An Efficient Topology Management Algorithm for Mobile Ad-Hoc Networks

Sourav Sen Gupta, Surjya S. Ray, Somnath Samanta and Mrinal K. Naskar
Electronics and Tele-Communication Engineering Department
Jadavpur University, Kolkata 700032, India
E-mail: *damal2003@rediffmail.com*

Abstract: In this paper, we propose an efficient and robust topology management algorithm for managing a fixed neighborhood topology in a mobile ad-hoc network. The basic concept of the proposed algorithm is to elect a movement coordinator which will control the movements of the nodes by message communication and retain a fixed network topology thereby eliminating routing overhead. The algorithm has been simulated using a few synthetically designed network scenarios and has been compared with a previous algorithm. The results show the effectiveness of the proposed algorithm, both in terms of efficiency and robustness.

Keywords: Mobile Ad-Hoc Network, Topology Management, Zone of Stability, Self-adaptive

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]. MANET is projected to play a vital role in applications like search-and-rescue operations, multi-platform battle deployment, mobile sensors or satellite networks for quick sharing of information or data acquisition in inhospitable terrain.

Routing is an essential requirement in MANET. There are many existing routing schemes for MANET [2]. However, none guarantees constant network connectivity and each has constant route maintenance overhead.

In this proposal, we suggest an efficient movement control algorithm of mobile nodes to maintain the network connectivity even during the positional variation of the nodes, thereby eliminating the routing overhead.

2. PROBLEM DEFINITION

Given a physical topology of a mobile ad-hoc network and the direction of movement of the nodes, the problem is to control the movements of the individual nodes 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.

Let us consider a MANET consisting of N mobile nodes $n_0, n_1, n_2 \ldots n_{N-1}$, each having a maximum transmission range of R_{max}. Two nodes n_i and n_j are called neighbors if they can communicate without the help of any routing. So, they will be neighbors if and only if $D(i,j) \le R_{max}$, $D(i,j)$ being the distance between the nodes n_i and n_j. The network neighborhood topology will be maintained if and only if: $D(i,j) \le R_{max} \quad \forall\ i, j = 0, 1, 2 \ldots N-1$.

3. PROPOSED ALGORITHM

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.

The algorithm prescribes three transmission ranges, namely Shortest Range (R_{min}), Mid Range (R_{mid}) and Longest Range (R_{max}). The algorithm uses five types of control messages, namely HELLO, START, STOP, RUSH1 and RUSH2. R_{min} is used by the nodes to send the HELLO message to the Coordinator while R_{mid} is used by the Coordinator to transmit the other control messages. Actual data packets are transmitted using R_{max}.

3.1 Zone of Stability

If all the nodes of the network reside within a particular region, defined as the Zone of Stability, with respect to the Movement Coordinator, then the network can be stabilized within a finite amount of time by movement control. The Zone of Stability is illustrated in the figure 1 below, where X is the radial distance between Short Range and Medium Range, that is, $X = (R_{mid} - R_{min})$.

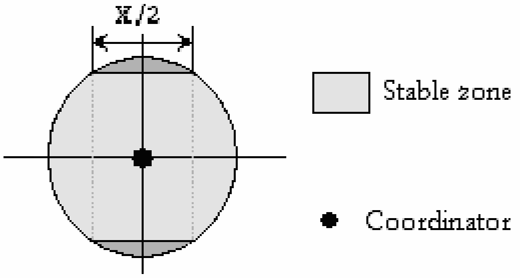

Figure 1: The zone of stability

3.2 Assumptions

The algorithm is based on some primary assumptions:

1. The nodes of the network have unidirectional motion.
2. All nodes have a predefined maximum velocity $V_{max.}$
3. If the nodes are within appropriate range, the messages will never be lost in transit.
4. 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

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. All the nodes, which are eligible to be a Coordinator, send a CONFLICT message to all other nodes along with own identification numbers, and thus all the nodes come to know about the eligibility of the other nodes. The eligible node with the least identification number declares it the Coordinator.

3.4 Movement Algorithm

Each node sends HELLO message to the Coordinator periodically with a period of T/4. 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 T/2 for RUSH1 and T for RUSH2, where $T = X/V_{max}$. Evidently a STOP message stops a node.

Once the Coordinator is elected, it sends a START message to all other nodes in the network. After each node receives the START message the network starts moving. The movement algorithm (flowchart given as figure 2) is as follows:

Step 1: If the Coordinator receives the HELLO messages at an interval of T/4 from ***all*** the nodes, normal motion is continued. But, if the Coordinator misses HELLO message from ***some*** node(s), it initiates Step 2.

Step 2: On the assumption that the divergent nodes have moved ahead, the Coordinator sends STOP message to those nodes, sends all the connected nodes the RUSH1 message and rushes along with those well-connected nodes for a time interval of *T/2*.

Step 3: If the Coordinator receives the next HELLO message from ***all*** of the stopped nodes, it sends a START message to all the nodes. On the other hand, if next HELLO is not received either from ***all*** or from ***some*** of the stopped nodes, Coordinator infers that those nodes are not ahead but have been left behind. So, it sends a RUSH2 message to each of those nodes and sends STOP message to all other well-connected nodes. The Coordinator itself also stops.

Step 4: After the above mentioned two steps, that is, after a time interval of 3T/2, all the divergent nodes will surely converge unless some emergency situation has crept in (proved in Lemma 2). So, Coordinator sends START message to all the nodes. But if the Coordinator does not receive HELLO message from ***some*** node(s) after a time interval of 3T/2, it declares an emergency situation (may be an accident) and plans accordingly.

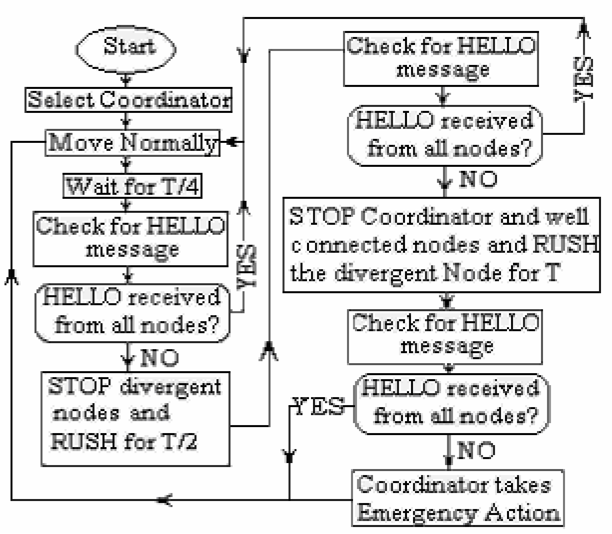

Figure 2: Algorithm flowchart

4. LEMMAS

Lemma 1: ***The radius of the mid range should be less than or equal to half of that of the longest range.***

Proof: Let p_c be the position of the coordinator. We require $D(i,c) \le R_{mid}$ and $D(j,c) \le R_{mid}$ for maintaining the network topology. By triangle law, we obtain: $\| p_i - p_j \| \le \| p_i - p_c \| + \| p_c - p_j \|$, that is, $D(i,j) \le D(i,c) + D(j,c)$. Therefore, in the worst case, $D(i,j) = 2R_{mid}$. Again, we require $D(i,j) \le R_{max}$. So, $2R_{mid} \le R_{max} \equiv R_{mid} \le R_{max}/2$. This proves the lemma.

Lemma 2: ***After divergence from the Zone of Stability, a node is converged back within a time interval of 3T/2.***

Proof: In the worst case, let us consider that when the last HELLO message was received, node n_k was just at a distance equal to $D(c,k) = R_{min}$ from the coordinator n_c. Let the relative velocity be $V_{rel} = V_k - V_c$.

Case 1: Let, n_k is ***ahead*** of n_c. The node will diverge for $V_{max} \geq V_{rel} > 0$. After time interval T/4, we get, $D(c,k) = R_{min} + V_{rel}T/4$. Therefore, $R_{min} < D(c,k) \leq R_{min} + V_{max}T/4$. Now, the node n_k is stopped and the other nodes rush for an interval of T/2, after which $D(c,k)_{new} = D(c,k)_{prev} - V_{max}T/2$. So, after time T/2, $R_{min} - V_{max}T/2 < D(c,k) \leq R_{min} - V_{max}T/4$. So, n_k has converged within a time interval of $T_{conv} = T/2$.

Case 2: Let, n_k is ***behind*** n_c. So, the node will diverge for $0 > V_{rel} \geq -V_{max}$. After time T/4, we get, $D(c,k) = R_{min} - V_{rel}T/4$. Therefore, $R_{min} < D(c,k) \leq R_{min} + V_{max}T/4$. Now, the node n_k is stopped and the other nodes rush for an interval of T/2. So, after T/2, we get, $D(c,k)_{new} = D(c,k)_{prev} + V_{max}T/2$, that is, $R_{min} + V_{max}T/2 < D(c,k) \leq R_{min} + 3V_{max}T/4$. As $D(c,k) > R_{min}$, all connected nodes are stopped and n_k rushes for an interval of T. So, after a time T/2 + T = 3T/2, we obtain, $D(c,k)_{new} = D(c,k)_{prev} - V_{max}T$, that is, $R_{min} - V_{max}T/2 < D(c,k) \leq R_{min} - V_{max}T/4$. So, n_k is converged within a time interval of $T_{conv} = 3T/2$.

Hence, in any case, $T_{conv} \leq 3T/2$. This proves the lemma.

Lemma 3: ***Just after convergence, a particular node will not diverge for a subsequent time interval of T/2.***

Proof: We get from Lemma 2 that, just after convergence, $R_{min} - V_{max}T/2 < D(c,k) \leq R_{min} - V_{max}T/4$. As $| V_{rel} | = | V_k - V_c |$, after a time T/4, we get, $D(c,k)_{new} = D(c,k)_{prev} + V_{rel}T/4$, that is, $D(c,k) \leq R_{min}$. After time T/4 more, $D(c,k)_{new} \leq D(c,k)_{prev} + V_{max}T/4 = R_{min} + V_{max}T/4$. So, the node n_k ***may*** diverge after time T/2 from convergence, but not before that. Hence, just after convergence, $T_{div} \geq T/2$. This proves the lemma.

5. EXPERIMENTAL RESULTS

5.1 Simulation Result: The initial locations of the member nodes are: $n_0(0,0)$; $n_1(40,0)$; $n_2(0,30)$; $n_3(-40,0)$; $n_4(0,-30)$ with respective velocities randomly chosen at each interval. R_{max} =100 km, R_{mid} = 50 km, R_{min} = 40 km and V_{max} = 60 km/hr. The Coordinator is n_0, chosen automatically by the algorithm. The unidirectional motion of the nodes is along the X-axis. Simulation result is shown in figure 3.

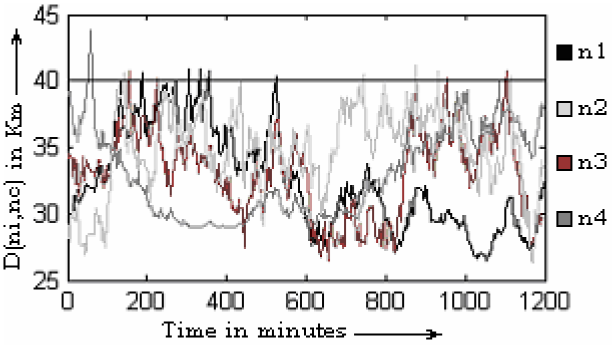

Figure 3: Simulation result

5.2 Comparative Study: We have also performed a comparative study of the proposed algorithm with a previous one proposed by Basu and Chaudhuri [3], and the result thus obtained is shown in figure 4.

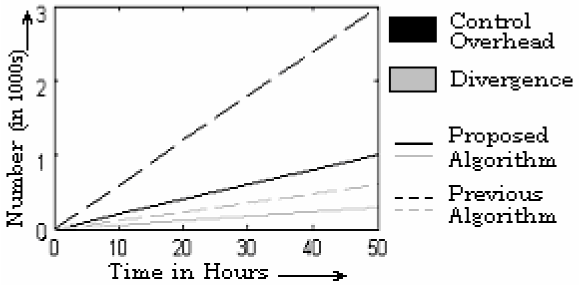

Figure 4: Comparative study

6. CONCLUSION AND FUTURE WORK

This paper presents an efficient topology management scheme that manages the network configuration by effective movement coordination. The experimental results prove the efficiency of the proposed algorithm through the following advantages over [3]:

1. Reduction of Control Message overhead.
2. Reduction in total number of divergences.
3. Prediction of accidents or emergency situations.
4. No nodes diverge while converging other nodes.
5. Prescribing a robust Zone of Stability.

Generalization of the algorithm by allowing the nodes to move in random directions will make it more practical.

REFERENCES

1. National Institute of Standards and Technology, *"Mobile Ad Hoc Networks (MANETs)"*, http://w3.antd.nist.gov/wahn_mahn.shtml.
2. E M Royer and C K Toh, *"A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks"*, IEEE Personal Communications, April 1999.
3. S S Basu and A Chaudhuri, *"Self-adaptive Topology Management for Mobile Ad-hoc Network"*, IE (I) Journal-ET, Vol. 84, July 2003.