

A Novel Distributed Algorithm to Maintain Connectivity with Fault Tolerance Scheme in Mobile Ad-Hoc Networks

Jishan Mehedi¹, Kanad Basu², Surendra Dalu³ and M.K.Naskar⁴

Advanced Digital and Embedded Systems Lab

Department of Electronics & Telecommunication Engineering

Jadavpur University, Kolkata – 700032

Email: ¹jmehedi2007@yahoo.co.in, ²kanadbasu2005@gmail.com, ³surendradalu@yahoo.co.in,

⁴mrinalnaskar@yahoo.co.in

Abstract— This paper presents a distributed algorithm to maintain connectivity of nodes in Mobile Ad-Hoc Networks (MANETs) even though fault occurs in the mobile node. It is assumed that each and every node is enabled with GPS receiver and all the nodes periodically communicate their position and velocity information to every node. Proposed algorithm maintains initial topology in every beacon interval. At any instant, if any node becomes faulty and discontinues, then connectivity is always recovered after at least one beacon interval. In proposed algorithm, after receiving information from the nodes each node will modulate its own velocity to maintain connectivity. In this approach initial topology is always maintained, hence there is no routing overhead. Simulation of proposed algorithm is carried out for four hours, for some specific network scenarios (arrangements) and the result obtained from this simulation proves the correctness of the algorithm.

I. INTRODUCTION

A mobile ad-hoc network (MANET) is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers connected by wireless links, the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily, thus, the networks wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the Internet if needed. These networks have an important advantage that they do not require any existing infrastructure or centralized administration. There are various applications of MANET in rescue operation, battle field, and mountain climbing.

As a mobile node can move with unpredicted velocity, the routing and topology management has become an important issue in a MANET. Many researchers have developed different routing protocols which ensure to find the exact route to connect the transmitting mobile

node to its intended caller, which may be outside the transmitting range of the transmitter, via other node(s). Existing routing protocols for MANET can be classified into four different basic categories namely flooding, proactive routing, reactive routing and dynamic cluster based routing. Flooding based routing requires no knowledge of the network topology. Although such protocols are effective under light load conditions, they generate excessive amount of traffic for large networks making it difficult to achieve flooding reliably. Proactive routing protocol is basically a table driven routing protocol where each node pre-computes the route to every possible destination as well as the path to be followed to minimize the cost. The protocol also periodically broadcasts routing information throughout the network. This approach however increases network traffic in highly dynamic networks. Several modified proactive routing protocol have been suggested to minimize the traffic.

Centralized topology management schemes in [1, 3] discuss a self-adaptive movement control algorithm, which gives an idea to maintain network connectivity even during the positional variation of the nodes.

In this paper, we have suggested a distributed algorithm, which not only ensures node connectivity if there is no fault but also ensure the restoring connectivity if connectivity is broken due to fault in nodes during movements within one beacon interval.

II. BACKGROUND CONCEPT

The proposed algorithm maintains connectivity and it is able to restore connectivity, if connectivity is lost due to node failure, within one beacon interval, after the node failure occurs. When there is no fault in the nodes then separation between each and every pair are allowed 'Rth' amount though maximum communication range 'Rmax' is greater than 'Rth' as shown in figure-1. In this

algorithm, initially all the nodes must maintain at the most 'Rth' amount of distance with its just front node. Our intension is to keep each successive pair of nodes within 'Rth' distance. So we define another distance, which is lesser than 'Rth' is termed as 'Rsafe' as shown in figure-1. When distance from its just front node is greater than 'Rsafe' then the node will move with 'RUSH1' velocity to bring the node within safe region. Due to failure of a node if the distance of the node with its front most nodes becomes more than 'Rmax' then the node will not get any information from its just front most node and the node will move with 'RUSH2' velocity in the next beacon interval. We want to maintain initial topology i.e. initial relative position intact, so we have defined another distance 'Rmin', which is the minimum separation between each and every successive pair of nodes along axis of propagation. If the separation between a pair of node become less than 'Rmin', then node behind will stop itself to maintain initial topology. In this algorithm every nodes are transceiver and are enabled with GPS receiver. All the nodes will transmit their position and velocity information periodically. After receiving information the node will take its own decision for movement in the next beacon interval.

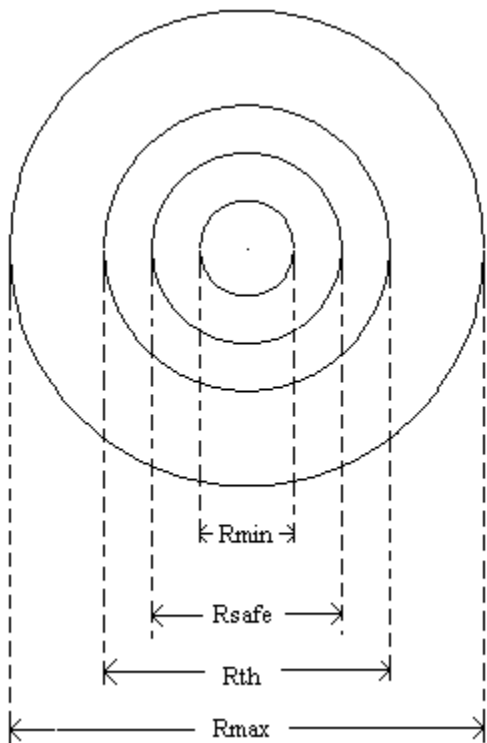


Fig-1: Showing Different distances

III. PROPOSED ALGORITHM

This algorithm is not only maintains the connectivity between the nodes in MANET but also recovers the connectivity, if connectivity is hampered due to failure of a single node. There is no leader or coordinator to control the movements of the node rather each node takes its own decision to change its movement to maintain connectivity and for restoring connectivity if connectivity is lost due to fault in node and again maintain connectivity leaving the faulty node.

A. Assumption:

1. All the nodes are enabled with GPS receiver.
2. Initially all the nodes are connected and maximum separation between each and every pair is not more than Rth.
3. All the nodes are moving in the same direction.
4. All the nodes will broadcast their position and velocity information periodically and this period is termed as beacon interval.
5. All the received information is correct.
6. There is predefined maximum preferred velocity (V_{max}), RUSH1 velocity, RUSH2 velocity and the predefined maximum communication range (R_{max}), maximum initial spacing between nodes (R_{th}), safe distance (R_{safe}) and minimum separation between each pair of nodes (R_{min}).
7. If ' R_{max} ' is maximum communication range, ' R_{th} ' is maximum allowable spacing when there is no fault, ' T ' is beacon interval, ' V_r ' is rush velocity, ' θ ' is maximum angle of deviation from the direction of propagation, then initial Y-axis separation between two nodes is not more than $(\sqrt{(R_{max}^2 - (2 T V_r)^2}) - 2 T V_r \sin \theta) / 2$ where direction of movement is along X-axis.

B. Algorithm:

1. Initially all the nodes will broadcast their position and velocity information.
2. Each and every node will now check which node is just behind and (which node is) just in front of it and that behind node will be behind index and the front node will be front index of that node. If there is no node behind a node means it is the last node and if

there is no front node means it is the front most node.

3. After receiving information each and every node will first check whether they are getting information from their front index node or not.

i). If front index node is not there.

a) Then it will check whether it is the front most node or not. If it is the front most node then it will move with its preferred velocity within the range of maximum velocity.

b) If it is not the front most node then it will check whether it is second front most node or not. If it is the second front most node then, it is sure that there is fault in front most node but there is no need to recover connectivity and it will move with preferred velocity.

c) Otherwise the node will be sure that there is fault in the front index node and move with 'RUSH2' velocity to restore connectivity with remaining nodes.

ii) If front index node is there.

Then it will check whether its distance from the just in front node is greater than safe distance or not.

a). If distance is less than or equal to the safe distance then it will check whether separation along direction of movement is less than or equal to minimum separation 'Rmin' or not. If the distance is less than 'Rmin', then the node will change its velocity to the zero velocity. But if the separation is greater than 'Rmin', then it will take its velocity according to its choice within the range of maximum preferred velocity

b). If distance is greater than the safe distance 'Rsafe', then it will increase its velocity to the 'RUSH1' velocity.

4. Now it will check whether its Y-coordinate has increased or decreased with respect to the initial Y-coordinate. If Y-coordinate is increased or decreased then it will change its direction by twice the angle of deviation from its previous beacon interval in the opposite direction i.e. if the angle of deviation in the previous beacon interval is positive with normal direction of movement then in the next beacon interval it will try to make same amount of negative angle of deviation with the direction of movement of that node.

IV. LEMMAS

A. Lemma-I: If R_{th} is maximum allowable spacing when there is no fault, and if 'Rsafe' is safe distance then if we choose beacon interval 'T' = $(R_{th} - R_{safe}) / V_{RUSH1}$, then no possibility for a node to go out side the 'Rth' .

Proof: According to proposed algorithm maximum relative velocity may be RUSH1 velocity for the following cases:

- i) Case1: The node at behind may move with RUSH1 and the front node with zero velocity.
- ii) Case2: The node at behind may be with zero velocity while the front node with RUSH1 velocity

Case1 can occur only when separation between the nodes are more than safe distance 'Rsafe'. In this case there is no chance to increase their separation since node behind is moving with more velocity rather their separation will decrease.

Case2 can occur only when two nodes are within the safe distance 'Rsafe'. In this case separation between the nodes may increase up to $T \cdot V_{RUSH1}$. Since the two nodes were in this case within safe distance 'Rsafe', so their initial separation may be maximum 'Rsafe'. So to keep the two nodes always within threshold distance 'Rth', after one beacon interval $(R_{safe} + T \cdot V_{RUSH1})$ may be maximum Rth. So in worst case,

$$R_{th} = R_{safe} + T \cdot V_{RUSH1}$$

$$T = (R_{th} - R_{safe}) / V_{RUSH1}$$

When two nodes is out of 'Rsafe' then in no way the node in front may take greater velocity than the node at behind. So if we choose beacon interval $T = (R_{th} - R_{safe}) / V_{RUSH1}$ then there is no possibility for any pair of node to go out side the maximum communication range.

B. Lemma-II: If we allow maximum R_{th} amount of spacing, when there is no fault, though R_{max} is the maximum communication range and if we choose $R_{max} \geq (3 \cdot R_{th} + T \cdot V_{RUSH1}) / 2$, then if any node becomes faulty and due to this fault if connectivity is broken, then it is possible to recover connectivity within one beacon interval.

Proof: Fault may occur in any beacon interval. Let us consider the worst case: If fault is occurred when four successive nodes are separated by 'Rth' amount of distance from its front node as shown in the figure-2

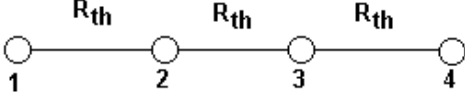


Fig-2: Showing the worst case separation

Now suppose node no.3 becomes faulty. So now the distance between node no.1 and node no.4 is $3 \times R_{th}$ and node no. 2 and node no. 4 become out of communication range. According to this algorithm, after next beacon interval node no. 2 and node no. 4 will regain connectivity without disturbing connectivity among other nodes. Now in worst case node no. 4 may be at R_{USH1} velocity and node no. 1 may be at zero velocity. If we consider this worst case then after next beacon interval separation between node no.1 and node no.4 will be $(3 \cdot R_{th} + T \cdot V_{RUSH1})$. So it is possible to restore connectivity among all the nodes leaving faulty node if and only if

$$3 \cdot R_{th} + T \cdot V_{RUSH1} \leq 2 \cdot R_{max}$$

i.e. $R_{max} \geq (3 \cdot R_{th} + T \cdot V_{RUSH1}) / 2$

C. Lemma-III: Due to movement of node if there is ' θ ' angle of deviation and then if the node can make ' 2θ ' angle of deviation with the opposite direction of the movement of that node in the next beacon interval i.e. it means ' $-\theta$ ' deviation angle with the direction of propagation of the nodes, then two node will be able to acquire their safe distance if they go out of the safe distance but will be within the communication range.

Proof:

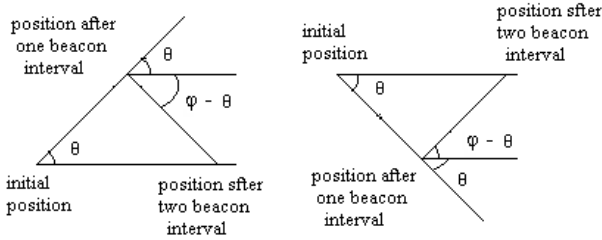


Fig-3: Demonstrating angle of deviation

Let 'Y' is the initial Y-coordinate of a particular node. Now after one beacon interval the node may move with a certain angle of deviation. Let the angle is ' θ '. The node may deviate in positive or negative direction with the direction of propagation as shown in the figure-3. Then after one beacon interval new Y-coordinate of the node will be

$$Y + T \cdot V \cdot \sin(\theta) \quad \text{when the angle of deviation is positive}$$

$$Y - T \cdot V \cdot \sin(\theta) \quad \text{when the angle of deviation is negative}$$

where 'T' is the beacon interval

'V' is the current velocity of the node.

Now in the next beacon interval the node has to make an angle ' ϕ ' with the previous direction of propagation of that node to come back to its original direction of propagation as shown in figure-3.

i) Now consider the velocity remains unchanged, then after next beacon interval Y-coordinate will be

$$Y + T \cdot V \cdot \sin(\theta) - T \cdot V \cdot \sin(\phi - \theta) \quad \text{for the first case}$$

$$Y - T \cdot V \cdot \sin(\theta) + T \cdot V \cdot \sin(\phi - \theta) \quad \text{for the second case}$$

Now the new 'Y' coordinate should be equal to the initial 'Y' coordinate for maintaining no deviation along 'Y'-axis i.e.

$$Y = Y + T \cdot V \cdot \sin(\theta) - T \cdot V \cdot \sin(\phi - \theta) \quad \text{for the first case}$$

$$\text{or,} \quad \sin(\phi - \theta) = \sin(\theta)$$

$$\text{or,} \quad \phi = 2\theta$$

$$Y = Y - T \cdot V \cdot \sin(\theta) + T \cdot V \cdot \sin(\phi - \theta) \quad \text{for the second case}$$

$$\text{or,} \quad \sin(\phi - \theta) = \sin(\theta)$$

$$\text{or,} \quad \phi = 2\theta$$

ii) Now consider velocity is increased, then in the next beacon interval as a result Y-coordinate will decrease for the first case by slightly lower value compared to the previous increment with reference to the initial Y-coordinate and reverse will occur in the second case as shown in figure-3. But, for the next beacon interval there may be a deviation in opposite or same direction according to the algorithm. In the worst case the displacement along Y-axis may be at most maximum deviation that is allowed by this algorithm.

iii) Now consider velocity is decreased, then as a result, after next beacon interval Y-coordinate will be greater than that value for the first case and reverse will occur for the second case. But in the next beacon interval deviation decreases according to algorithm. And in next beacon interval if there is deviation whether positive or negative it will try to minimize that. So movement of the node along Y-axis will be such that difference of distance along Y-axis with respect to initial Y-coordinate will not be more than 'Rth'.

So in each case there is no possibility to go out of the communication range for each pair of nodes. If after few beacon interval, they go out of the 'Rsafe' distance, then they will come back after one beacon interval in the 'Rsafe' distance, if velocity is not changed. But if the velocity is changed, in the worst case, they will comeback after a number of beacon intervals.

D. Lemma-IV: If 'Rth' is maximum allowable spacing when there is no fault, safe distance is 'Rsafe', 'T' is beacon interval, 'V_{RUSH1}' is 'RUSH1' velocity, 'θ' is maximum angle of deviation from the direction of propagation and if initial Y-axis separation between two nodes do not exceed by $(\sqrt{(Rth)^2 - (2 T V_{RUSH1})^2}) - 2 T V_{RUSH1} \sin \theta$ where direction of movement is along X-axis, then two node will not go out the communication range.

Proof: Let maximum allowable separation along Y-axis is 'd₁'. Now consider the case: When separation between two nodes along X-axis is just less than minimum separation 'Rmin' and separation along Y-axis is d₁. In this case again the front node may get 'RUSH1' velocity while the node behind may get zero velocity. So after next beacon interval there is no possibility to increase the separation along Y-axis, but separation along X-axis may increase at the most (T.V_{RUSH1}). So this is the worst case where the separation between two nodes along Y-axis is maximum. Now maximum we can allow 'Rth' amount of separation. So

$$d_1 = \sqrt{(Rth)^2 - (Rmin + T.V_{RUSH1})^2}$$

Again minimum separation = T.V_{RUSH1}

So,
$$d_1 = \sqrt{(Rth)^2 - (T.V_{RUSH1} + T.V_{RUSH1})^2}$$

$$= \sqrt{(Rth)^2 - (2 T.V_{RUSH1})^2}$$

Let initial maximum allowable separation along Y-axis is 'd'. If the maximum allowable angle of deviation with the direction of movement is θ, then maximum separation along Y-axis may occur when both the nodes will move with 'RUSH1' velocity and will make maximum allowable deviation angle with the direction of movement in opposite direction. So maximum separation along Y-axis is (d + 2 T V_{RUSH1} Sin θ).

So,

$$d_1 = d + 2 T.V_{RUSH1} \sin \theta$$

Therefore $d = d_1 - 2 T.V_{RUSH1} \sin \theta$

$$= \sqrt{(Rth)^2 - (2 T.V_{RUSH1})^2} - 2 T.V_{RUSH1} \sin \theta$$

Now if we consider the fault in a node then allowable separation will be half of 'd'.

So if we allow $(\sqrt{(Rth)^2 - (2 T V_{RUSH1})^2}) - 2 T V_{RUSH1} \sin \theta$ / 2 amount of separation along Y-axis then

connectivity will be maintained and connectivity can be recovered if it is broken due to fault in any node.

E. Lemma-V: To keep relative position of the nodes constant, minimum separation along the direction of propagation must be (T.V_{RUSH1}). Where 'T' is the beacon interval and 'V_{RUSH1}' is the RUSH1 velocity.

Proof: Maximum relative velocity between two nodes is 'RUSH1' velocity V_{RUSH1}. So after one beacon interval, maximum change in distance along X-axis between two nodes may be of T.V_{RUSH1}. According to this algorithm, the node behind may take rush velocity only when separation with just front node is more than safe distance 'Rsafe'. So after next beacon interval the node behind will not be able to cross the node just in front of it if we choose minimum separation along direction of propagation = T.V_{RUSH1}

F. Lemma-VI: When there is no fault, then according to this algorithm the node will not go out of 'Rth', if we choose Rsafe = Rth - 2 T Vmax.

Proof: According to this algorithm, if fault do not occur, no node will go out of the threshold distance 'Rth'. We are choosing the safe distance 'Rsafe' as boundary to keep all the nodes within 'Rth'. According to this algorithm if separation between two node becomes more than 'Rsafe' the behind node will take 'RUSH1' velocity to reduce the separation. Maximum relative velocity may be 'RUSH1' when front node is at 'RUSH1' velocity and behind node is at zero velocity with maximum 'Rsafe' amount of separation among the nodes. So in worst case, after next beacon interval, separation may be maximum (Rsafe + T.V_{RUSH1}). Now according to algorithm, maximum allowable separation is 'Rth'.

So,

$$Rth = Rsafe + T.V_{RUSH1}$$

i.e. $Rsafe = Rth - T.V_{RUSH1}$

G. Lemma-VII: 'RUSH1' velocity 'V_{RUSH1}' will be = 2.Vmax and 'RUSH2' velocity 'V_{RUSH2}' will be = (Rth + T.V_{RUSH1}) / (2.T)

Proof: Let us consider the case, when two nodes are at a distance of 'Rsafe'. Then in worst case, in the next beacon interval, the separation between the two nodes may be (Rsafe + T.Vmax). According to the algorithm in next beacon interval the nodes have to come within the safe region. But in worst case the front node may be with the 'Vmax' velocity. So now the behind node will move with 'RUSH1' velocity and relative velocity is now

RUSH1 – V_{max} . Now to bring these two nodes within the safe distance in next beacon interval

$$R_{safe} = R_{safe} + T.V_{max} - (V_{RUSH1} - V_{max}). T$$

i.e. $V_{RUSH1} = 2.V_{max}$

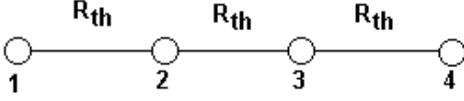


Figure-4: Demonstrating position of the nodes in a particular case

In worst case, when fault occurred, the separation between each pair of node may be at the most ‘ R_{th} ’. Now after the fault the node behind the faulty node will not get any response from the faulty node and in the next beacon interval it will move with ‘ $RUSH2$ ’ velocity to recover the connectivity without disturbing other nodes connectivity. So in worst case separation between the node no.1 and node no.2 may be $(R_{th} + T.RUSH2)$ and this distance may be at the most ‘ R_{max} ’.

Therefore,

$$R_{max} = R_{th} + T.V_{RUSH2}$$

If we choose minimum value of ‘ R_{max} ’ from lemma-II, then

$$R_{th} + T.V_{RUSH2} = (3.R_{th} + T.V_{RUSH1}) / 2$$

$$V_{RUSH2} = (R_{th} + T.V_{RUSH1}) / (2. T)$$

V. SIMULATION RESULTS

For simulation of the algorithm, the threshold distance ‘ R_{th} ’ = 15km, maximum allowable preferred velocity ‘ V_{max} ’ = 60km/hr are taken. So safe distance become ‘ R_{safe} ’=10km, ‘ R_{min} ’=5km and maximum communication range (‘ R_{max} ’) become greater than or equal to 25km. In simulation we have taken ‘ R_{max} ’=25km. The beacon interval is $(15-10) / 2 \times 60$ hr. i.e. 2.5 minutes, RUSH1 velocity = 120km / hr and RUSH2 velocity will be 240 km / hr. Maximum angle of deviation from the direction of movement is taken as 30^0 . So maximum allowable spacing along Y-axis is 3.09 km if direction of movement is along X-axis.

The algorithm is simulated with some sample network scenarios. For each sample, movement snaps after each beacon interval for one hour are shown in tabular form and the simulation result for four hour is shown graphically.

A. Sample-I

The network is of five nodes , their initial coordinates are (0,0) , (15,0), (30,0) , (45,0) , (60,0) and their initial velocities are 50km / hr.,30km / hr.,21km / hr.,10km / hr.,45km / hr respectively.

TABLE1.1: SIMULATION RESULT BEFORE FAULT

Time in minutes	Distance between last two node (km)	Distance between third and fourth node(km)	Distance between second and third node (km)	Distance between first and second node (km)
0	15.0000	15.0000	15.0000	15.0000
2.5	15.1405	15.1936	15.0255	11.9012

TABLE1.2: SIMULATION RESULT AFTER FAULT

Time in minutes	Distance between 1 st pair of node (km)	Distance between 2 nd pair of node(km)	Distance between third pair of node (km)
5	15.1454	30.5632	8.7183
7.5	19.9030	22.8947	8.5470
10	19.7535	19.2176	9.0570
12.5	19.7388	15.0811	9.9689
15	19.7736	11.6881	10.0688
17.5	20.0490	11.6747	6.8575
20	20.0379	7.2454	8.1028
22.5	16.3865	7.4373	8.3042
25	12.5125	7.8424	8.5474
27.5	8.0679	7.6414	10.0258
30	7.9910	12.4695	7.1893
32.5	11.5428	7.7214	8.7945
35	6.8510	9.6191	8.4027
37.5	6.4616	11.5171	8.1684
40	10.4816	8.6258	7.8946
42.5	5.5045	9.6773	8.6713
45	6.2329	9.6049	9.3769
47.5	5.9123	9.7739	9.7040
50	5.1936	10.5291	9.2330
52.2	8.5354	7.1201	9.4831
55	7.7228	8.6605	9.3140
57.5	8.6036	7.4030	10.2799
60	7.2706	11.2854	7.1217

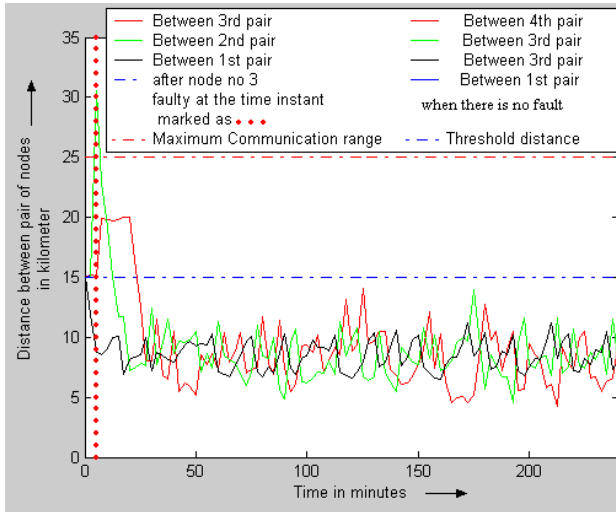


Figure-5: Showing the distance between each pair of nodes for ten hours, for sample-I

B. Sample-II

The network is of five nodes , their initial coordinates are (0,0) , (3,13) , (5,25) , (8,28) , (11,41) and their initial velocities are 30km / hr.,20km / hr.,50km / hr.,6km / hr.,40km / hr respectively

TABLE2.1: SIMULATION RESULT

Time in minutes	Distance between last two node (km)	Distance between third and fourth node(km)	Distance between second and third node (km)	Distance between first and second node (km)
0	13.3417	12.1655	4.2426	13.3417
2.5	13.6256	7.3745	8.4520	10.8410
5	10.0268	7.8828	11.6955	7.9541
7.5	6.7452	11.0131	7.0758	9.8513
10	11.5120	6.5809	8.5003	9.9939
12.5	9.1932	5.2184	9.8963	10.8193
15	8.2531	5.4914	12.6895	8.0933
17.5	6.8699	9.9607	9.0037	9.7105
20	7.4456	8.8962	10.3716	10.1483
22.5	6.5378	13.0149	9.5770	8.2840
25	10.4210	10.4985	8.8300	8.9594
27.5	10.1214	7.2683	8.7642	9.4962
30	7.3705	6.4551	10.0960	9.6519
32.5	7.6718	9.3605	5.5004	11.5908
35	8.3984	9.3597	8.8612	8.9900
37.5	9.5027	8.0775	8.1268	10.6243
40	10.5659	7.2462	11.4934	8.8930

42.5	6.6564	11.1362	8.7295	8.6017
45	10.4684	7.7012	8.8280	9.3912
47.5	7.3234	6.8256	9.5584	10.4304
50	6.4526	7.0075	12.8086	7.5593
52.2	5.7383	11.0820	9.8703	8.2699
55	10.4909	6.2835	10.3518	9.4539
57.5	7.7108	10.3808	6.8161	10.5909
60	11.6347	6.9202	10.6853	7.8471

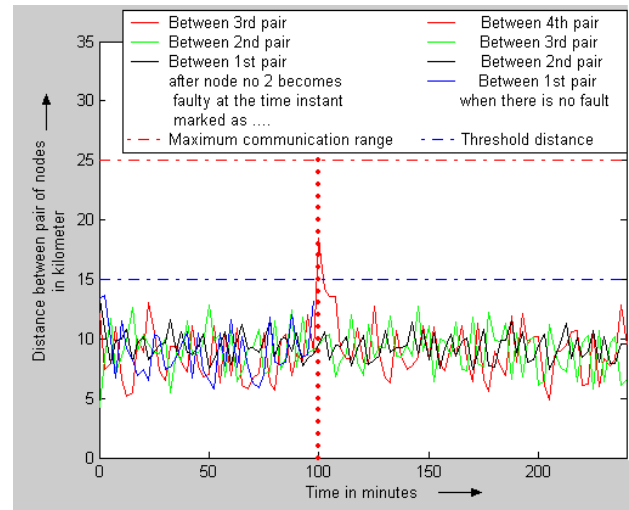


Figure-6: Showing the distance between each pair of nodes for ten hours, for sample-II

C. Sample-III

The network is of five nodes , their initial coordinates are (0, 0) , (15,0) , (25, 5) , (38, 8) , (50, 11) and their initial velocities are 30km / hr.,10km / hr.,50km / hr.,60km / hr.,55km / hr respectively.

TABLE3.1: SIMULATION RESULT BEFORE FAULT

Time in minutes	Distance between last two node (km)	Distance between third and fourth node(km)	Distance between second and third node (km)	Distance between first and second node (km)
0	15.0000	11.1803	13.3417	12.3693
2.5	15.0526	10.1169	14.2297	9.9267
5	15.7252	10.1920	11.2577	10.1022
7.5	15.7742	11.8436	10.6633	7.4563

TABLE3.2: SIMULATION RESULT AFTER FAULT

Time in minutes	Distance between 1 st pair of nodes(km)	Distance between 2 nd pair of nodes(km)	Distance between third pair of nodes (km)
10	15.4952	17.3999	9.2735
12.5	15.5636	13.9058	10.3047
15	15.8841	14.1464	7.5407
17.5	16.1042	10.3139	8.7397
20	16.1411	6.4789	9.7449
22.5	12.0742	8.4556	9.8055
25	7.8288	8.5355	11.6659
27.5	7.2403	11.8296	9.3799
30	9.3151	7.7846	10.4008
32.5	10.8682	10.5458	7.7475
35	10.7991	7.3817	8.3893
37.5	8.1064	6.0831	10.6286
40	8.9668	8.1515	8.6196
42.5	10.3908	5.9447	10.8448
45	7.1943	9.8523	8.0297
47.5	6.3535	11.7533	8.0693
50	10.6027	8.3057	8.6074
52.2	7.0418	7.6823	9.9661
55	6.3542	8.3040	11.2893
57.5	5.5091	11.6821	8.3372
60	10.3394	7.2307	10.1644

remain maintained by discarding faulty node with in one beacon interval.

VII. CONCLUSION

Lot of research work is going on for maintaining connectivity in MANET by using centralized manner or by distributed manner but no one is considering the fault recovery for faulty node, which may be occurred during movement of the nodes. The main theme of the proposed algorithm is that it not only maintains the connectivity of nodes in distributed manner but also able to restore connectivity even though any node becomes faulty during movement. The fault may be due to battery failure or any other failure. Proposed algorithm maintains and restores connectivity without putting any extra control overhead on the system. The proposed algorithm also maintains the initial topology during movements of the nodes and so the routing overhead is nil.

REFERENCES

- [1]. S.S.Basu and A.Chaudhari, "Self-adaptive Topology Management for Mobile Ad- Hoc Network", IE (I) Journal-ET, Vol.84, July 2003
- [2]. David B. Johnson and David A. Maltz "Dynamic Source Routing in Ad Hoc Wireless networks", Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 – 3891
- [3]. Soumya Sankar Basu, Atal Chaudhari, "Self-Adaptive MANET: A Centralized Approach", Foundations of Computing and Decision Sciences, Vol.29, 2004.
- [4]. Elizabeth M. Royer, Santa Barbara and Chai -Keong Toh , "A Review of Current Routing Protocols for Ad-hoc Mobile Wireless Networks", IEEE Personal Communications, April 1999
- [5]. Samir Biswas, Kanad Basu, Avik Ray and M.K.Naskar "A Distributed Approach to Maintain Network Connectivity in Mobile Ad-Hoc Networks", proceedings of 5th Asian Mobile Computing Conference, 2007.

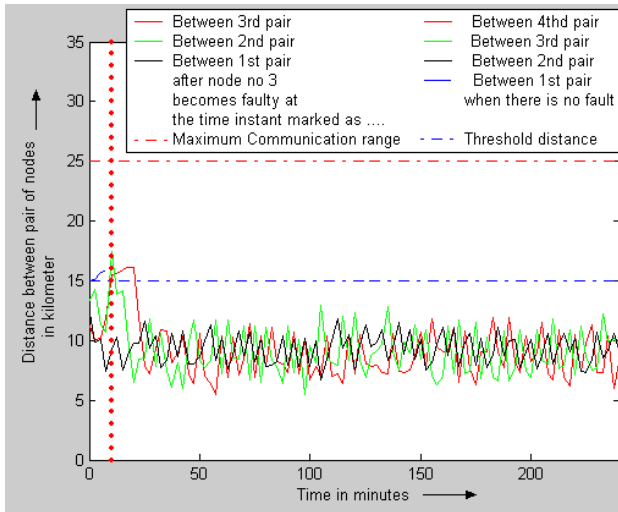


Figure-7: Showing the distance between each pair of nodes for ten hours, for sample-III

VI. OBSERVATION

From the obtained simulation result it is observed that, connectivity is always maintained in nodes when there is no fault. If any fault occurs even though connectivity is