A Novel Algorithm for Managing Network Configuration of a Mobile Ad-hoc Network

Somnath Samanta, Surjya S. Ray, Sourav Sen Gupta and Mrinal K. Naskar
Electronics and Tele-Communication Engineering Department
Jadavpur University, Kolkata 700032
E-mail: somsamhulo@yahoo.co.in

Abstract

In this paper, we propose a topology management algorithm for maintaining a fixed neighborhood topology between the nodes of a mobile ad-hoc network. A frequent change in the relative positions of the nodes in such a network will result in a significant increase in the message overhead. The proposed algorithm elects a movement coordinator which regulates the movements of the nodes by message communication and maintains a fixed network topology thereby reducing message overhead. The simulation run of the algorithm is carried out on a few synthetically generated network scenarios. The results thus obtained show that in all cases, the algorithm maintains a fixed network topology and allows constant connectivity between the nodes.

1. Introduction

A Mobile Ad-Hoc Network (MANET) is a decentralized network of autonomous mobile nodes able to communicate over wireless links without the help of any fixed infrastructure [1,2]. In such network mobile nodes connect dynamically in an arbitrary fashion. These nodes behave as usual trans-receivers and also as routers, taking part in discovery and maintenance of routes to other nodes in the network. MANET is projected to play a vital role in applications like search-and-rescue operations, meetings and conventions, multi-platform battle deployment, mobile sensors or satellite networks for quick sharing of information or data acquisition in inhospitable terrain.

Mobile nodes may move frequently and it may so happen that when a node wants to transmit data to another node, the intended receiver may fall outside the range of communication of the transmitting node. This makes routing an essential requirement in MANET. The current focus of many researchers is to find an efficient routing protocol which will ensure node connectivity whenever required without much delay and unnecessary overhead. There are many existing routing schemes for MANET that can be divided into four basic types namely flooding, proactive routing, reactive routing and dynamic cluster based routing [3].

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. This makes it difficult to achieve flooding reliably [3]. 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 [3] to minimize the traffic. M.Joa-Ng and I.T.Lu proposed a zone based routing protocol [4] where the network is divided into several non-overlapping zones. Reactive routing is a very lazy on-demand approach in the sense that it takes a long time to find a route from source to destination and uses query-response mechanism to find the route. Ad-hoc on-demand distance vector (AODV) routing is a good example of reactive routing. In this approach overhead may increase significantly due to frequent route finding for highly dynamic networks. Several reactive routing protocols have been proposed so far. C.K.Toh proposed association based reactive routing [5] to find a stable route. The temporary ordered routing algorithm (TORA) is another reactive routing scheme where multiple nodes from source to destinations are calculated by localizing the control messages to a very small set of nodes. The main drawback of reactive approach compared to proactive routing is the significant delay of route setup time and also the large volume of control traffic, which is required to support route query mechanism. In dynamic cluster based routing protocol [6] the network is dynamically organized into partitions known as clusters to maintain a stable effective topology. Several clustering algorithms are also proposed, which differ from one another in the criteria used to organize the clusters, such as prediction of node mobility etc. However, none of the proposed schemes guarantees constant network connectivity during the movement and each of these schemes have constant route maintenance overhead. A particular node may even be disconnected in the worst case.

In this proposal, we suggest an efficient self-adaptive movement control algorithm of mobile nodes to ensure the retention of network connectivity even during the positional variation of the nodes. The key concept is to elect a movement coordinator in the network to direct the movement of the other nodes. The nodes must move in such a fashion that the distance from any node to the coordinator does not exceed a predefined maximum value. This maximum range of movement will ensure that, two particular nodes which were neighboring nodes at the beginning will remain so during the movement too. Eventually, the path between any two nodes will not change throughout the entire movement and hence the routing overhead can be eliminated.

The paper is organized as follows. In the next section, we formally define the topology management problem. In the third section, we present the proposed algorithm for maintaining the topology. Section 4 includes the necessary lemmas along with their proofs. The following section presents the simulation results obtained for a number of synthetically designed scenarios. This section also presents a comparative study between the algorithm proposed in [7] and our algorithm. Finally, section 6 concludes the paper.

2. Topology Management Problem

Given a physical topology of a mobile ad-hoc network, the problem is to control the movements of the individual nodes of the network so as to maintain a stable neighborhood topology. The objective is to allow the nodes to communicate amongst themselves without the need of any routing. It has been assumed that the movements of the nodes are unidirectional.

Let us consider a MANET consisting of N number of mobile nodes $n_0, n_1, n_2 \ldots n_{N-1}$. Let us also assume that each node of the network 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 help of any routing. So, the two nodes will be neighbors if and only if $\mathbf{D(i,j) \le R_{max}}$ where D(i,j) is defined as the relative distance between the nodes n_i and n_j. The network neighborhood topology will be maintained if and only if: $\mathbf{D(i,j) \le R_{max}}$ $\forall$ i, j = 0, 1, 2 … N-1.

3. The Proposed Algorithm

The proposed algorithm maintains neighborhood topology in a mobile ad-hoc network through restricted movements of all the mobile nodes using message communication. The basic philosophy of this algorithm is to elect a movement coordinator, which controls the movements of the other nodes to maintain the network neighborhood topology. The algorithm is divided into two parts, namely Coordinator Election Algorithm and Movement Algorithm.

Each individual node in a MANET is a mobile trans-receiver. In this scheme, a node can also vary the transmission range stepwise whenever required. The algorithm prescribes three transmission ranges, namely Shortest Range (R_{min}), Mid Range (R_{mid}) and Longest Range (R_{max}).

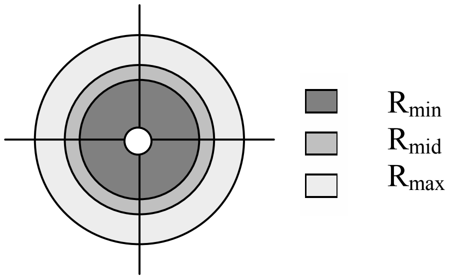


Figure 1: The three communication ranges

The algorithm defines four types of control messages, namely HELLO-message, START-message, STOP-message and RUSH-message. There are two types of RUSH messages: RUSH1 to rush for an interval of T/2 and RUSH2 to rush for an interval of T, where the time interval T is the time to cross the distance between R_{min} and R_{mid} with the predefined maximum velocity V_{max}. Mathematically: $T = (R_{mid} - R_{min})/V_{max}$.

The shortest range of communication, R_{min}, is used by the nodes to send the HELLO message to the coordinator while the middle range, R_{mid}, is used by the coordinator to transmit the control instructions like START, STOP or RUSH for controlling the movement of the nodes. The longest range of communication, R_{max}, is the one used for transmitting actual data packets.

3.1. Zone of Stability

We call a node to be stable if it can pursue its normal movement with its own preferred velocity, and can transmit data packets directly to any other node within the network. We define the stability of the whole network with respect to the position of the coordinator. If all the nodes of the network are stable with respect to the coordinator, we call the whole network to be stable.

The network is said to be unstable if any one of the nodes becomes unstable. As per the proposed algorithm, the coordinator tries to stabilize that node by controlling the movement of the whole network. If all the nodes of the network reside within a particular region, defined as the Zone of Stability, then the network can be stabilized within a finite amount of time by movement control. The Zone of stability is illustrated in the figure 2.

Here, X is the radial distance between Short Range and Medium Range. Therefore, $X = (R_{mid} - R_{min})$. The Zone of stability selected by virtue of Lemma 3.

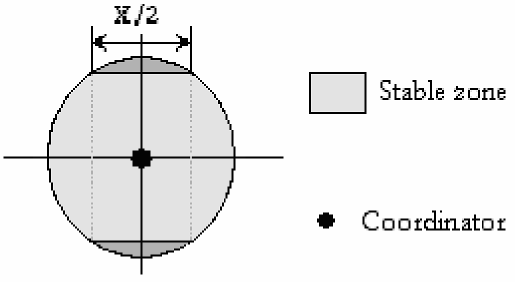


Figure 2: The zone of stability

3.2. Assumptions

- All the nodes must move in the same direction.
- All nodes have a predefined maximum velocity $V_{max.}$
- Acceleration and deceleration of the nodes are taken to be instantaneous.
- If the nodes are within appropriate range, the instruction messages will never be lost in transit.
- At the beginning, there must be at least one node whose distance from all the other nodes is within the zone of stability.

3.3. Coordinator Election Algorithm

Any node, whose distance from all other nodes in the network is within the stable zone as defined above, is eligible for being the coordinator before the network starts moving. All the nodes are provided with an identification number. Two extra messages namely elect and conflict are defined to elect the coordinator.

At the beginning of the movement, all nodes broadcast an elect message to all other nodes using the shortest range of communication. A node that receives the elect message from all other nodes automatically becomes eligible for being the coordinator. If multiple nodes are eligible for being the coordinator, the conflict situation arises. In this case, the node with the least identification number is elected as the coordinator. All the nodes, which are eligible to be a coordinator, send a conflict message to all other nodes along with the identification number, and thus all the nodes come to know about the eligibility of the other nodes. The eligible node that finds that all the other eligible nodes have numbers more than it declares itself as the coordinator. Then it sends the normal start message to all other nodes.

3.4. Movement Algorithm

Each node sends HELLO message to the Coordinator periodically with a period of T/2, using the shortest communication range R_{min}. On receiving the START message, a node can decide to move with its own preferred velocity. RUSH message to a node indicates that the node must move with the predefined maximum velocity V_{max}. The duration of rushing is determined by the two different RUSH messages. Evidently a STOP message stops a node.

Once the coordinator sends a START message to all other nodes in the network, the network starts moving, following the movement algorithm:

Step 1: The coordinator listens to the HELLO messages from all the other nodes at a period of T/2. On missing HELLO message from some nodes it assumes that those nodes have moved out of the range R_{min} from the Coordinator.

Step 2: On the assumption that the nodes has moved ahead, the coordinator first stops those nodes, sends all other nodes the RUSH1 message and rushes along with the well-connected nodes for a time interval of T/2.

Step 3: If the coordinator receives the next HELLO message from the nodes that were stopped, it sends a START message to each of the stopped nodes. On the other hand if next HELLO is not received either from all or from the some of the stopped nodes coordinator detects that those nodes are not ahead but have been left behind. So it sends a RUSH2 message to each of them and sends STOP message to all other well-connected nodes. The coordinator itself also stops. The nodes which receive the RUSH2 message rushes with the predefined maximum velocity V_{max} for a time interval of T.

Step 4: After the above mentioned two steps, the nodes those had moved out from the range R_{min} from the coordinator would surely come back within R_{min} if no emergency situation has crept in (proved in Lemma 2).

Step 5: If at this stage, the Coordinator does not receive HELLO message from some of the nodes, then it declares an emergency situation (may be an accident) and plans accordingly.

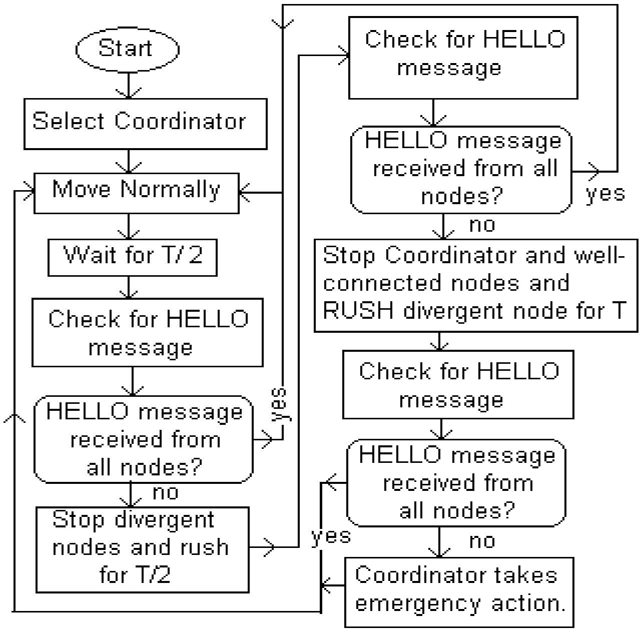


Figure 3: Algorithm flowchart

4. Lemmas

Lemma 1: Mid-Range Selection

The radius of the mid range of communication should be less than or equal to half of that of the longest communication range for maintaining neighborhood criterion.

Proof: Two nodes n_i and n_j are neighbor to each other if $D(i,j) \leq R_{max}$, where $D(i,j) = \| p_i - p_j \| = D(j,i)$. Now, let us consider the position of the coordinator to be p_c. Then, as per our algorithm, we require $D(i,j) \leq R_{max}$,, $D(i,c) \leq R_{mid}$ and $D(j,c) \leq R_{mid}$ for maintaining the network topology.

Now, by triangle law, we get: $\| p_i - p_j \| \leq \| p_i - p_c \| + \| p_c - p_j \|$. That is, $D(i,j) \leq D(i,c) + D(j,c) \leq R_{mid} + R_{mid}$. Therefore, in the worst case, $D(i,j) = 2R_{mid}$. Again, we require $D(i,j) \leq R_{max}$. Therefore, we obtain: $2R_{mid} \leq R_{max} \equiv R_{mid} \leq R_{max}/2$.

Hence, we take the maximum possible limit of the mid communication range, that is $R_{mid} = R_{max}/2$ in case of our algorithm.

Lemma 2: Achieving Convergence

In case of any node going outside the shortest range of communication with respect to the coordinator, this proposed algorithm makes it converge within the zone in no more than a time interval of 3T/2.

Proof: Since we are dealing with maximum time required to converge hence we take only the worst possible case to prove it. In the limiting case let us consider that when the last HELLO message was received the node n_k was just at a distance equal to R_{min}.

Case 1: Let, initially, $D(c,k) = R_{min}$ and n_k be ahead of the coordinator n_c. The relative velocity between the node and the coordinator is $V_{rel} = V_k - V_c$. The node will diverge for $V_k > V_c$, that is, for $V_{max} \geq V_{rel} > 0$. After time interval T/2, we get, $D(c,k) = R_{min} + V_{rel}T/2$. Therefore, $R_{min} < D(c,k) \leq R_{min} + V_{max}T/2$. That is, $R_{min} < D(c,k) \leq R_{min} + X/2$ as $V_{max}T = X$. Now, the node n_k is stopped and the other nodes rush for an interval of T/2. So, after an interval of T/2, we get, $D(c,k)_{current} = D(c,k)_{previous} - V_{max}T/2$. Therefore, $R_{min} - X/2 < D(c,k) \leq R_{min}$ after a total time of T/2 from divergence. So, n_k has converged within a time interval of $T_{conv} = T/2$.

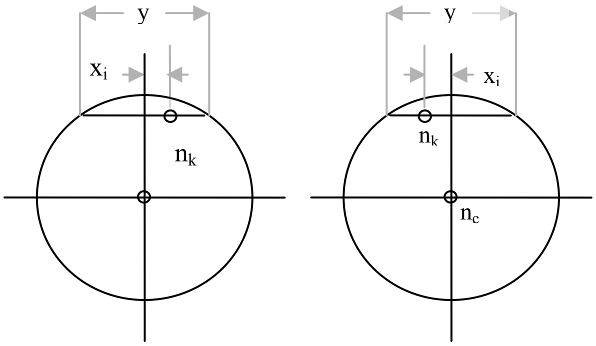
Case 2: Let, in this case, the initial distance of n_k from the coordinator n_c be $D(c,k) = R_{min}$, but n_k is behind_the coordinator. Here, n_k will diverge for $V_k < V_c$, that is, $0 > V_{rel} \geq -V_{max}$. After time interval T/2, we get, $D(c,k) = R_{min} - V_{rel}T/2$. Therefore, $R_{min} < D(c,k) \leq R_{min} + X/2$. Now, the node n_k is stopped while the other nodes rush for an interval of T/2. So, after a time of T/2, we get, $D(c,k)_{current} = D(c,k)_{previous} + V_{max}T/2$. That is, $R_{min} + X/2 < D(c,k) \leq R_{min} + X$. Now as $D(c,k) > R_{min}$, all the other nodes are stopped and n_k rushes for an interval of T. After the interval T, we obtain, $D(c,k)_{current} = D(c,k)_{previous} - V_{max}T$. Therefore, $R_{min} - X/2 < D(c,k) \leq R_{min}$ after a total time of $T/2 + T = 3T/2$ from divergence. So, n_k is converged within a time interval of $T_{conv} = 3T/2$.

As we have considered the worst possible situations on both the ends, we can state that the total time required for the convergence of a node is always less than or equal to 3T/2. That is, $T_{conv} \leq 3T/2$. This proves the lemma.

Lemma 3: Stable Zone

If a node is within the zone of stability as defined before, it will either be stable or can be made stable easily after divergence.

Proof: We take a node n_k within the circle R_{min} such as the horizontal band on which it resides has a length of y, as shown in figures 4. Let the initial horizontal distance of the node n_k from the coordinator n_c is given by x_i.



4(a) 4(b)

Figure 4: Stable Zone Lemma

Case 1: Let n_k be ahead of n_c, as in figure 4(a). So, if the relative velocity $V_{rel} = V_k - V_c$ be positive, then after a time T/2, the distance is $D(c,k) = x_i + V_{rel}T/2 = x_i + X_{rel}/2$, where we denote $X_{rel} = V_{rel}T$. Now, if $D(c,k) = x_i + X_{rel}/2 > y/2$, the node n_k diverges and so it is stopped while the other nodes rush for a time T/2. After this interval of T/2, we get: $D(c,k) = x_i + X_{rel}/2 - V_{max}T/2 = x_i + X_{rel}/2 - X/2$. The divergent node n_k will be converged if and only if $| x_i + X_{rel}/2 - X/2 | \leq y/2$. If we assume that in the first interval of T/2, n_k was just diverged, that is, $x_i + X_{rel}/2 \approx y/2$, then we obtain: $| y/2 - X/2 | \leq y/2 \equiv y \geq X/2$.

Case 2: On the other hand, let us suppose that n_k was x_i distance behind n_c at the beginning, as in figure 4(b). So, if the relative velocity $V_{rel} = V_k - V_c$ be negative, then after a time T/2, the distance is $D(c,k) = x_i + V_{rel}T/2 = x_i + X_{rel}/2$. Now, if $D(c,k) = x_i + X_{rel}/2 > y/2$, the node n_k diverges and so it is stopped while the other nodes rush for a time T/2. After this interval of T/2, we get: $D(c,k) = x_i + X_{rel}/2 + V_{max}T/2 = x_i + X_{rel}/2 + X/2$. Now, the node n_k rushes for a time interval of T while the other nodes remain stationary. After this interval T, we get: $D(c,k) = x_i + X_{rel}/2 + X/2 - V_{max}T = x_i + X_{rel}/2 - X/2$. So, similar to Case 1, the condition that $y \geq X/2$ has to be true so as to make n_k converge within time interval of 3T/2 after divergence.

As we have considered both the extreme cases of motion for node n_k, we can state that the horizontal band on which a node resides while in motion should have a length of $y \geq X/2$ to satisfy the convergence claim (Lemma 1) of our algorithm. We have defined the Zone of Stability as the region where $y \geq X/2$. So, any node residing within the zone will surely follow the algorithm as well as the claim of convergence within time 3T/2.

5. Experimental Results

The performance of the proposed algorithm is discussed in two stages, namely, Simulation Results of the algorithm and Performance Comparison of this proposed algorithm with an already existing topology management algorithm proposed by S S Basu and A Chaudhuri [7].

5.1. Simulation Results

The proposed algorithm is simulated using C programming language on an MS-DOS platform. The simulation is performed on a synthetically designed situation where a MANET with 5 nodes has been considered. All the nodes of the network are allowed to move with their preferred random velocities unless directed otherwise by the movement coordinator.

The initial condition for the simulation is chosen as follows: R_{max} = 100 km, R_{mid} = 50 km, R_{min} = 40 km, V_{max} = 60 km/hour and hence $T = (R_{mid} - R_{min})/ V_{max}$ =10 min. Coordinate-wise positions of the nodes are: $Node_0$ (0,0), $Node_1$ (30,0), $Node_2$ (-30,0), $Node_3$ (0,30), $Node_4$ (0,-30). Initial velocities of the nodes are v_0 = 50 km/hr, v_1 = 55 km/hr, v_2 = 60 km/hr, v_3 = 60 km/hr, v_4 = 20 km/hr where v_i denotes the velocity of the i-th node. All the nodes are moving along the x-direction. We also assume that a node changes its velocity at time nT/2 where n is a non-negative integer. For the simplicity of calculation, we assume that this velocity remains constant for the following time interval T/2.

The results obtained through this simulation are presented in figure 5. The figure shows the plot of the Distance of Each Node from the Coordinator (in km) versus Time (in min) over an interval of 3500 minutes. The horizontal margin at 40 km mark the Shortest Communication range (R_{min}) and the topmost margin at 50 km marks the Mid-Range of Communication (R_{mid}).

It can be seen readily from the plot that though each of the four member nodes chose their individual velocities randomly, they have never moved out of the R_{mid} range from the coordinator. Again, whenever a node crossed the boundary of R_{min} that is whenever the network became unstable, it was stabilized within a maximum time interval of 15 min, which is equal to 3T/2 in this case. Hence the simulation result proves the efficiency of the proposed algorithm.

5.2. Performance Comparison

We have also performed a comparative study of the proposed algorithm with a previous one proposed by S S Basu and A Chaudhuri [7]. The performance comparison is done through the simulation of both the algorithms on a set of synthetically designed network scenarios. In all the simulations, the time intervals are in minutes and the distances are in kilometers. D (*i, j*) denotes the distance between the *i*-th and *j*-th nodes. Unless otherwise stated, the initial locations of the member nodes are: $Node_0$: n_0 (0,0), $Node_1$: n_1 (40, 0), $Node_2$: n_2 (0, 30), $Node_3$: n_3 (-40, 0), $Node_4$: n_4 (0,-30). The Coordinator in all the cases is chosen automatically by the algorithm to be the $Node_0$.

Scenario 1: ***Normal Movement disruption – Overhead increase***

The proposed algorithm guarantees that the network will be stabilized within a maximum time interval of 3T/2, but in the algorithm proposed in [7] does not guarantee this. This is clear from table 1 and 2.

Scenario 2: ***Prediction of Accidents in case of Abnormal Movement***

In the algorithm proposed in [7], when there is a disruption of the normal movement, the coordinator cannot know whether the disruption is caused due to any accident or not. In the following simulation, we consider a situation where a node lags behind the group due to an accident. In [7], the coordinator proceeds with its Stop-Rush paradigm without knowing whether the node is able to follow its instructions at all. Hence the accident is not noticed and no rescue operation takes place. Our proposed algorithm however, is able to identify an emergency situation and the coordinator takes appropriate action. This is clear from table 3 and 4.

Scenario 3: ***Some nodes Diverge in the process of converging some other nodes***

The algorithm in [7] states that if the HELLO message from a node fails to reach the coordinator, the coordinator asks that node to STOP while it, along with the well-connected nodes, move with their *preferred velocities.* But it may so happen that, while stabilizing one node, another node which was previously well-connected, goes out of the shortest range. Our algorithm prevents the occurrence of such a situation as all the well-connected nodes move with same velocities while stabilizing the network. This is clear from table 5 and 6.

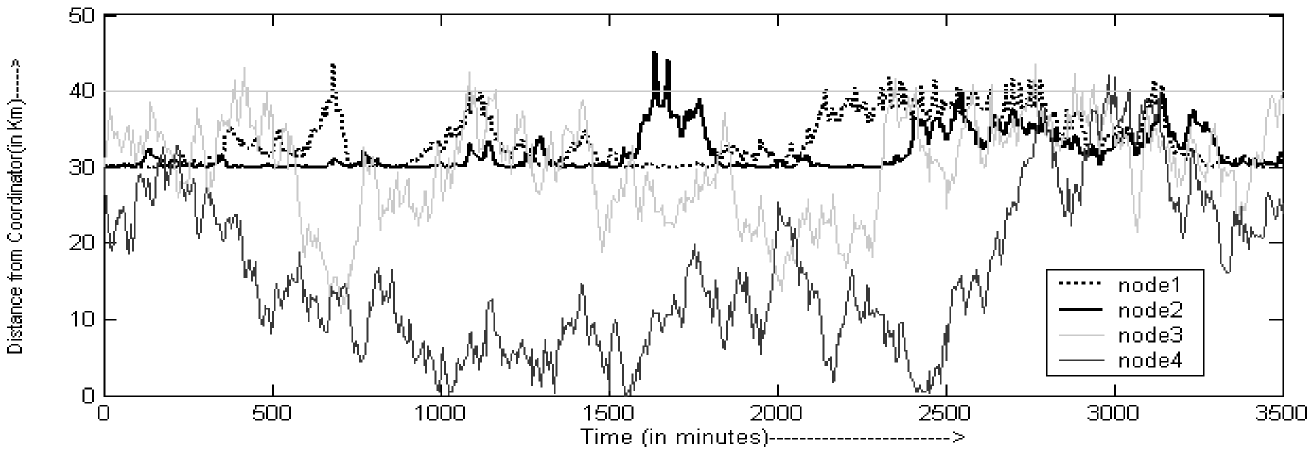


Figure 5: Simulation Result

Table 1: Simulation results for proposed algorithm

Time	D(0,1)	D(0,2)	D(0,3)	D(0,4)	Coordinator's action
0	40.0000	30.0000	40.0000	30.0000	no action taken
5	40.0000	30.0000	42.5000	30.0000	Sends STOP to 3 and RUSH1 to 0,1,2,4
10	40.0000	30.0000	47.5000	30.0000	Sends STOP to 0,1,2,4 and RUSH2 to 3
15	40.0000	30.0000	42.5000	30.0000	----------
20	40.0000	30.0000	37.5000	30.0000	no action taken
25	40.0000	30.0000	40.0000	30.0000	no action taken

Table 2: Simulation results for algorithm presented in [7]

Time	D(0,1)	D(0,2)	D(0,3)	D(0,4)	Action of Coordinator
0	40.000	40.000	40.000	40.000	Coordinator sends a START message to every node
5	40.000	40.000	42.500	40.000	Coordinator requests 3 to STOP
10	40.000	40.000	46.667	40.000	Stops and sends STOP message to 1,2,4 and sends RUSH message to 3
15	40.000	40.000	41.667	40.000	Sends STOP message to 3, RUSH message to 1,2,4 with itself rushing
20	40.000	40.000	46.667	40.000	Stops and sends STOP message to 1,2,4 sends RUSH message to 3
25	40.000	40.000	41.667	40.000	Sends STOP message to 3, RUSH message to 1,2,4 with itself rushing

Table 3: Simulation results for proposed algorithm

Time	D(0,1)	D(0,2)	D(0,3)	D(0,4)	Coordinator's action
0	40.0000	30.0000	40.0000	30.0000	No action taken
5	40.0000	30.0000	42.5000	30.0000	Sends STOP to 3 and RUSH1 to 0,1,2,4
Accident occurs to 3					
10	40.0000	30.0000	47.5000	30.0000	Sends STOP to 0,1,2,4 and RUSH2 to 3 - unable to comply
15	40.0000	30.0000	47.5000	30.0000	-------------
Coordinator does not receive HELLO from 3 after 15 minutes (3T/2) of divergence					
20	40.0000	30.0000	47.5000	30.0000	Coordinator declares an Emergency situation and plans a rescue operation for 3

Table 4: Simulation results for algorithm presented in [7]

Time	D(0,1)	D(0,2)	D(0,3)	D(0,4)	Action of Coordinator
0	40.0000	40.0000	40.0000	40.0000	Sends a START message to every node
5	40.0000	40.0000	42.5000	40.0000	Requests 3 to STOP
Accident occurs to 3					
10	40.0000	40.0000	46.6667	40.0000	Stops and sends STOP message to 1,2,4 Sends RUSH message to 3 - unable to comply
15	40.0000	40.0000	46.6667	40.0000	Sends STOP message to 3 Sends RUSH message to 1,2,4 with itself rushing
20	40.0000	40.0000	51.6667	40.0000	Stops and sends STOP message to 1,2,4 Sends RUSH message to 3
3 has moved out of the mid-range and can not receive instructions from the coordinator					
But, the coordinator has got no hint of the accident occurred to 3					

Table 5: Simulation results for proposed algorithm

Initial location of Node 1: n1 (39.167, 0)

Time	D(0,1)	D(0,2)	D(0,3)	D(0,4)	Coordinator's action
0	39.1667	30.0000	40.0000	30.0000	no action taken
5	40.0000	30.0000	42.5000	30.0000	Sends STOP to 3 and RUSH1 to 0,1,2,4
10	40.0000	30.0000	47.5000	30.0000	Sends STOP to 0,1,2,4 and RUSH2 to 3
15	40.0000	30.0000	42.5000	30.0000	----------
20	40.0000	30.0000	37.5000	30.0000	no action taken

Table 6: Simulation results for algorithm presented in [7]

Initial location of Node 1: n1 (39.167, 0)

Time	D(0,1)	D(0,2)	D(0,3)	D(0,4)	Action of Coordinator
0	39.166668	40.000000	40.000000	40.000000	Sends a START message to every node
5	40.000000	40.000000	42.500000	40.000000	Requests 3 to STOP
10	40.833332	40.000000	46.666668	40.000000	Stops and sends STOP message to 2,4 Sends RUSH message to 1,3
15	45.833332	40.000000	41.666668	40.000000	Sends STOP message to 1,3 Sends RUSH message to 2,4 with itself rushing
20	40.833332	40.000000	46.666668	40.000000	Stops and sends STOP message to 2,4 Sends RUSH message to 1,3
25	45.833332	40.000000	41.666668	40.000000	Sends STOP message to 1,3 Sends RUSH message to 2,4 with itself rushing

6. Conclusion

In this paper, we have developed a novel topology management scheme that retains the network configuration by controlling the movements of the individual nodes of the MANET. The simulation result as well as the comparison with the algorithm in [7] proves the efficiency of the proposed algorithm. Presently, we are working on the generalization of the algorithm by allowing the nodes to move in random directions within an angular limit of ± Δθ with respect to the direction of motion of the coordinator.

7. References

[1] National Institute of Standards and Technology, "Mobile Ad Hoc Networks (MANETs)", *http://w3.antd.nist.gov/wahn_mahn.shtml*.

[2] IEEE 802.11. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification", *Standard,* IEEE, New York, November 1997.

[3] Elizabeth M. Royer and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", *IEEE Personal Communications, Vol. 6, No. 2*, April 1999, pp. 46-55.

[4] M J Ng and I T Lu, "A Peer-to-Peer Zone based Two-Level Link State Routing for Mobile Ad Hoc Networks.", *IEEE Journal on Selected Areas in Communications*, *Special Issue on Ad-Hoc Networks*, August 1999, p 1415.

[5] C K Toh, "Long-lived Ad-Hoc Routing based on the Concept of Associativity", *IETF Draft,* March 1999.

[6] M. Jiang, J. Li and Y. C. Tay, "Cluster Based Routing Protocol (CBRP)", *http://www.ietf.org/internet-drafts/draft-ietfmanet-cbrp-spec-00.txt*, August 1998.

[7] S S Basu and A Chaudhuri, "Self-adaptive Topology Management for Mobile Ad-hoc Network", *IE (I) Journal-ET, Vol. 84*, July 2003.