



Design of Novel High Density, Fault Tolerant Protocol for Cluster Based Routing in Ad-Hoc Networks

Frodouard Minani and Santhi Kumaran^(✉)

African Center of Excellence in Internet of Things (ACEIoT),
College of Science and Technology, University of Rwanda, Kigali, Rwanda
frodominani@gmail.com, santhikr@yahoo.com

Abstract. Clustering in ad-hoc networks is a widely used approach to ease implementation of various problems such as routing and resource management. Ad-hoc wireless networks perform the difficult task of multi-hop communication in environment without a dedicated infrastructure, with mobile nodes and changing network topology. In this article, after determining the cluster members and cluster head, distance and energy for each node are carried out per cluster head and cluster; data gathered from cluster members are relayed to the main server. The results showed that the proposed algorithm is scalable, has less number of dropped packets, uses high energy during the path discovery, and handles fault tolerance problem during the packets transmission. A new algorithm for clustering which merges network nodes to form higher level clusters by increasing their levels, their energy and their density is proposed. Its operations are provided in simulation environment of network simulation tool 2 (ns-2).

Keywords: Cluster density · Fault tolerance performance metrics
Network lifetime · Routing protocol · Throughput

1 Introduction

Ad-hoc network which is a local area network (LAN) that is built spontaneously as devices connect where there are a number of wireless nodes that are self-organized. The movement of the nodes enables them to generate multiple routes. Thus, accurate routes must be determined for those nodes. A new algorithm must therefore be developed to design a routing protocol that adapts to network topology changes. Clustering organizes the ad-hoc networks hierarchically and creates clusters of nodes which are geographically adjacent and each cluster is managed by a cluster head and other nodes may act as cluster gateway or cluster member [1]. Each node in the network or cluster also acts as a router forwarding data packets for other nodes. The cluster density is defined as the expected number of clusters per unit area [2]. Due to node mobility, there may be a chance of node or link failure and node have to consume more energy to transfer the packets from source to destination. Node failure occurs when there is a lack of power and it causes route failure in the network.

We proposed a novel high density, fault tolerant protocol for cluster based routing in ad-hoc networks with varying node's density which handles fault tolerance problem. The algorithm supports addition and deletion of dynamic nodes, ensuring the continuation of service despite the presence of failures [3]. Results were simulated using ns-2. The rest of the paper is organized as follows: Sect. 2 describes the related works. Section 3 talks about the proposed models, clustering algorithm and parameter metrics. Section 4 deals with the analysis of simulation results. Section 5 provides conclusion and future works.

2 Related Works

In Ad-hoc networks, many clustering algorithms have been proposed in the past. An algorithm called Linked Cluster Architecture (LCA) which was initially proposed by Baker and Ephremides [4]. A small variation to LCA was proposed by Ephremides, Wieselthier and Baker in [5] as a lowest ID algorithm. In this algorithm, a node which has the lowest ID among its neighbors is selected as the head node (Cluster-head). It retains its utility as a point of reference for producing reasonably stable cluster control architecture as written by Gerla and Tsai in Xue and Nahrstedt [6] confirmed that devising a fault-tolerant routing algorithm for ad-hoc networks is inherently hard. In [7], the authors designed an efficient algorithm, called the end-to-end fault tolerant routing (E2FT) algorithm, which is capable of significantly lowering the packet overhead, while guaranteeing a certain packet delivery rate. Following the work of Xue and Nahrstedt, Oommen and Misra proposed weak-estimation learning based fault-tolerant routing protocol for Ad-hoc networks.

In [8] Zhou, Xia proposed an algorithm called location based fault tolerant routing algorithm (FTRA). In this algorithm based on geographical location information networks divided in to grid, when faults occur, the proposed algorithm select alternate route from unused at hop in normal routing path, the route selection depends upon location information of its neighbors grids. In [9], Qin and Pang proposed fault-tolerance cluster head based (FTCH) routing protocol reduce misbehaving node in the network. If faulty node occurs, the proposed protocol provide packet delivery fraction guarantee and reduce routing overhead. In [10], they proposed a trusted fault tolerant (TFT) model based on location aided routing protocol along with user recovery features. It covers link failures during packet forwarding and location failures.

3 Proposed Models, Parameter Metrics and Clustering Algorithm

3.1 Proposed Model

As shown on Fig. 1, in this model, around 33 nodes and 3 clusters have been used where each cluster is composed of 11 nodes. Each node in the group communicates regularly with the leader by sending his information: identifier, energy level, neighboring nodes, number of times it stopped, and the directory replicas that exist in the

node. It is assumed that each node has a unique address, called the ID_node, a line refers to one given. For each cluster, a cluster head (CH) is elected which initiates the communication among the others. To elect a cluster head, the cluster energy of a node is measured and calculated. During the packet transmission, when a fault occurs in the system, the algorithm designed searches for a new way so that the communication continues to happen by using the backup nodes. Every node in the network can communicate straightaway with the other nodes located in the transmission range.

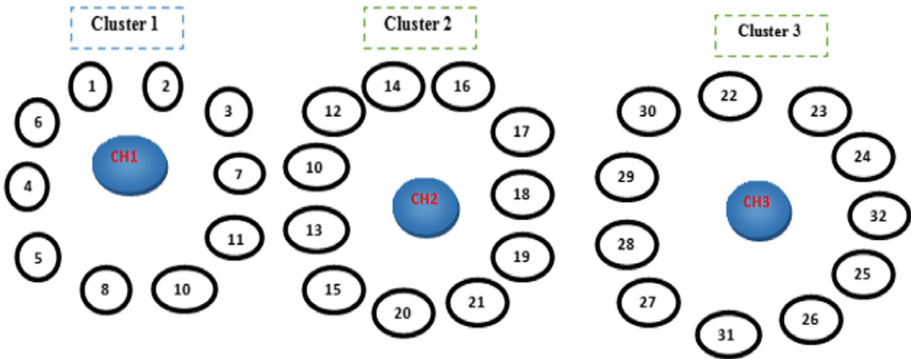


Fig. 1. Proposed model (cluster and cluster head formation)

The failure of some links and nodes is considered as criticism and can divide the network into several partitions. This reduces the data availability and leads to data inconsistency. To increase the data availability, a new protocol or algorithm has been designed to help the communication keeps going on within clusters. The model proposed for tolerating faults in the ad-hoc networks is a model which consists of four sub-services as shown in Fig. 2 below:

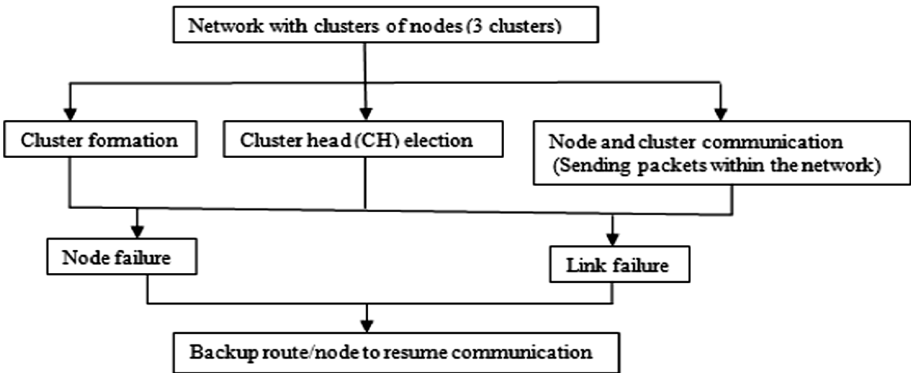


Fig. 2. Fault tolerance model of the algorithm

According to Fig. 2, the fault tolerance algorithm functions in two phases:

- Choosing the backup nodes or routes.
- Resume the communication of nodes within clusters.

3.2 Proposed Flowcharts

As shown in Fig. 3 of this model (flowchart), after forming nodes and clusters, cluster head for each cluster is selected. When faults (node or link failure) occur, the communication is resumed by choosing the backup routes or backup nodes. As indicated on Fig. 4, the average energy of the network is calculated; the nodes having energy more than average energy of network are added to the set of candidate cluster heads. Distance between each candidate cluster head and center is calculated. Candidate cluster head having lowest distance from the center is selected as cluster head for this cluster and remaining nodes from the candidate cluster head set becomes normal nodes.

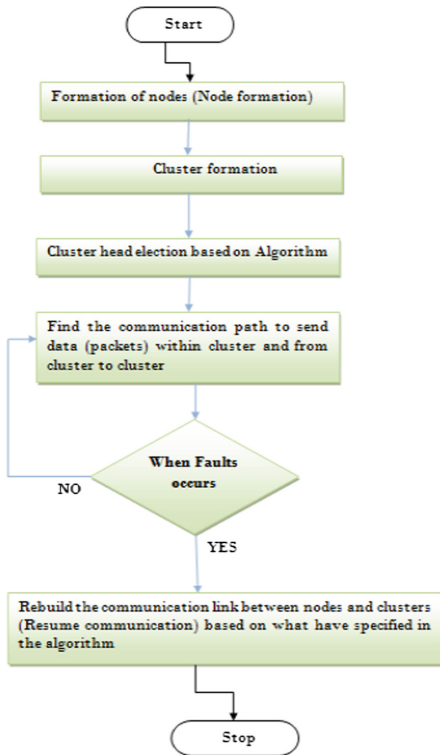


Fig. 3. Proposed flowchart for fault tolerance

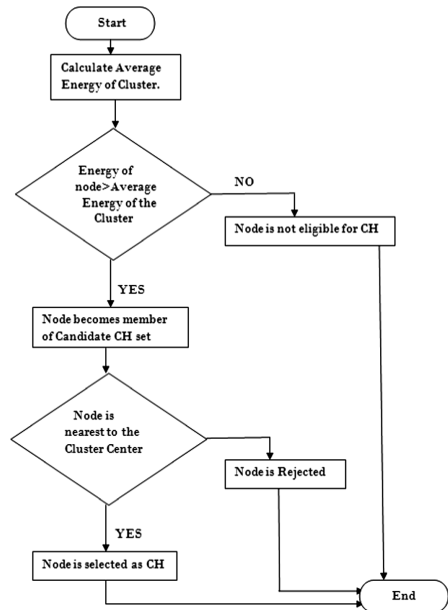


Fig. 4. CH election flowchart

3.3 Parameter Metrics

(i) The Cluster Density

The experiment showed that the cluster density (ρ) i.e., the ratio of the number of cluster head (N) neighborhoods to cluster diameter (d) (if the given node become cluster head) is given by the following formula:

$$\rho = \frac{N}{d}. \quad (1)$$

(ii) Energy Consumed by Node within Cluster

A node with higher energy consumption fails faster. When a cluster head depicts its energy, it fails and also shatters the whole cluster with its links and entails lots of maintenance costs. On the other hand, cluster heads have the highest responsibilities among other nodes. Therefore, they need and consume the highest energy in network and are more likely to drain the energy. As a result, when choosing the clusters, their remaining energy needs to be concerned. Taking this factor into account, the nodes with the highest residual energy and the least consumed energy among their neighbors are selected as the cluster heads. The factor is given as follows:

$$E_n = E_{amp} * k * d. \quad (2)$$

where:

E_n : Energy consumed by node in cluster

E_{amp} : Amplifier coefficient

k : Number of transmitted data bits

d : distance between a sensor node and its respective cluster head or between a CH to another CH nearer to the BS or between CH and BS (Base station).

(iii) Packet Delivery Ratio Calculation

The Cluster head (CH) initiates sending the packets between nodes and between clusters. The Packet delivery ratio is the ratio between the number of packets transmitted by a traffic source and the number of packets received by a traffic sink. It represents the maximum throughput that the network can achieve in ad-hoc networks. A high packet delivery ratio is desired in an ad-hoc networks and it is computed as follows:

$$PDR = \frac{\sum PRCD}{\sum PSCS} * 100. \quad (3)$$

where:

- PRCD: Packets received by cluster destination
- PSCS: Packets sent by cluster source
- PDR: Packets delivery ration.

(iv) End-to-end Delay Computation

Because of packets movement between nodes to nodes and clusters to clusters, delays can happen. End-to-end delay is the average time delay for data packets to reach from the source node to the destination node. It includes processing, queuing and propagation delay of the link. The performance is better when packet end-to-end delay is low. It is computed by the below formula:

$$EED = \frac{1}{N} \sum_{n=1}^N (TS_n - TR_n). \quad (4)$$

where

- TR_n: Time at which data packets n has been sent
- TS_n: Time at which data packets n has been received
- N: Total number of data packets received

(v) Throughput Calculation

For the cluster based routing, the parameter throughput has been taken into consideration which is the total packets successfully delivered to individual destinations over total time.

$$Tp = \frac{\sum P}{T}. \quad (5)$$

where:

- Tp: Throughput
- P: Number of packets successfully received at the cluster destination
- T: Unit time.

3.4 Clustering Algorithms

(i) Algorithm 3.3.1: Cluster Creation Algorithm

```

1: Begin
2: Cluster (N);
3: Set the position of N;
4: Calculate the speed of each node;
5: From N;
6:   If avg speed of each node is low then
7:     Form the cluster size smaller;
8:   Else make the cluster size bigger;
9:   Endif;
10: End;

```

(ii) Algorithm 3.3.2: Density-Based Algorithm

```

Step 1: Begin
Step 2: Create a graph whose nodes are the points to be clustered;
Step 3: For each core-point  $c$ , create an edge from  $c$  to every point
         in the  $\square$ -neighborhood of  $c$ ;
Step 4: Set  $N$  to the nodes of the graph;
         If  $N$  does not contain any core points then
           Terminate;
         Endif;
Step 5: Pick a core point  $c$  in  $N$ ;
Step 6: Let  $X$  be the set of nodes that can be reached from  $c$  by going
         forward;
         Create a cluster containing  $X \square \{C\}$ ;
          $N = N / (X \square \{C\})$ ;
         Continue with step 4;
End;

```

As indicated in the Algorithm 3.3.2, c represents the core point and N represents a set of nodes of a graph. \square represents the edge of points in the cluster

(iii) Algorithm 3.3.3: Fault Tolerant Clustering Algorithm

```

Step 1: Begin
Step 2: Find the eligible node;
         For all the nodes in a cluster;
           If node=already (CH) then
             go to next node;
           Endif;
         Check the node status in memory table;
         If nodestatus=fail(trans_id) then
           go to next node;
         else node can be selected;
         Endif;
         move to step3;
         refresh;
         check for node availability for n sec;
         if NodeAvailability then
           Continue;
         select the next CH;
         else Put in (ListLow);
         Endif;

```

Step 3: Calculate the average threshold (Th_{avg});
 For each node in cluster $C_1 \dots C_n$;
 Calculate threshold;

$$Th(n) = \left[\frac{p}{1 - p[r \bmod (\frac{1}{p})]} \right]$$
 if n belongs to G;
 $Th(n)=0$ if n doesn't belong to G;
 Select nodes with values above the threshold value;

$$Th(avg) = \frac{\sum Th(n)}{n};$$

 If $Th(n) \geq Th(avg)$ then
 put in (ListHigh);
 else put in (ListLow);
 Endif;

Step 4: Calculate the available node energy;
 For all the nodes in the ListHigh;
 Energy=P*T;
 Calculate the available energy of each node;
 $E_{avail} = E / E_{max}$;
 Create a List $E_{availsort}$ in descending order;

Step 5: Calculate minimum distance and throughput;
 For the first node in $E_{availsort}$;
 Calculate minimum distance;

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]};$$

 Calculate throughput;

$$Tp = \frac{Sp}{Tt};$$

 If 2 nodes have same energy then
 Node with $d_{min}=CH$;
 If 2 nodes have same energy and same minimum distance then
 Node with $Tp_{max}=CH$;
 Endif;
 Endif;

Step 6: Check the link/node failure;
 For the transmission;
 If node/link failure then
 Backup node/link;
 else system is down;
 Endif;

End;

As shown in Algorithm 3.3.3, for finding the eligible nodes, we focus on the nodes past record of not being a cluster head already and not failed for the last n operations. The average threshold (Th_{avg}) of the node is the value for which the node will survive in the network. Before electing any node as CH, we take into account of its threshold value where in its formula, p is the prearranged percentage of cluster heads (e.g. $p = 0.1$), r is the current round of iteration and G is the set of nodes that have not been cluster heads in the last $1/p$ rounds and n is a random number between 0 and 1. For energy calculation, P is the power and T is the time. In the formula of distance, x_1 , x_2 , y_1 and y_2 are the coordinates. Throughput (Tp) is calculated as size of the packet (Sp) over the transmission time (Tt) while the available energy of each node equals to the current energy (E) over maximum energy (E_{max}).

4 Analysis of Simulation Results

In order to evaluate the performance of the proposed clustering algorithm, we simulated and showed the results in ns-2 [11]. The nodes are randomly deployed in a circular area. The number of nodes varies from 33 to 35. The initial energy of normal node is 35 J. Table 1 summarizes the simulation parameters and their default values.

Table 1. Simulation parameters

Parameter	Value
Number of nodes N	33–35
Initial energy E0	35 J
Simulation time	2.0 ms
Channel type	Wireless channel
End simulation time	10.0 ms
Scenario size	200 m × 200 m

Figure 5 shows the simulation environment where the nodes are formed and grouped into clusters as indicated. Each node is given a number in each cluster to be differentiated with others. The nodes are numbered from 0 to 34. It also shows the packets dropped from the network. Figure 6 indicates the CH election in each cluster based on the energy consumption of each node, the cluster head (CH) is the node with high energy and it is represented by red color.

Figure 7 indicates the process of fault tolerance (FT) which is the ability of a system to continue its operation after the failure of one of the nodes or link. When faults occur, choose the backup route or nodes to avoid the system failure and maintain the system functionality (availability). At the node 6 (with blue color) in the first cluster, there is a failure but there is a backup route where data which was supposed to pass through that node tried to find another route to pass through. Figure 8 (X-axis: Number of nodes, Y-axis: size of the network area) shows the system performance with the parameter density. The algorithm indicates that when the size of the area increases while the number of nodes remains constant, the cluster density decreases while it increases when the area becomes big at the same time the number of nodes increases.

Figure 9 (X-axis: Node energy, Y-axis: Nodes) indicates the Cluster head selection graph where the node with high energy is selected as CH. The initial node energy was 0.0000 J and there was not CH cluster head elected but as long as the Energy of the node increases (many nodes which have the higher energy), many CHs are elected.

From Fig. 10 (X-axis: Time, Y-axis: Packet delivery ratio), we can see that when time increases, many packets are lost as well as they delay to reach the destination. The red line indicates dropped packets and green line shows the delayed packets. The delayed packets are implicitly increased when the time is moving but this delay starts at certain time after the packets transmission. You can see that the delay starts at 3.000 ms after the starting of the simulation and when the simulation time was 0.000 ms, no packet loss (PDR was 0.000).

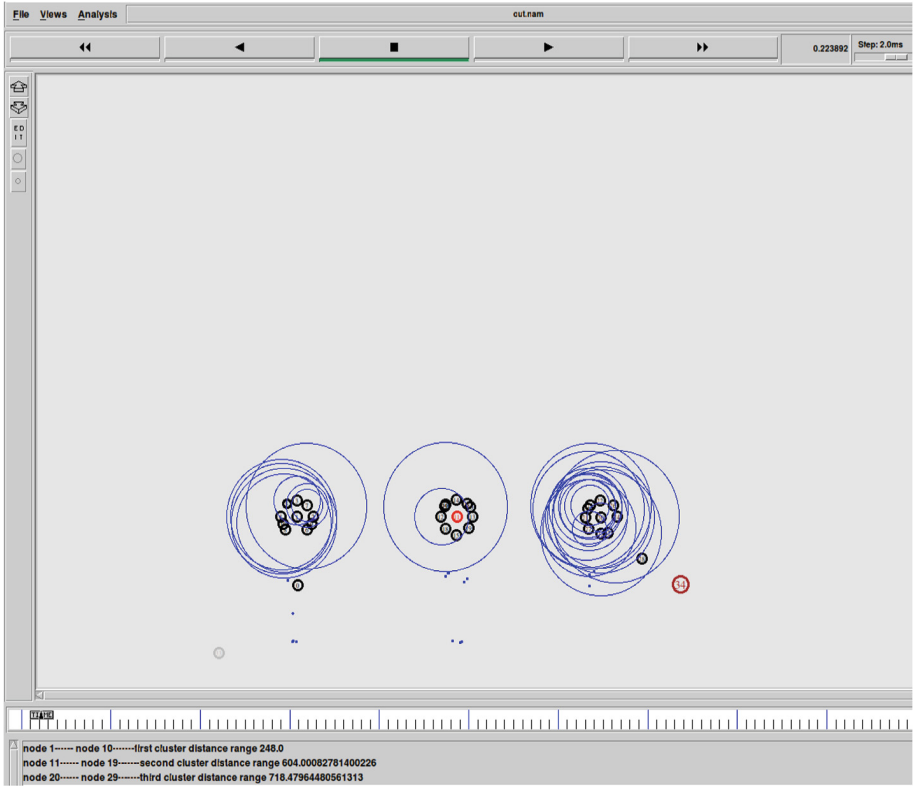


Fig. 5. Nodes with cluster formation and dropped packets

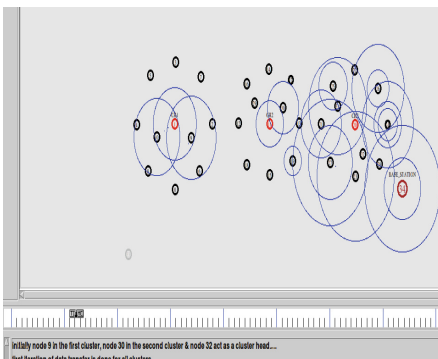


Fig. 6. CH election scenario (Color figure online)

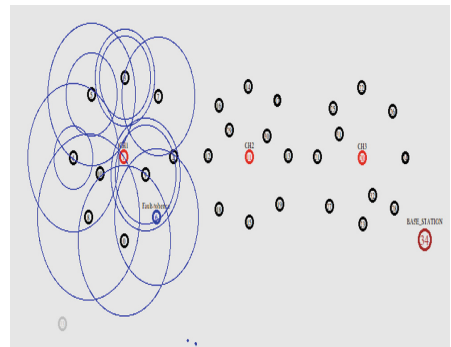


Fig. 7. Node failure in cluster 1(Node 6) (Color figure online)

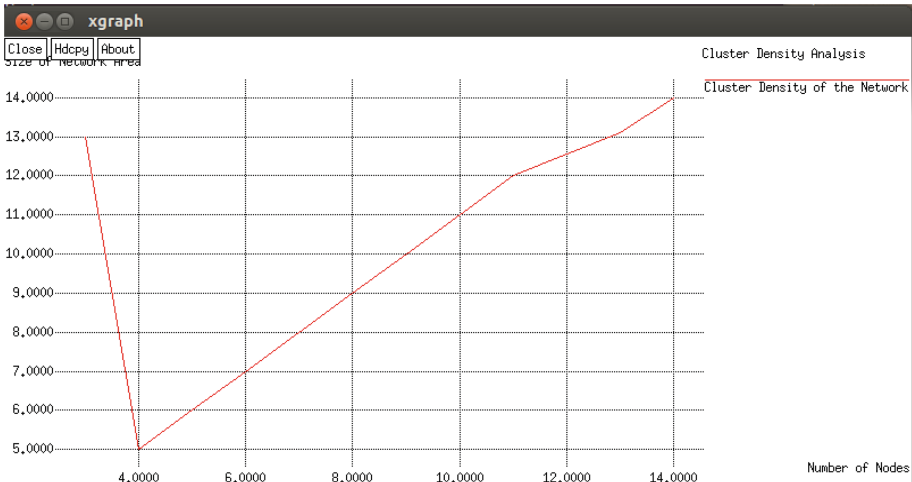


Fig. 8. Cluster density analysis

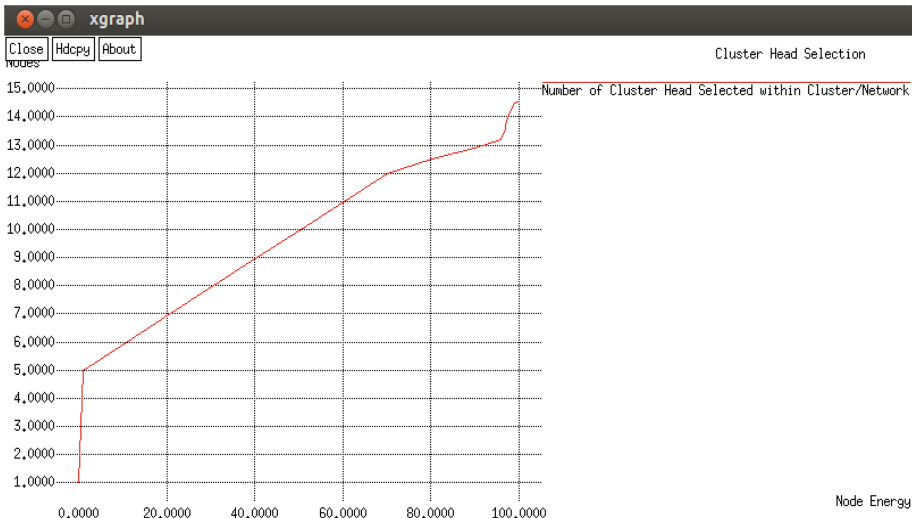


Fig. 9. CH selection graph

Figure 11 (X-axis: Time, Y-axis: Nodes) shows the number of dead (failed) nodes with time. The failed nodes are indicated by green line and the active nodes are represented by red line. It is indicating that as the time increases, active nodes are higher which means that during the packets transmission, small number of nodes failed to transmit as well as the backup nodes or routes is taking place to resume the communication.

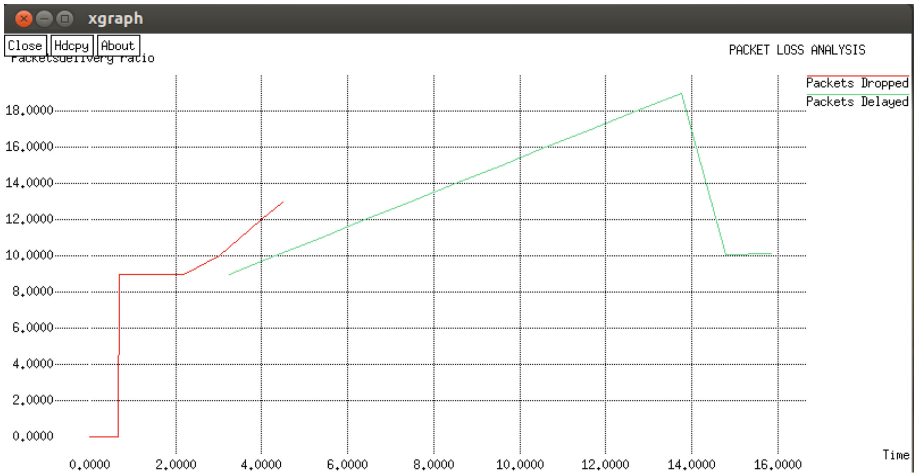


Fig. 10. Packet loss analysis within network vs. time (Color figure online)

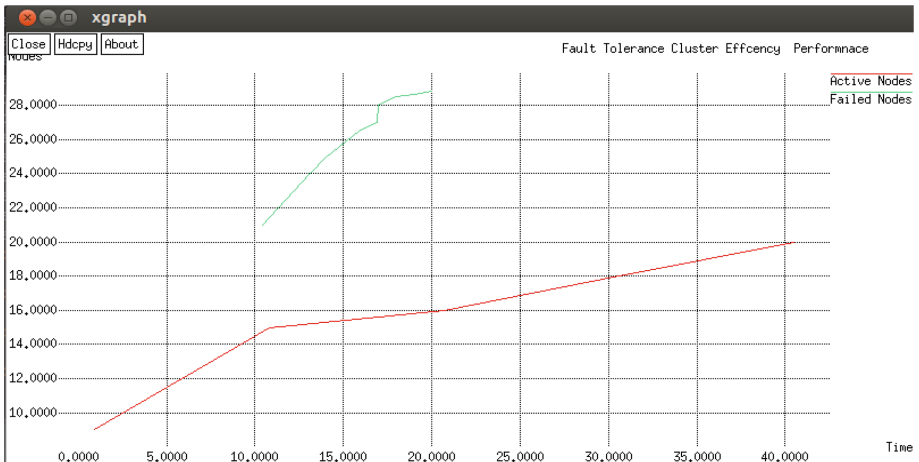


Fig. 11. Fault tolerance cluster efficiency performance (Color figure online)

5 Conclusion and Future Works

In this paper, the design of novel high density, fault tolerant protocol for cluster based routing in ad-hoc networks is about to increase the system performance when the faults occurs within nodes or within the transmission link and to test the system performance in case network nodes are increased within cluster (cluster density). Several network performance metrics have been tested such as the network throughput, the packets loss, the cluster density, the network lifetime to see how the new developed algorithm works. The goal was to investigate current research work on fault tolerance and to

detect the faults in ad-hoc networks and test the system performance in case number of nodes is increased. Currently, we implemented the algorithm that detects the faults during data transmission and elaborate backup nodes/routes so that communication keeps going on. The energy of each node is considered in order to get energy efficient path to send the data between the nodes. Simulation results showed that the proposed clustering scheme gives better performance in terms of cluster size and number of single node clusters. As today, technologies are advancing, in the future work on this paper, we recommend to develop a routing algorithm which handles the fault tolerance problems and high density in other network technologies such LTE or LTE-Advanced where the cells act instead of having networks nodes as it is in ad-hoc networks.

References

1. Rezaee, M., Yaghmaee, M.: Cluster based routing protocol for mobile ad hoc networks. Department of Computer Engineering, Ferdowsi University of Mashhad, Iran, pp. 1–5 (2009)
2. Natsheh, E.: Impact of varying offered data load on density-based routing in mobile ad-hoc networks. *J. Netw. Technol.* **6**, 87–109 (2015)
3. https://www.researchgate.net/publication/269305441_Node_Failure_Effect_on_Reliability_of_Mobile_Ad-Hoc_Networks
4. Baker, D.J., Wieselthier, J.E., Ephremides, A.: A design concept for reliable mobile radio networks with frequency hopping signaling. *Proc. IEEE* **75**(1), 56–73 (1987)
5. Xue, Y., Nahrstedt, K.: Fault tolerant routing in mobile ad hoc networks. In: Proceedings of the IEEE Wireless Communications and Networking Conference, New Orleans, Louisiana, pp. 1174–1179 (2003)
6. Xue, Y., Nahrstedt, K.: Providing fault-tolerant ad hoc routing service in adversarial environments. *Wirel. Pers. Commun.* **29**, 367–388 (2004)
7. Zhou, J., Xia, C.: A location-based fault-tolerant routing algorithm for mobile ad hoc networks. In: WRI International Conference on Communications and Mobile Computing, vol. 2, pp. 92–96 (2009)
8. Qin, Y., Kong L.P.: A fault-tolerance cluster head based routing protocol for ad hoc networks. In: IEEE Vehicular Technology Conference, pp. 2472–2476 (2008)
9. Chandrasekaran, S., Udhayakumar, S., Mohan Bharathy, U., Jitendra Kumar Jain, D.: Trusted fault tolerant model of MANET with data recovery. In: 4th IEEE International Conference on Intelligent Networks and Intelligent Systems, pp. 21–24 (2011)
10. Hauspie, M., Simplot, D., Carle, J.: Partition detection in mobile ad-hoc networks. In: Proceeding of the 2nd Mediterranean Workshop on Ad Hoc Networks, Mahdia, Tunisia (2003)
11. The Network Simulator - ns-2. <http://www.isi.edu/nsnam/ns/>