A Fuzzy based Adaptive Distributed Algorithm for Topology Management in Mobile Ad-Hoc Network

Samir Biswas, Jishan Mehedi and M.K.Naskar

Advanced Digital and Embedded Systems Lab,
Department of Electronics & Telecommunication Engineering, Jadavpur University, Kolkata – 700032
E-mail: samir_etceju@yahoo.com ; jmehedi2007@yahoo.co.in ; mrinalnaskar@yahoo.co.in

ABSTRACT

The highly dynamic character of a Mobile Ad-Hoc Network (MANET) poses significant challenges on network communications. Previous work on MANET has resulted in numerous routing protocols aiming to maintain network connectivity among the active nodes. Topology Management can be considered with the same goal. Beside this, it uses a fixed routing table which eliminates the routing overhead in the network. In this paper, we introduce a distributed algorithm for adaptive movement of nodes in a MANET to maintain the overall topology of the network. A fuzzy logic based approach has been incorporated to modify the node velocity. Simulation run on synthetically generated networks indicates the effectiveness of the proposed algorithm and the maintenance of the topology.

Keywords: *Mobile Ad-Hoc Network, Topology Management, Distributed Algorithm, Fuzzy Logic.*

1. INTRODUCTION

A Mobile Ad-Hoc Network (MANET) is a group of wireless autonomous mobile nodes forming a typical environment to communicate with each other dynamically without any fixed infrastructure or administration [1]. In such a network the nodes can move in an arbitrary manner without any prediction. Each node acts as a usual trans-receiver, which can also find the network routes to aid the network to form a complete connected graph. As a result, routing and topology management has become an important issue in a MANET. Efficient routing protocols have been developed which ensure connectivity between transmitting mobile node to its intended caller, may be outside the transmitting range of the transmitter, via other node(s) without having much delay and unnecessary control overhead.

Existing routing protocols for MANET can be classified into four different basic categories namely flooding, proactive routing, reactive routing and dynamic cluster based routing [2]. However none of these routing schemes guarantees constant network connectivity and have constant route maintenance overhead. Even a disconnected node may leads to a non-operating network.

Centralized topology management schemes in [3 and 4] ensure the retention of network connectivity. But there is a coordinator to be elected and all other nodes should follow the instructions from the coordinator to maintain the topology. This increases the control overhead and non-scalability. Once the coordinator fails to perform, the whole network becomes non-functional. Distributed schemes [5, 6] may be a solution to this but at an expense of complex calculation in velocity manipulation.

In this paper, we have suggested a fuzzy logic based distributed topology management algorithm where all the nodes of the network will remain connected and maintained their initial topology throughout the operation so that the nodes can follow a fixed routing, which eliminates the routing control overhead. Each node is provided with a GPS trans-receiver facility that receives position as well as the velocity (in both magnitude and direction) information from its neighbor determined in the initial network establishment stage. With this information each node modifies its velocity in the next beacon interval in a distributed manner so that the topology is maintained. A very simple velocity modification process has implemented and compared with [5], which minimizes the need of complex computing element in each mobile node.

The paper is organized as follows. Next section provides the formal definition of the topology management problem [5]. Section 3 deals with some elementary concept of fuzzy logic [7] related to our work. We present the proposed distributed algorithm for maintaining the topology in section 4. Simulation results and performance comparison are presented in the section 5 and 6 respectively. We finally conclude the paper in section 7. The appendix contains Lemma 1 and Lemma 2.

2. THE TOPOLOGY MANAGEMENT PROBLEM

Given the physical topology of a mobile ad-hoc network, the problem is to control the movements of the individual nodes so as to maintain a stable neighborhood topology so that the nodes are able to communicate amongst themselves without the need of any routing protocols.

Let us consider a MANET consisting of N number of nodes $n_0, n_1, n_2 \ldots n_{N-1}$. We assume that each node has a maximum transmission range of R_{max}. Now, any two nodes n_i and n_j are called neighboring nodes if they can communicate amongst themselves without the need of any routing. So any two nodes n_i and n_j will be neighbors if the distance between them $D(i,j) \le R_{max.}$ Hence the network topology will be maintained if $D(i.j) \le Rmax$ for any two neighboring nodes n_i n_j at any time t.

3. SOME CONCEPTS OF FUZZY LOGIC

3.1 Fuzzy Set

A fuzzy Set A is a set of ordered pairs, given by,

$$A = \{(x, \mu_A(x)) : x \in X\}$$

Where X is a universal set of objects (also called universe of discourse) and $\mu_A(x)$ is the grade of membership of the object x in A. Usually $\mu_A(x)$ lies in the closed interval of [0,1].

3.2 Membership Functions

A membership function $\mu_A(x)$ is characterized by the following mapping:

$$\mu_A(x) \rightarrow [0, 1] : x \in X$$

Where x is a Real number describing an object or its attribute and X is the universe of discourse and A is a subset of X.

E.g. Consider the age of a person is denoted by x. The person may be assigned to the fuzzy set BABY, YOUNG or OLD by the following membership function, shown in Fig. 1:

$$\mu_{BABY}(x) = \exp(-\alpha.x) \quad \text{for } \alpha > 0$$
$$\mu_{YOUNG}(x) = \exp[-(x-25)^2/2\sigma^2] \quad \text{for } \sigma > 0$$
$$\mu_{OLD}(x) = 1- \exp(-\beta.x^2) \quad \text{for } \beta > 0$$

3.3 Fuzzy T-norm

For any two fuzzy sets A and B under a common universe X, the intersection of the fuzzy sets, characterized by a T-norm, is given by

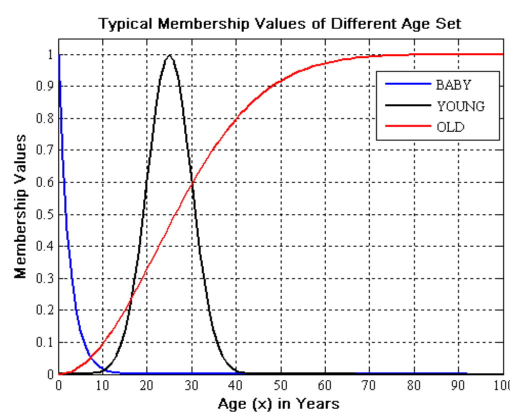

Fig. 1: Illustration of Membership Functions

$\mu_{A \cap B}(x) = \Im(\mu_A(x), \mu_B(x))$

For any membership values a, b, c and d and the T-norm operator T can be formally defined as:

$\Im(0,0) = 0$; $\Im(a,1) = \Im(1,a) = a$; (boundary)

$\Im(a,b) = \Im(c,d)$ if a = c and b = d; (monotonic)

$\Im(a,b) = \Im(b,a)$ (commutative)

$\Im(a, \Im(b,c)) = \Im(\Im(a,b),c)$ (associative)

e.g.:

(a) Minimum: $\Im_{min}(a,b) = \min(a,b)$

(b) Algebraic Product: $\Im_{ap}(a,b) = a.b$

(c) Einstein Product: $\Im_{ep}(a,b) = a.b/[2-(a+b-ab)]$

(d) Drastic Product: $\Im_{ap}(a,b)$ = a if b = 1

= b if a = 1

= 0 otherwise

3.4 Fuzzy S-norm

For any two fuzzy sets A and B under a common universe X, the union of the fuzzy sets, characterized by a S-norm, is given by

$\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x))$

For any membership values a, b, c and d and the S-norm operator *S* can be formally defined as:

$S(1,1) = 1$; $S(a,0) = S(0,a) = a$; (boundary)

$S(a,b) = S(c,d)$ if a = c and b = d; (monotonic)

$S(a,b) = S(b,a)$ (commutative)

$S(a,S(b,c)) = S(S(a,b),c)$ (associative)

e.g.:

(a) Maximum: $S_{max}(a,b) = \max(a,b)$

(b) Algebraic Sum: $S_{as}(a,b) = a+b-a.b$

(c) Einstein Sum: $S_{ep}(a,b) = (a+b)/(1+ab)$

(d) Drastic Product: $S_{ap}(a,b)$ = a if b = 0

= b if a = 0

= 1 otherwise

In this paper, we use minimum function as T-norm and maximum as S-norm operator.

4. PROPOSED ALGORITHM

4.1 System Modeling

(1) All nodes are enabled with GPS trans-receivers which can furnish the current position and velocity of each node itself.

(2) All the nodes have a predefined maximum velocity, V_{max}. In a particular direction x or y, it may attain a maximum velocity of V_{max} /v2 (Proof given in Lemma 2).

(3) Acceleration and deceleration of the nodes are taken to be instantaneous.

(4) It is assumed that if the nodes are within the appropriate range, the instruction message will never be lost in transit.

(5) At the beginning, all the nodes have a configuration that creates a connected topology.

(6) Each node has a unique identification number.

4.2 Definition and Selection of Neighborhood

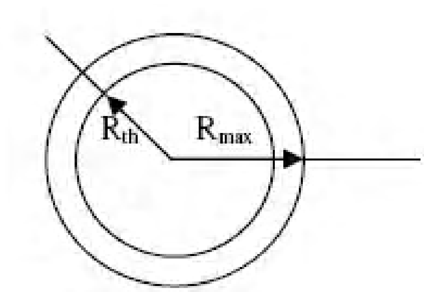

Fig. 2: Illustration of Neighborhood

Let, R_{max} be the maximum range of message communication of each node. R_{th} be the intended range within which pair of nodes called neighbor nodes having 1 in the routing table reside. Then at the beginning a node selects as its neighbor, all nodes which are at a distance less than R_{th} from that node. Next through Hello packets it sends its current position, velocity and node identification number to all its neighboring nodes and also receives the same from its neighbors. A node stores position and velocity information and identification number of its neighbors. This concludes the neighborhood selection procedure. We define a time interval called 'Beacon

Interval' after which the network updates. The choice of R_{th} is given in Lemma 1.

4.3 Concept of Fixed Routing

This algorithm presumes that the routing between any two nodes must follow a fixed table to eliminate routing overhead. In this table, 1 denotes single-hop connection, 0 denotes no direct link but there must be a multi-hop connection between two corresponding nodes as a network must form a connected graph initially.

Table 1: A Typical of Routing Table

	Node 1	Node 2	Node 3	Node 4
Node 1	-------	1	0	1
Node 2	1	-------	1	0
Node 3	0	1	-------	1
Node 4	1	0	1	-------

We must ensure the existence of single-hop connection as in this routing table to maintain the topology at any instant of time. This table must be kept in each node for proper routing.

4.4 Movement Algorithm

The algorithm starts with the breakup of positions and velocities along two mutually perpendicular axes; say X and Y. Nodes transmit the data packet containing their position and velocity in this fashion also.

At the outset, each node is assigned a random position and a velocity, of course complying with the requirement of creating a connected network. Now, each node calculates its relative distance and relative velocity with respect to each of its neighbors. Here, two cases may arise:

Case-1: If the node finds that its relative velocity has the same directional components (i.e., positive or negative) as the relative distance in both X and Y direction, it realizes that it needn't alter its velocity.

Case-2: If the node finds that the above components are not same in either of the directions or both, it realizes that it needs to modify its velocity. This modification is discussed on the condition of the position of the neighbor nodes.

(1) Condition 1: For a node which is in front

Let, the beacon interval = T.

The velocity of node B (as received by A) = $V_{B.}$

The current distance between node A and B = $d_{AB.}$

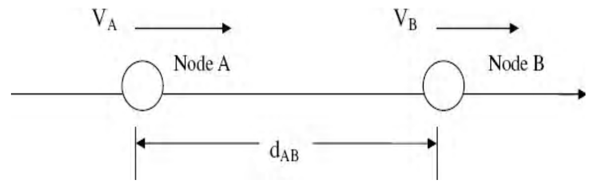

Fig. 3: Illustration of Condition-1

Node A will calculate its velocity such that in the next beacon interval it doesn't cross the node B. Let, the calculated velocity is $V_{A.}$

Therefore,

$$V_A < V_B + d_{AB}/T$$

Or $$(V_A)_{max}|_B = V_B + d_{AB}/T$$

Similarly, node A calculates its maximum velocity for all nodes K_1, $K_{2......}K_N$ in front of it and chooses V_{max} as $(V_A)_{max} = \min((V_A)_{max}|K_1, (V_A)_{max}|K_2 \ldots (V_A)_{max}|K_N)$

Without loss of generality,

$$(V_A)_{max} = \min(V_{K1}, V_{K2} \ldots V_{KN}) + \min(d_{AK1}/T, d_{AK2}/T \ldots . d_{AKN}/T)$$

$$\text{Or, } (V_A)_{max} = \Im(V_{K1}, V_{K2} \ldots V_{KN}) + \Im(d_{AK1}/T, d_{AK2}/T \ldots . d_{AKN}/T)$$

$$\text{Finally, } (V_A)_{max} = \Im(V_{K1}, V_{K2} \ldots) + \Im(d_{AK1}, d_{AK2} \ldots)/T$$

(2) Condition 2: For a node which is behind

Let, the velocity of node C (as received by A) = $V_{C.}$

The current distance between node A and C = $d_{CA.}$

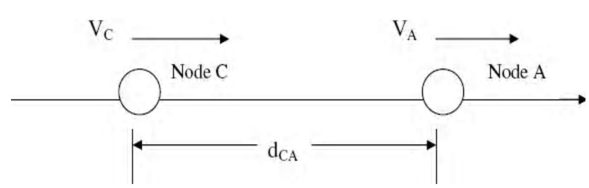

Fig. 4: Illustration of Condition-2

Node A will calculate its velocity such that in the next beacon interval node C is not able to cross itself.

Let, the calculated velocity is $V_{A.}$

Therefore,

$$V_A > V_C - d_{CA}/T$$

Or, $(V_A)_{min}|_C = V_C - d_{CA}/T$

Similarly, node A calculates its minimum velocity for all nodes $K_1, K_{2......}K_M$ which are behind it and chooses V_{min} as $(V_A)_{min} = \max((V_A)_{min}|K_1, (V_A)_{min}|K_2 \ldots (V_A)_{min}|K_M)$

Without loss of generality,

$$(V_A)_{min} = \max(V_{K1}, V_{K2} \ldots V_{KM}) - \min(d_{AK1}/T, d_{AK2}/T \ldots d_{AKM}/T)$$

$$(V_A)_{min} = S(V_{K1}, V_{K2} \ldots V_{KM}) - \Im(d_{AK1}/T, d_{AK2}/T \ldots d_{AKM}/T)$$

Finally, $(V_A)_{min} = S(V_{K1}, V_{K2} \ldots) - \Im(d_{AK1}, d_{AK2} \ldots)/T$

Now, for the next beacon interval, node A must have the velocity between $(V_A)_{max}$, $(V_A)_{min}$. and calculates its velocity as

$$V_{A=}(W_{max}*(V_A)_{max} + W_{min}*(V_A)_{min})/(W_{max}+W_{min})$$

Where W is the weight associated with the two different velocity term $(V_A)_{max}$, $(V_A)_{min}$. We choose W such that the deviation of the calculated velocity from current node velocity should become small.

$$W_i = \exp[-|\{(V_A)_{current} - (V_A)_i\}/\{V_{max} - (V_A)_i\}|]$$

where, i = max, min

$(V_A)_{current}$ is the current node velocity.

V_{max} is the maximum allowed velocity.

If all the nodes calculate its velocity in a similar manner for all its neighboring nodes then the relative position of the nodes remain same with respect to each other. In this way the topology of the whole network remains same.

However, a problem may arise due to this movement algorithm. Although, the main topology remains same, the nodes may move out of the range of each other. To counter the problem, we impose a condition to the movement algorithm.

Consider two nodes A and B as shown in the Fig 2. Let the node A is behind node B and the distance between the two is greater than R_{th} (but obviously less than R_{max}). In this case, it is obvious that velocity of A has to be increased and that of B has to be decreased.

Let, the distance between A,B be R, such that $R>R_{th}$.

Let us define a variable X such that, $X = R-R_{th}$.

Then we propose that the new velocity that A and B must be taking is given by

$$(V_A)_{new} = V_A + X/T;$$

$$(V_B)_{new} = V_B - X/T;$$

Where V_A and V_B are the initial velocities of A and B respectively, and T is the beacon interval.

5. SIMULATION RESULTS

We simulated the algorithm assuming a hypothetical network using Mat lab software in Windows environment and obtained encouraging results. For our simulation we considered five nodes and the system parameters R_{max}=100 Km, V_{max}=80 Km/hr, R_{th}=50 Km. and T=6min. The initial positions and velocities of the nodes were as follows.

Table 2: Initial position and velocities of nodes

Node	Initial Position	Initial Velocity	Neighboring Nodes
1	(0,0)	$35\hat{a}_x + 10\hat{a}_y$	2,3,4,5
2	(30,0)	$25\hat{a}_x + 35\hat{a}_y$	1,4,5
3	(-30,0)	$35\hat{a}_x + 25\hat{a}_y$	1,4,5
4	(0,30)	$45\hat{a}_x + 15\hat{a}_y$	1,2,3
5	(0,-30)	$15\hat{a}_x + 45\hat{a}_y$	1,2,3

Where, $\hat{a}_x$ and $\hat{a}_y$ are the unit vectors along X and Y axis.

The simulation was carried out for an interval of 60 hours=3600 minutes. The results obtained are shown in Fig. 5 to Fig. 10. Fig. 5 and 6 show the initial and final topology of the network respectively. From figures it is proved that the initial topology is maintained.

Fig. 7 and 8 show the distance variation of the neighbors from node 1 and node 3 respectively. It is

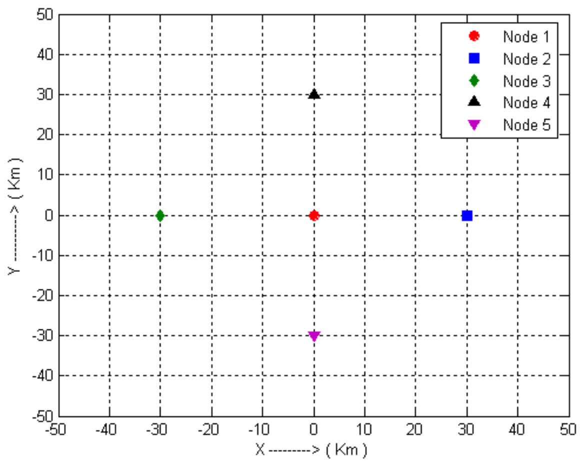

Fig. 5: Initial Topology

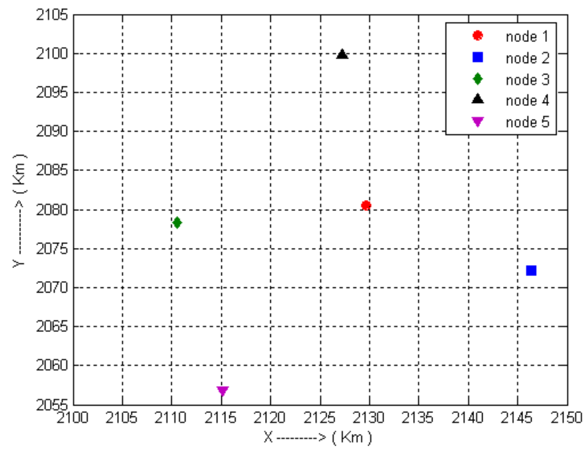

Fig. 6: Final Topology

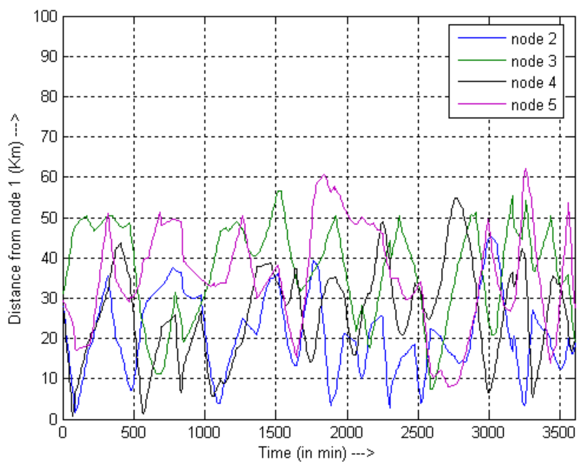

Fig. 7: Variation of distance of neighboring nodes w.r.t. node 1

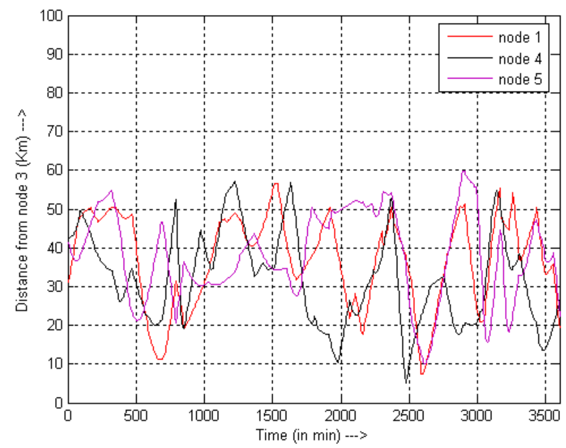

Fig. 8: Variation of distance of neighboring nodes w.r.t. node 3

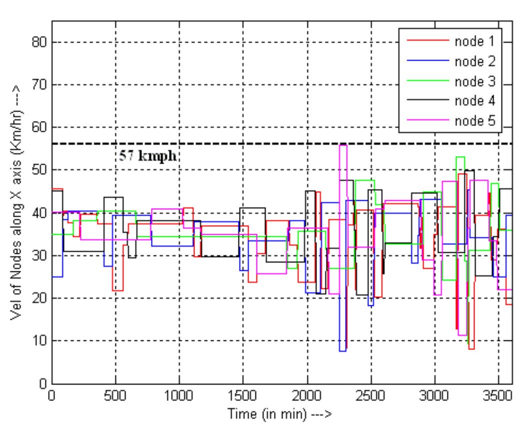

Fig. 9: Velocity of nodes along X-axis

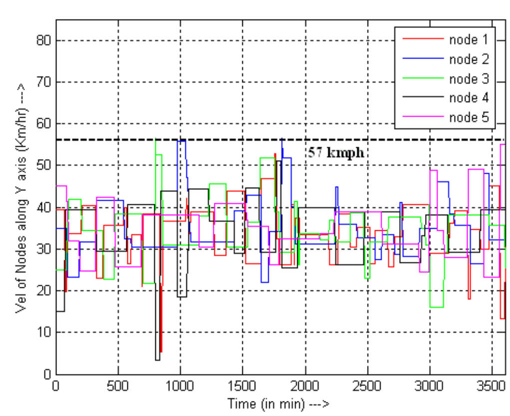

Fig. 10: Velocity of nodes along Y-axis

clear that the distance between the neighboring nodes never exceed the R_{max}. These distances remain less than R_{th} in most of the times. Hence this clearly shows the effectiveness of the proposed algorithm is maintaining the topology of a MANET.

Figure 9 and 10 shows the velocity distribution of different nodes along X and Y axis which are always lesser than or equal to the maximum allowable speed $V_{max}/\sqrt{2}$ (proof in Lemma 2) along a particular axis.

6. PERFORMANCE COMPARISON

We have simulated our algorithm in an environment very similar to that assumed in [3], [5], and [6]. From the results as presented above we see that our algorithm is as effective as [3], [5]. However, this algorithm may be called better as it completely eliminates the control overhead of the coordinator, which was present in the centralized approach. In centralized algorithm [3] the coordinator had to issue appropriate commands to the nodes in order to maintain the network topology and to ensure that a node always remains in contact with the coordinator. But this greatly increases the control overhead of the coordinator. In distributed algorithm [5], the method of velocity modification is much more complex than we have proposed in this paper. This reduces the need of high computing power which in turn reduces the cost per node at the expense that the distance of two neighbor node goes beyond R_{th} for a very small fraction of time. Here a node can suitably adjust its velocity so that the topology of the network always remains the same and at the same time the neighboring nodes always remain within the communication range of each other. Thus, the control overhead of a node [3] is reduced to just sending Hello messages containing its current position and velocity and the computing power [5] is reduced by taking a much simpler approach.

Using the fuzzy based velocity modification process the more flexibility is given to each of the nodes to change the velocity in next beacon interval from current velocity than [6]. In this approach nodes change their velocity by a lesser amount in most of the time which might be an advantage over [6]. We make a comparative study on velocity modification with [6] using the parameter $(V_A)_{max}$ = 70 km/h, $(V_A)_{min}$ = 10 km/h and V_{max} = 80 km/h. So average velocity in next interval V_{avg} = 40 km/h. Fig. 11 and 12 show the Velocity and percentage Change of velocity from current one respectively.

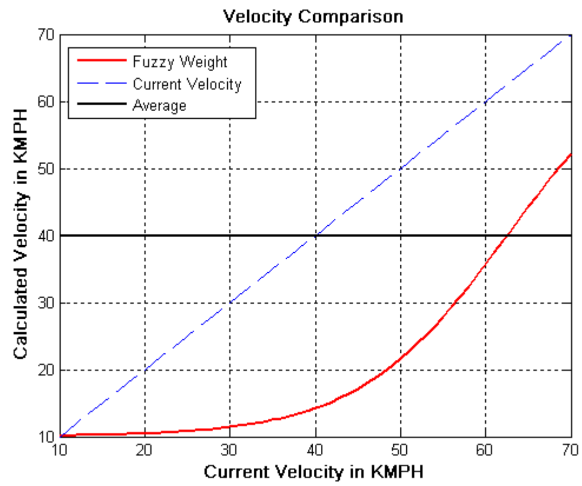

Fig. 11: Velocity Comparison between Fuzzy Weight Case and Average Case

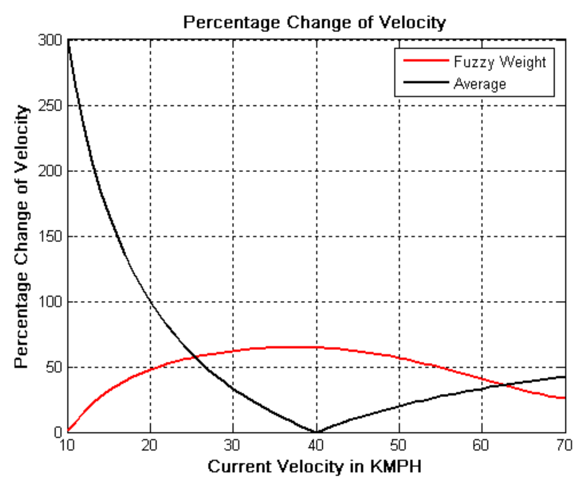

Fig. 12: Percentage Change of Velocity Comparison between Fuzzy Weight Case and Average Case

From Figure12, it is clear that the change is quite small taking the fuzzy weight modification when the current velocity is within a certain range. Here one such example is shown and we find that this is true for different cases. So our fuzzy based model works efficiently in lower velocity range. Normally the MANET finds its application in such an area where it does not require a high velocity and our model worth's itself.

7. CONCLUSION

In this paper, we have introduced a fuzzy based adaptive distributive adaptive algorithm for mobile nodes in a MANET to maintain the network topology. This scheme has its applications even when all the nodes are not moving in the same direction. Due to the presence of the distributed scheme, the control overhead is reduced. Moreover, the topology is not vulnerable if one of the nodes becomes non-functional, as there is no concept of central coordinator. It also requires an elementary computing element to modify the velocity. Future research may be build by considering the data packet loss in the transmission and reception.

REFERENCES

1. National Institute of Standard and Technology, "Mobile Ad-hoc Networks (MANETs)", *http:// w3.antd.nist.gov/wahn_mahn.shtml.*
2. Elizabeth M.Royer and Chai-Keong Toh, "A Review of current routing protocols for Ad-Hoc Mobile Networks", *IEEE Personal Communications, Vol. 6, No. 2, pp. 46-55, April 1999.*
3. S.Samanta, S.S.Ray, S.SenGupta, M.K.Naskar, "A Novel Algorithm for Managing Network Configuration", Asian International Mobile Computing Conference 2006, Kolkata, Jan 04-07, pp. 51-58.
4. S.S.Basu and A.Chaudhari, "Self-adaptive Topology Management for Mobile Ad-hoc Network", *IE(I) Journal-ET, Vol.84, July 2003.*
5. Avik Ray, Kanad Basu, Samir Biswas, Mrinal K. Naskar, "A Novel Distributed Algorithm for Topology Management in Mobile Ad-hoc Networks" , *International Conference on Computers and Devices for Communication 2006 (CODEC 06)*, pp 99-102, December'2006, Kolkata, India.
6. S. Biswas, J. Mehedi, M. K. Naskar, "A distributed algorithm On Topology Management for Mobile Ad-Hoc Networks", *National Conference on Engineering Education*, Feb'2008, Hooghly, West Bengal, India, pp 105-108.
7. Amit Konar, "Computational Intelligence: Principles, Techniques and Applications", published by Springer, 2005.

APPENDIX

Lemma 1: The maximum value for R_{th} is half of R_{max}.

Proof: Consider a case that two nodes A and B are as shown in figure-2. Let the distance between the two is R_{max} in worst case and also assumes node A has an initial velocity 0 and B has a velocity V_{max}.

According to our algorithm, the modified velocity of A is X/T; and the modified velocity of B is V_{max}-X/T; where $X=R_{max}-R_{th}$.

The relative velocity of A with respect to B is $(2X/T-V_{max})$. Now, moving with this velocity, A should not cross B within the beacon Interval T. Imposing this condition, we get,

$$(2X/T-V_{max})T= R_{max}.$$

Replacing value of X, $V_{max}=(R_{max}-2R_{th})/T$;

Since V_{max} is positive, $R_{max}=2R_{th}$

Therefore, $R_{th}=R_{max}/2$; (Proved)

Lemma 2: The maximum allowable velocity along X or Y axis $(V_x)_{max}$ or $(V_y)_{max}$ is $V_{max}/v2$ where V_{max} is the maximum pre-defined velocity.

Proof: We may decompose velocity in two mutually perpendicular axes,

$$V= V_x\, \hat{a}_x + V_y\, \hat{a}_y$$

Now we can write, $V^2 = V_x^2 + V_y^2$

Maximum allowed velocity is V_{max} and if we assume symmetrical distribution of velocity in both axes in maximum case then, $V_x = V_y$ and $V = V_{max}$

So, $2 V_x^2 = V_{max}^2$

Or, $V_x = V_y = V_{max}/v2$. (Proved)