CHs so as to make the cluster much farther from the base station has larger cluster radius, or otherwise. Thus the balanced clustering structure is built and the network lifetime is enhanced. However, switching or reelecting cluster head only in the same or ‘old’ cluster could make the nodes at the edge of clusters be as CHs resulting in unbalanced energy consumption, even isolate the cluster from the network. So EAUCF [11] works in rounds as LEACH to maintain the unequal clusters. Besides, in EAUCF, the residual energy and distance to the BS are used for the computation of the tentative CHs’ competition radius by a fuzzy logic system which can cope with different uncertainty in the network [12], and the tentative CH with highest residual energy within the cluster radius becomes the final CH. Afterwards, the CMs join the CH nearest to them. But in EAUCF, once much more number of nodes is close to the cluster near the BS, the energy of the CH depletes very quickly since many nodes close to the CH join in the cluster. To overcome this issue, FBUC [13] which is an improvement of EAUCF introduces one more variable node degree in the competitive radius computation where the competition radius determines the size of the cluster. Moreover, the CMs join in the CH based on the distance to the CH and cluster head degree which is the ratio of the number of nodes within its competition range of the total number of nodes using another fuzzy logic system so as to utilize the energy efficiently and to extend the network lifetime. The major drawback in EAUCF and FBUC is the restriction in number of members for a CH by the competitive radius, which leads to uneven energy consumption in the network. Also, these algorithms do not consider the energy consumption due to high intra cluster communication which affects the overall performance [3]. Therefore, to solve these problems, DUCF [3] assigns maximum limit of number of members for a

CH based on its residual energy, node degree and distance to BS by a fuzzy output variable 'size' in order to balance the energy consumption. Moreover, DUCF uses the second fuzzy output variable 'chance' with the same inputs as 'size' for electing the CHs so as to make the node have higher 'chance' value than its neighbors elect itself as CH. Also, the CHs will check its 'size' for acceptance of new members when receiving a joining message. If the number of member nodes is more than its 'size' value, the CH send a message which indicates no space for the new member node. And this new member node will send another joining message for the next nearest CH apart from the previous one and this process continues till it joins a CH node. In the worst case, when a non-CH node cannot join any CH, it gets elected itself as CH. Afterwards, a TDMA mechanism is adopted to communicate among CH and its member nodes. Furthermore, a multi-hop data transmission scheme is used to reduce energy consumption in CHs. Though these related works show improvement than one another, less attention on fault tolerance of the nodes is paid to improve the reliability and extend the lifetime of the network.

Usually, BCHs are used to take over the responsibility once the elected CHs fail or reach its preset energy threshold [14-16]. In [15], an algorithm to enhance the ability of fault-tolerance of CHs is proposed. In the algorithm, the status of CHs is examined in different time slot. If there is no response from the cluster head, a node close to it is selected to be the BCH by a fuzzy logic system, and the BCH will substitute the faulty CH. However, this algorithm focuses on only the CH failure while ignoring CM failure. Moreover, the temporary CH failure is not considered, and only one BCH is considered. However, there is no guarantee that the determined BCHs are always fully functional as the sensor nodes do not consume energy equally. In some situations, the BCHs might consume more energy than the other nodes [16]. To overcome this issue, SCCH [16] constructs a list of BCHs for each CH during the process of CHs selection, which priorities its CMs to be BCHs. To maintain the created clusters, each CH obtains the output of its fuzzy logic system, and if the output is less than the preset threshold, the first node in the backup list replaces the CH, at the same time, the CMs is informed. In order to elect the proper BCH, the list is updated based on receiving the CMs' residual energy. The updated list is sent to the CMs periodically with a data request message to make sure that the most suitable BCH is available. Furthermore, a TDMA mechanism is used to monitor the CHs and CMs, through two frames data transmission between the CH and its CMs, the temporary and permanent failure of CH and CM is decided by checking the number of error mark. If only one error mark is existed, it is temporary failure. Otherwise, it is permanent failure. Then, the CH is replaced by the first node in the backup list or the failed CM is removed. However, the proposed algorithms perform reclustering in rounds which mitigates the energy consumption due to the cluster reconfiguration in each round [8]. Both ECPF [17] and EADC-FL [8] perform reclustering on demand during the steady phase. When a CH finds that its residual energy falls below the preset threshold, it sets a prespecified bit in a data packet which is ready to be sent to the BS in the current TDMA frame. Once the BS receives this data packet, it will inform the network to perform the cluster setup phase at the beginning of coming round. However, the threshold is difficult to be decided for real WSNs.

3 System model

FUCFT can balance the energy consumption among the clusters by forming appropriate sized clusters, and tolerate the failure of the CHs and CMs by TMDA mechanism, moreover, the overhead is significantly reduced by reclustering on-demand, so the overall network lifetime is increased. In this section, the system model is described in detail including the network, energy and fuzzy controller models.

3.1 Network model

In order to simplify the network, the assumptions on the network properties are made as follows:

- N nodes $\mathrm{S}=\{\mathrm{s}_1,\mathrm{s}_2,\ldots,\mathrm{s}_\mathrm{n}\}$ are distributed in a square field, and each node s_i has a unique identity.
- Nodes are stationary after deployment with limited energy.
- Nodes are homogenous in terms of initial energy, processing power, memory, transmission and reception capabilities.
- The distance between nodes can be estimated by using received signal strength indicator (RSSI) which is a common measurement technique to find distance in WSN and has negligible impact on the system.
- During the very first time of deployment, distance to BS, distance to neighbors and number of neighbors will be computed at each node by Hello message interactions.

3.2 Energy model

The energy dissipated by transmitting l -bit message to the distance d is given by:

$$E_{tx}=\begin{cases} l*E_{elec}+\varepsilon_{fs}*d^2, & if\ d<d_0 \\ l*E_{elec}+\varepsilon_{mp}*d^4, & if\ d\geq d_0 \end{cases} \tag{1}$$

where E_{elec} is transmitter energy to run the transmitter or receiver circuitry and ε_{fs} 、 ε_{mp} are energy dissipation values to run the amplifier for close and far distances with the threshold $d_0=\sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$ respectively. Energy consumed in receiving l -bit message is calculated as follows:

$$E_{rx}=l*E_{elec} \tag{2}$$

Moreover, energy consumption due to data aggregation with l -bit is represented in Eq. (3).

$$E_{DA}=l*E_{pDb} \tag{3}$$

where E_{pDb} is energy consumption for single bit data aggregation.

3.3 Fuzzy controller model

Fuzzy logic is a efficient method used to solve various problems in WSNs with lots of uncertainties, which is based on human decision-making behavior and experience [3,12]. Generally, fuzzy clustering algorithms merge different clustering parameters for CH election. As shown in Fig.1, a Mamdani fuzzy controller [3,8,12] is used, which consists four elements: fuzzifier, inference engine, rule base and defuzzifier. Fuzzifier converts the crisp input data into appropriate fuzzy linguistic variable, rule base contains a set of fuzzy rules describing the dynamic behavior of the controller, and inference engine is used to form inferences and make decisions based on the rules, defuzzifier converts the fuzzy output of inference engine into crisp values.

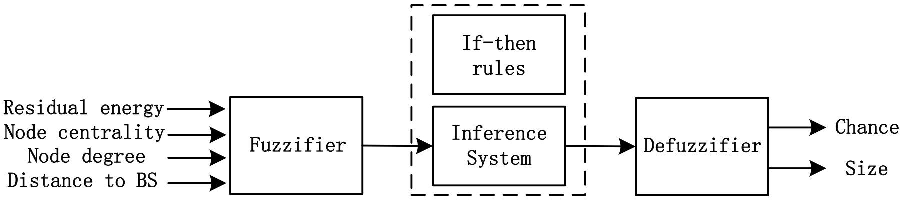


Fig. 1. Fuzzy controller of FUCFT

The input parameters of the fuzzy controller are ‘residual energy’, ‘node centrality’, ‘node degree’ and ‘distance to BS’.

- **Residual energy:** the remaining energy of a node, which is needed to perform the activities such as data collection, aggregation and aggregated data transmission.
- **Node centrality:** the value shows how central the node is among its neighbors proportional to the network dimension, which can be calculated from the Eq. (4).

$$Node\_centrality = \frac{\sqrt{\sum_{j \in N_i} d^2(i,j) / |N_i|}}{S_{area}} \tag{4}$$

Where $|N_i|$ is the number of neighbors of node i, and S_{area} is the size of the sensing field area.

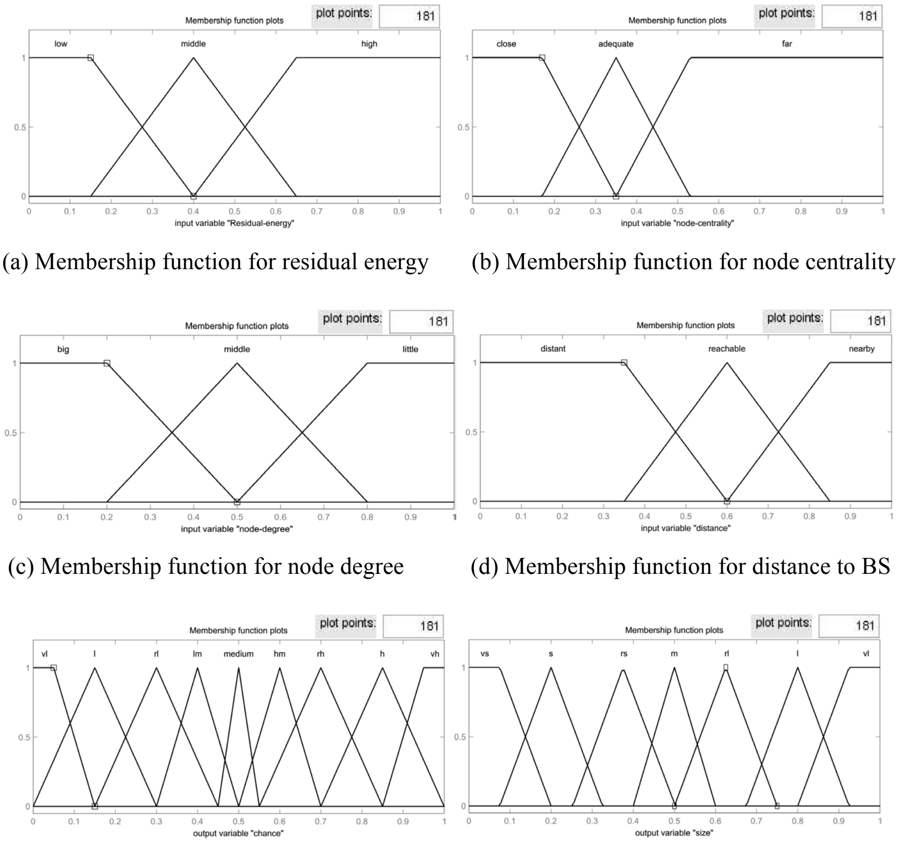
- **Node degree:** the number of neighbors within the communication radius of a node, which could reduce the intra cluster distance for a cluster.
- **Distance to BS:** the distance between node i and the BS, which is used to make the clusters in different sizes so as to avoid the hot spot problem.

Furthermore, FUCFT has two output variables which are also called ‘chance’ and ‘size’ like in DUCF [3].

- **Chance:** the value shows the ability for a node to be selected as CH, which is based on residual energy, node centrality and distance to BS.

- **Size:** the maximum number of CMs can be assigned for a particular CH, which is based on residual energy, node degree and distance to BS.

Next, the membership function for the inputs and outputs are given according to the experimental findings in [3,10] and also from our own experimental results. Low, medium, high is the fuzzy linguistic variable for 'residual energy', close, adequate, far for 'node centrality', big, medium, little for 'node degree', distant, reachable, nearby for 'distance to BS'. Low, high, close, far, big, little, distant and nearby follows trapezoidal membership function, whereas the others have a triangular membership function. In addition, the output 'chance' has nine linguistic variables and they are very low, low, rather low, low medium, medium, high medium, rather high, high, very high. The output 'size' has very small, small, rather small, medium, rather large, large, very large as its seven linguistic variables. The membership function for the inputs and outputs is depicted in Fig.2.



(a) Membership function for residual energy

(b) Membership function for node centrality

(c) Membership function for node degree

(d) Membership function for distance to BS

(e) Membership function for chance

(f) Membership function for size

Fig. 2. Membership functions for inputs and outputs

Based on the membership functions, the crisp input values are fuzzified to appropriate linguistic variable by the inference engine of the fuzzy controller. Then, the fuzzified variables are processed through the if-then rule base which consists of 27 rules, these rules are based on the mentioned above combination of different linguistic variables. The if-then rules are specified in Table 1.

Table 1. Fuzzy if-then rules

No.	**input values**				**output values**	
	energy	***centrality***	***degree***	***distance***	***chance***	***size***
1	low	close	big	nearby	low medium	small
2	low	close	big	reachable	rather low	rather small
3	low	close	big	distant	low	rather small
4	low	adequate	medium	nearby	rather low	small
5	low	adequate	medium	reachable	low	rather small
6	low	adequate	medium	distant	very low	rather small
7	low	far	little	nearby	low	very small
8	low	far	little	reachable	very low	very small
9	low	far	little	distant	very low	small
10	middle	close	big	nearby	high medium	medium
11	middle	close	big	reachable	medium	rather large
12	middle	close	big	distant	low medium	rather large
13	middle	adequate	medium	nearby	medium	rather small
14	middle	adequate	medium	reachable	low medium	medium
15	middle	adequate	medium	distant	rather low	medium
16	middle	far	little	nearby	low medium	small
17	middle	far	little	reachable	rather low	rather small
18	middle	far	little	distant	low	rather small
19	high	close	big	nearby	very high	very large
20	high	close	big	reachable	high	very large
21	high	close	big	distant	rather high	large
22	high	adequate	medium	nearby	high	rather large
23	high	adequate	medium	reachable	rather high	rather large
24	high	adequate	medium	distant	high medium	large
25	high	far	little	nearby	rather high	medium
26	high	far	little	reachable	high medium	medium
27	high	far	little	distant	medium	rather large

The output given by the inference engine is also a fuzzy linguistic variable, center of area method like in [3,8] is used to defuzzify the outputs to crisp values ‘chance’ and ‘size’.

4 FUCFT

FUCFT adopts distributed fuzzy controller to calculate the ‘chance’ and ‘size’ of each node so as to make the optimal nodes elected as CHs and appropriate cluster size assigned. FUCFT works in two different phases: cluster formation phase and cluster maintenance phase.

4.1 Cluster formation phase

Initially, each node calculates its ‘chance’ and ‘size’ by the fuzzy controller. Then each node broadcasts a beacon message *Msg_Candidate* to its neighbors, and the *Msg_Candidate* message contains the node ID and ‘chance’ value. The node with higher ‘chance’ value than its neighbors elects itself as CH and sends an *Msg_Head* message within its communication radius. At the same time, each CH lists its neighbors based on the ‘chance’ value from high to low. Of course, a node may receive more than one *Msg_Head* message. In such case, it will choose to join the nearby CH by sending *Msg_Join* message. Once receiving the *Msg_Join* message, the CH checks its ‘size’ for acceptance of new members. If the current number of member nodes is less than ‘size’, a new member is allowed to join with it by sending back an *Msg_Success* message. Otherwise it sends back an *Msg_Failure* message which indicates no space for the new member node. When a node receives an *Msg_Failure* message, it sends again an *Msg_Join* message for the next nearest CH apart from the previous one and this process continues till it join a CH node. In the worst case, if a node can’t join any CH within its communication radius, it will elect itself as CH. When the cluster is finished, the CH updates its list by removing the non-CMs and broadcasts *Msg_list* message to its member nodes. The pseudo code of cluster formation is illustrated in Fig. 3.

Cluster formation

```
$n$ =number of nodes, $m$ = number of CHs, $ListH$ = list of CHs, $ListM$ =list of Members;
$n = N$ , $m$ =0, $S_i$=the ith node in the network, $Si.state = iniState$;
for i=1 to n do
    $ListH_i = \varnothing$, $ListM_i = \varnothing$;
    $S_i.Er$ =ResidualEnergy, $S_i.Dn$ =NodeDegree, $S_i.Cn$ =NodeCentrality, $S_i.BSn$ =DistanceToBS;
    $Chance, Size = FuzzyCtr(s_i.Er, s_i.Dn, S_i.Cn, s_i.BSn)$;
    sends and receives Msg_Candidate message to and from neighbors
    $Msg\_Candidate = [MSG\_TYPE1, ID_i, Chance_i]$;
    lists neighbors based on the ‘chance’ value according to the received messages;
    $ListBackup_i = \{..., ID_x, ID_y, ...\}$ ;
end for
```

```
for i=1 to n do
  if ( $s_i.Chance > s_j.Chance, \forall s_j \in N_i$ )
      sends Msg_Head message $Msg\_Head = [MSG\_TYPE2, ID_j]$;
      $s_i.state = chState$; $m$ +=1;
  else if ( receives a Msg_Head message from node $s_j$ )
  Adds an item to $ListH_j$={ ..., $s_j$ } ordered by the distance between $s_j$ and $s_i$ ;
      sends Msg_Join message to the nearest $CH_x$
      $Msg\_Join = [MSG\_TYPE3, ID_i, CH\_ID_x]$;
      updates $ListH_j$={ ..., $s_j$ } by removing $s_x$;
  else if (receives a Msg_Join message from Sj)
      if ( $|ListM_i| < s_i.size$ )
  sends a Msg_Sucess message to $s_j$ $Msg\_Success = [MSG\_TYPE4, ID_i, ID_j]$;
        adds an item to ListMi $ListM_i$={..., $S_j$};
      else
        sends a Msg_Failure to $s_j$ $Msg\_Failure = [MSG\_TYPE5, ID_i, ID_j]$;
      end if;
  else if (receives Msg_Sucess message from $S_j$ )
      $s_i.state = cmState$; $s_i.CH = s_j$; $ListH_i = \varnothing$;
  else if (receives Msg_failure from $S_j$ )
      if ( $ListH_i \neq \varnothing$ )
        $S_x$ = the first item in $ListH_i$, removes $S_x$ from $ListH_i$;
        sends Msg_Join message to $S_x$;
      else
        announce itself as CH;
        $s_i.state = chState$;
      end if;
  end if;
end for;
for i=1 to m do
  updates $ListBackup_i$ by removing the items which are not existing in $ListM_i$
;
   broadcasts $ListBackup_i$ to its CMs;
end for;
```

Fig. 3. Pseudo code of cluster formation

4.2 Cluster maintenance phase

After the clusters are created, each CH allocates a TDMA schedule for its CMs. Each CM is awake only during the assigned timeslot to transmit its sensed data. The CH aggregates the received redundant data packets into a single packet and sends it to BS in multi-hop fashion. At this stage, CHs and CMs inevitably face to energy depletion, physical damage and other inference which would make them disordered [16]. Once CHs fail, the entire area of their interest will be unmonitored, so the CMs are required to be noticed quickly to prevent data losing in the cluster. Moreover, disordered CMs also affect the eligibility of the next elected CH, which should be removed by their CH from the list of cluster members $ListM$. Fig.4 shows the process of fault tolerance.

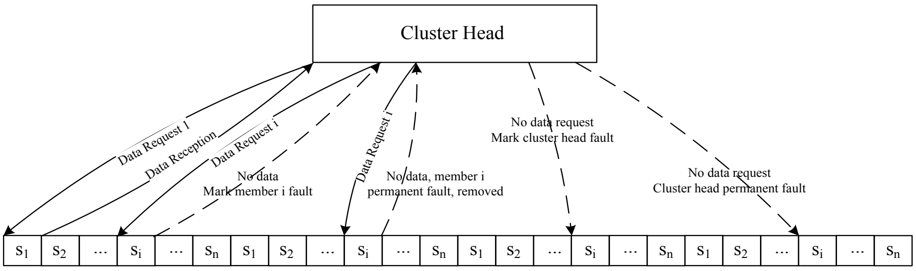


Fig. 4. Data collection and fault tolerance

Seen from Fig.4, the CMs send their sensed data attached with newly chance values upon receiving a data request message at the allotted timeslot. Once the CH can't receive data packet at the end of the frame, it checks whether there is an error mark, and if the error mark has existed, the CH realizes that it is a permanent failure and remove the CM from its list of cluster member. Otherwise, it will mark an error for the CM. Moreover, the error mark is moved by the CH when it receives a data packet with another request so as to tolerate the temporary fault. Similarly, when a CM can't receive data request, it will wait for the next frame to receive the request as it might be a temporary failure. In case of not receiving the request in the second frame, it will be required to replace its CH. To replace the CH, the CM needs to check its received latest and updated BCHs' list which is periodically sent by its CH. Then it sends an Msg_Join message to the first available BCH and wait for the acknowledge message. However, there is a possibility that the BCH might be disordered. In the case of not receiving the acknowledge message from the BCH, the CM sends another Msg_Join message to the second available BCH until it joins to a CH.

Furthermore, reclustering on demand rather than in rounds is used to reduce the overhead of the algorithm and extend the network lifetime. Whenever the 'chance' value of a CH is less than the average 'chance' value of its CMs', the CH will set a prespecified bit in the data packet, which is prepared to be transmitted to the BS. Once the BS receives this data packet, it will inform the network to perform cluster formation. On-demand reclustering results in significant reduction of overhead by the continuous cluster formation phases. Consequently, the energy consumption of network decreases and the network lifetime extends. The pseudo code of on-demand reclustering is given in Fig.5.

Reclustering on-demand

```
$m$ = number of CHs, $ListM$ =list of Members;
for i=1 to m do
    calculates average 'chance' $AverChance = \frac{\sum_{j=1}^{|ListM_i|} s_j.size}{|ListM_i|}$
    if ($s_i.chance < AverChance_i$)
        sets a predefined bit in the data packet to inform BS;
    end if;
end for;
if (BS receives a data packet with predefined bit=True)
    broadcasts Msg_Reclustering message in the network;
end if;
```

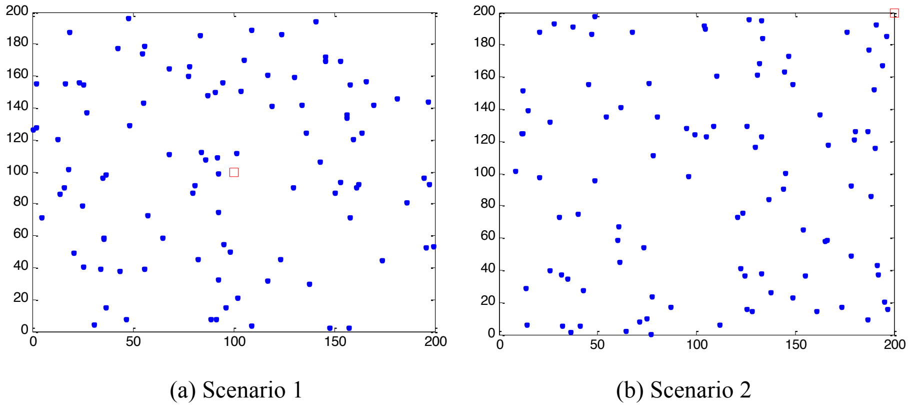
Fig. 5. Pseudo code of on-demand reclustering

5 Simulation results

In this section, simulations are conducted to evaluate the performance of the proposed algorithm FUCFT compared with DUCF [3], LEACH [5] and EADC-FL [8]. In the simulations, 100 nodes are deployed randomly in a square field of area $200m \times 200m$. FUCFT, DUCF, LEACH and EADC-FL have been tested in different scenarios. In Scenario 1: BS node's location is $(50,50)$, which is in the middle of region of interest; Scenario 2: BS node's location is $(200,200)$, which is at the corner of region of interest. The scenarios are shown respectively in Fig.6. The rectangle red box in Fig.6 represents BS and blue spots represents sensor nodes. The initial energy of each node is 1J. Every simulation result is the average of 50 independent experiments, and the parameters of the simulations are listed in Table 2.

Table 2. Simulation parameters

Parameters	Values
l	4000
node initial energy	1 J
E_{elec}	50 nJ ·bit $^{-1}$
ε_{fs}	0 pJ ·bit^{-1}
ε_{mp}	0.0013 pJ ·bit-^{1}
d_0	87 m
E_{pDb}	5nj/bit
data packet size	500 bytes
control packet size	25 bytes



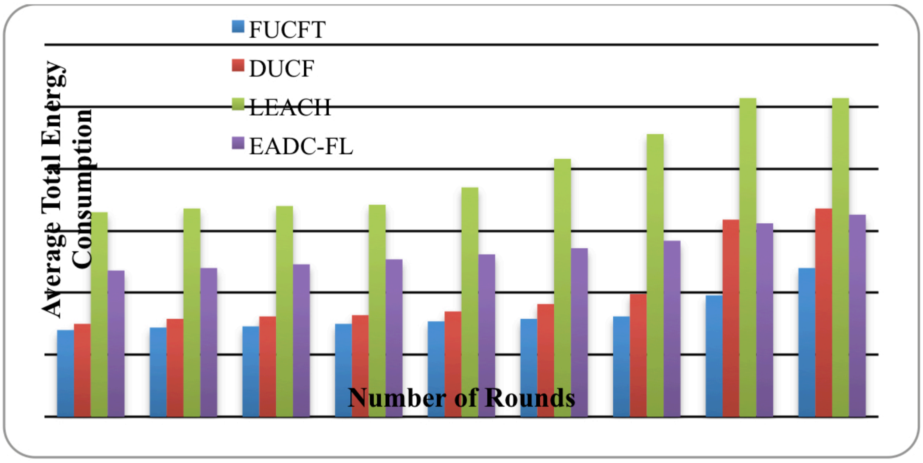
(a) Scenario 1

(b) Scenario 2

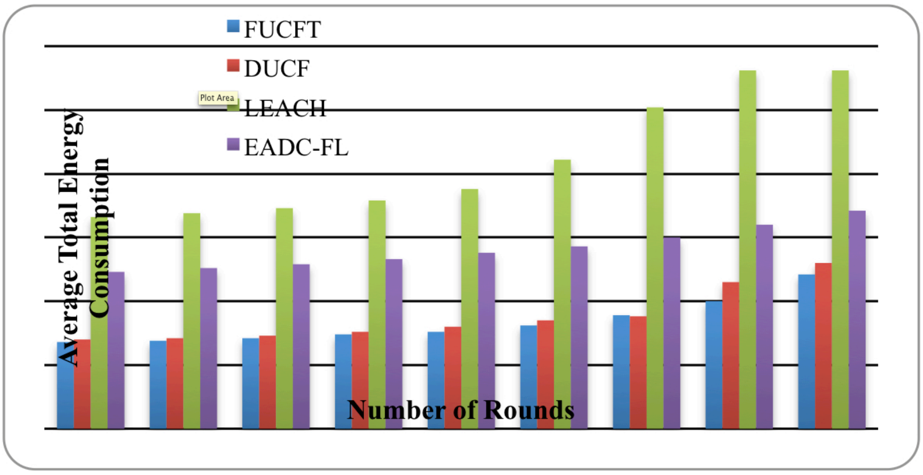
Fig. 6. Network topology in two scenarios

Firstly, the average energy consumption is measured by using the four algorithms in two *Scenarios 1, 2* respectively, and the results are depicted in Fig.7. The results show that LEACH consumes more energy than the other algorithms because its CHs are elected randomly and communicate with BS directly. EADC-FL comes next, which can obtain a better distribution of CH number than LEACH because it considers the node density, node centrality and node energy in selection of CHs as well as communicates with BS in multi-hop fashion. However, it still lags behind DUCF and FUCFT, because DUCF and FUCFT assigns a proper number of member nodes to a CH based on its capacity so as to form unequal clusters. Moreover, FUCFT considers residual energy and node centrality to obtain the optimal 'size' and 'chance' respectively, which reduces the intra-cluster communication compared to DUCF. Thus, the proposed algorithm FUCFT achieves the best energy efficiency.

Secondly, the number of useful messages over the time is performed to evaluate the communication efficiency. The results are given in Fig.8. LEACH shows worst performance than others because it does not restrict the number of members in a cluster which will reduce the number of data message from individual nodes to BS. The same problem continues in EADC-FL which don't take care about the numbers of nodes in a cluster though it forms adaptive clusters based on node density, node centrality and node energy as factors in determining the CH chance of a node. DUCF gives better results than LEACH and EADC-FL since it restricts the number of members through the second fuzzy output variable 'size'. Furthermore, FUCFT tolerates the CH and CM failure to avoid data packets losing between CMs and CH so as to significantly improve the communication efficiency of the network.

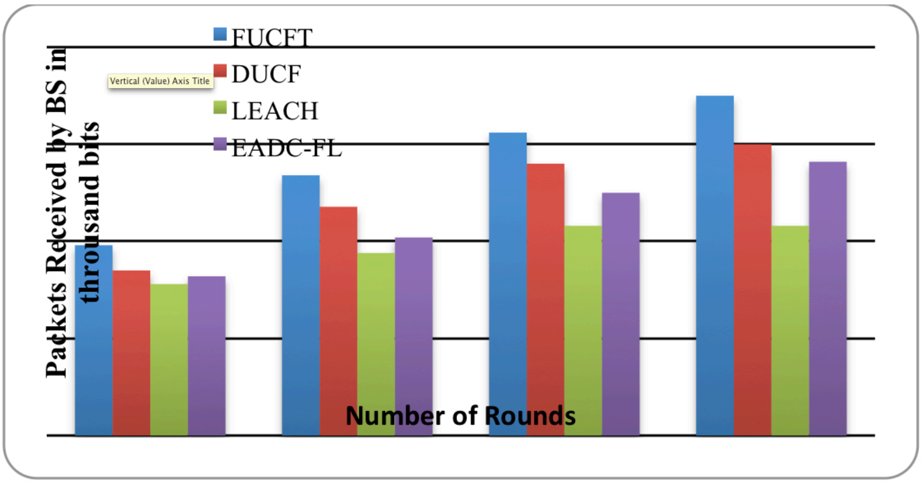


(a) Scenario 1

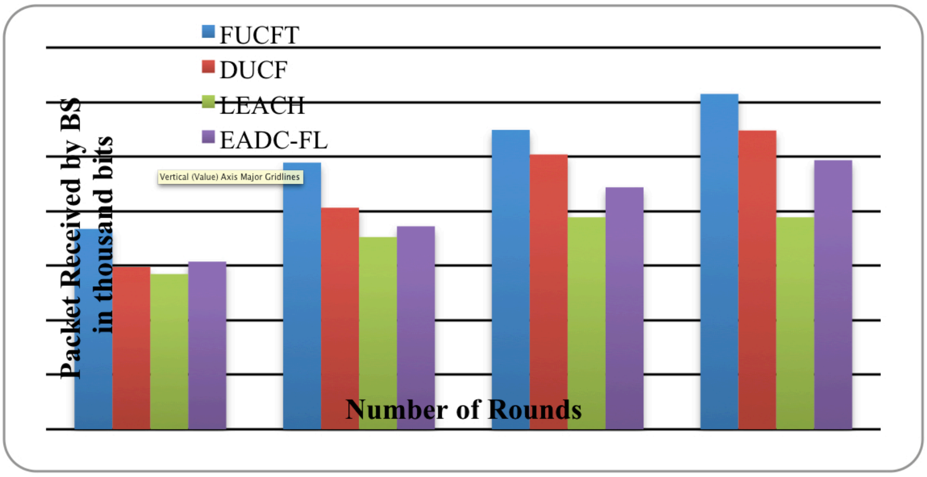


(b) Scenario 2

Fig. 7. Average total energy consumption



(a) Scenario 1



(b) Scenario 2

Fig. 8. Number of packets received by the base station

Finally, the performance comparison in terms of survival nodes which is used to evaluate the network lifetime is presented in Fig.9. Fig.9 shows the average number of survival nodes between *Scenario 1 and Scenario 2.* It's obvious that FUCFT outperforms LEACH, EADC-FL and DUCF. The results show that LEACH has worst performance due to its without considering residual energy in CH selection so as to the sensor nodes with low energy may die prematurely and reduce the network lifetime. DUCF solves the problem of imbalanced energy consumption with unequal clustering to improve the network lifetime. However, reclustering in rounds in DUCF increases the energy dissipation of the network due to the overhead of consecutive clustering formation phases. Therefore, as time goes on, EADC-FL obtains more survival nodes than LEACH and DUCF due to its on demand reclustering scheme. What is more, FUCFT forms unequal clusters with optimal size, especially, replaces the failed CHs and removes the failed CMs. Thus, the network lifetime is extended.

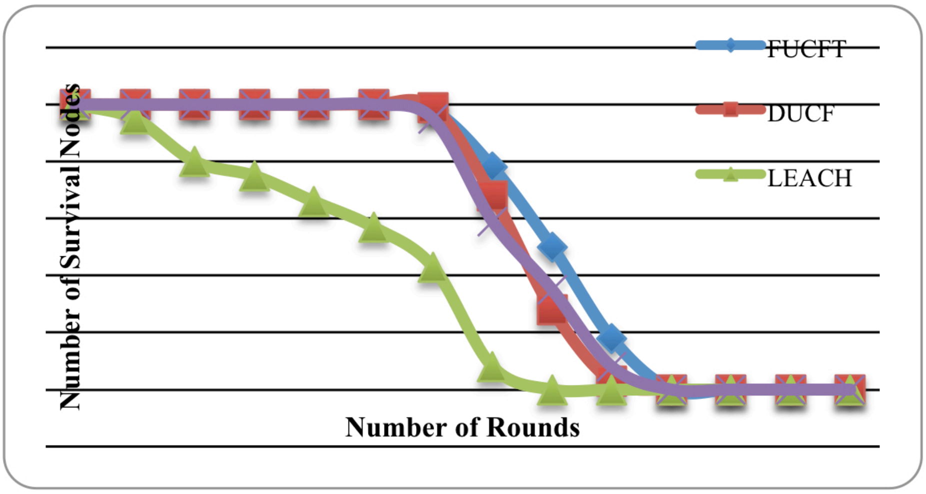


Fig. 9. Comparison of survival nodes

6 Conclusion

A new unequal clustering algorithm with fault tolerance FUCFT is proposed for WSNs using fuzzy controller to decrease the traffic as well as extend the network lifetime. FUCFT considers the residual energy, node centrality and distance to BS as parameters inputted into the fuzzy controller to evaluate the 'chance' of the candidate CHs, a node with the maximum 'chance' will be elected as CH and the rest of the nodes as BCHs. Thus, the CMs can ensure that there is always a BCH for their CHs so as to replace the failed CH with an appropriate BCH using a TDMA scheme. Moreover, FUCFT forms the clusters with restricted numbers by considering residual energy, distance to BS and node degree so as to balance the energy consumption among the nodes. Reclustering on demand is used to reduce the energy dissipation of the consecutive clustering in rounds. The simulation results show that FUCFT outperforms other clustering algorithms, it can improve the energy consumption and lifetime of the network by forming optimal clusters and tolerating the failure of CHs and CMs, which makes it be suitable for many real time applications.

7 References

[1] Borges L. M., Velez F. J., Lebres A. S.(2014). Survey on the characterization and classification of wireless sensor network applications. IEEE Communications Surveys & Tutorials. https://doi.org/10.1109/COMST.2014.2320073

[2] Afsar M M, Tayarani M H(2014). Clustering in sensor networks: a literature survey. Journal of Network and Computer Applications.46: 198-226. https://doi.org/10.1016/j.jnca.2014.09.005
[3] Baranidharan B, Santhi B(2016). DUCF: Distributed load balancing Unequal Clustering in wireless sensor networks using Fuzzy approach. Applied Soft Computing. 40: 495-506. https://doi.org/10.1016/j.asoc.2015.11.044
[4] Sariga A, Pothula S(2017). A survey on unequal clustering protocols in wireless sensor networks networks. Journal of King Saud University-Computer and Information Sciences, 29(4): 428-448. https://doi.org/10.1016/j.jksuci.2017.03.006
[5] Heinzelman W R, Chandrakasan A, Balakrishnam H. Energy-efficient communication protocol for wireless microsensor networks. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. Piscataway, NJ: IEEE, 2000: 3005-3014. https://doi.org/10.1109/TWC.2002.804190
[6] Lee J S, Cheng W L(2012). Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication. IEEE Sensors Journal. 11(9):2891–2897. https://doi.org/10.1109/JSEN.2012.2204737
[7] Attea B A, Khalil E A(2011). A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks. Applied Soft Computing. 12(7):1950–1957. https://doi.org/10.1016/j.asoc.2011.04.007
[8] Nooshin N, Zurina M H, Shamala S, Mohamad A M(2015). An Energy aware distributed clustering algorithm using fuzzy logic for wireless sensor networks with non-uniform node distribution. Wireless Pers Commun. 84:395-419. https://doi.org/10.1007/s11277-015-2614-9
[9] Yu J, Qi Y, Wang G, Gu X(2012). A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution. International Journal of Electronics and Communications. 66(1):54-61. https://doi.org/10.1016/j.aeue.2011.05.002
[10] Ying L, Huang Q, Weiquan L(2013). Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks. IEEE Sensors Journal. 13(5): 1498-1506. https://doi.org/10.1109/JSEN.2012.2227704
[11] Bagci H, Yazici A(2013). An energy aware fuzzy approach to unequal clustering in wireless sensor networks. Applied Soft Computing. 13(4): 1741-1749. https://doi.org/10.1016/j.asoc.2012.12.029
[12] Mirjana M, Valdimir V, Vladimir M(2014). Fuzzy Logic and Wireless Sensor Networks-A Survey. Journal of Intelligent & Fuzzy Systems. 27(2): 877-890. https://doi.org/10.3233/IFS-131046
[13] Logambigai R, Kannan A(2016). Fuzzy logic based unequal clustering for wireless sensor networks. Wireless Network. (22): 945-957. https://doi.org/10.1007/s11276-015-1013-1 https://doi.org/10.1007/s11276-015-1013-1
[14] Zadi D, Ahawajy J, Ghanavati S(2013). A new energy efficient cluster-head and backup selection scheme in WSN. In Proc IEEE 14th Int. Conf. Inf. Reuse Integr(IRI), Aug.2013, 408-415. https://doi.org/10.1109/IRI.2013.6642500
[15] Farnaz P, Mansour E(2016). Fuzzy logic method for enhancement fault-tolerant of cluster head in wireless sensor networks clustering. TEM Journal. 5(3): 268-276. https://doi.org/10.18421/TEM53-03
[16] Davood I, Jemal A, Sara G(2015). An alternative clustering scheme in WSN. IEEE Sensors Journal. 15(7): 4148-4155. https://doi.org/10.1109/JSEN.2015.2411598
[17] Taheri H, Neamatollahi P, Younis O M, etc(2012). An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic. Ad Hoc Networks. 10(7): 1469-1481. https://doi.org/10.1016/j.adhoc.2012.04.004

8 Authors

Wang Chuhang received master degree from Jilin University in 2005. She is currently an associate professor in College of Computer Science and Technology, Changchun Normal University. Her main research interests are wireless sensor networks and real-time embedded systems. (e-mail: wangchuhang@cncnc.edu.cn)

Hu Huangshui received master and doctor degree from Jilin University in 2005 and 2012 respectively. He is currently an associate professor in College of Computer Science and Engineering, Changchun University of Technology. His main research interests are wireless sensor networks, multifunction vehicle bus networks. (e-mail: huhuangshui@ccut.edu.cn

Article submitted 09 June 2017. Published as resubmitted by the authors 13 August 2017.