Self-adaptive Topology Management For Mobile Ad-hoc Network

S S Basu, *Non-member*
Prof A Chaudhuri, *Fellow*

This paper proposes an algorithm for coordinated adaptive movement of mobile nodes in a wireless ad-hoc network to maintain the neighbourhood topology by message communication. A wireless ad-hoc network operates without a fixed infrastructure and is mobile because of the proposed applications (battlefield, mountain climbing). A coordinator is elected within a network to control movement of the other nodes in the network by directing them to stop, start or rush. The coordinator is not a newly introduced node rather one of the member nodes. The concept is simulated in a few example networks. The sample simulation results, where snaps of movement for continuous two hours are collected, shows that the networks always remain connected during movement, for each network.

Keywords : Mobile; Self adaptive; Network topology; Mobile adhoc network (MANET)

INTRODUCTION

Mobile users can access information and services electronically; regardless of their geographic position, through wireless networks[1, 2]. Demands for more user mobility and portable computing, guide in the development of a class of self- organizing, quickly deployable network architecture known as mobile ad-hoc network (MANET)[3]. A MANET consists of wireless node and operates without any fixed infrastructure. As a result ad-hoc networks cannot rely on dedicated routers for routing like infrastructure networks and mobile nodes in a network expected to act as routers and at the same time adapt the network to the highly dynamic state of its links and movement patterns. MANET is projected to play a vital role in the scenarios where mobile access to an infrastructure network is either ineffective or impossible.

Communication between two arbitrary nodes in an ad-hoc network requires routing through mobile nodes that is expected to move frequently. Node mobility causes frequent link failure. An effective adaptive routing algorithm should have the ability to adjust with this kind of network movement frequently. Current focus of the researchers is to find stable route that can adapt with network mobility.

Existing routing schemes for MANET can be categorized into four types namely flooding, proactive routing, reactive routing and dynamic cluster based routing[3, 4].

Flooding based routing does not require any knowledge of network topology. It can be reasonably robust under light traffic conditions. However, it generates excessive amount of traffic in large networks and is difficult to achieve flooding reliably[5]. In proactive routing protocols, each node pre-computes routes to all possible destinations. The protocol also broadcasts routing information periodically throughout the network. This approach increases network traffic in highly dynamic networks. Several modified approaches have been suggested to minimize the traffic. M Joa-Ng and I T Lu introduced division of the network into non-overlapping zones and proposed a zone based routing protocol[6]. However, proactive approach does not scale well to large, highly dynamic network, although it can ensure high quality routes in a static topology[3].

Reactive routing is a lazy on demand approach and uses query-response mechanism in route finding. Ad-hoc on demand distance vector (AODV) routing[7] is an example of reactive routing. Frequent route finding is a significant overhead in this approach for highly dynamic networks. C K Toh[8, 9] introduced an associativity based reactive routing to probably find a stable route. While finding routes, it considers those nodes that are in long association with the source node. In this direction R Dube[10] proposed signal strength routing to probably find stable stronger route. The temporally ordered routing algorithm (TORA) localizes the control messages to a very small set of nodes and finds multiple route from source to a destination[11]. In a different direction, previous work[12] had introduced transmitter range control to find stable route. However, the main downside of reactive approach compared to proactive routing is significant delay of route setup time and the large volume of control traffic required to support route query mechanism.

The network is dynamically organized to partitions known as clusters to maintain a relatively stable effective topology in dynamic cluster based routing[13]. Several clustering strategies are also proposed[3, 14 – 17]. These strategies differ in the criteria used to organize the clusters such as prediction of node mobility etc.

The major focus of above mentioned approaches is to find a relatively stable route and reduce control message overhead. None of the proposed schemes guarantee constant network connectivity during the movement and each of these schemes have constant route maintenance overhead. A particular node may be disconnected in the worst case.

This paper suggests a self adaptive movement of mobile nodes to ensure the retention of network connectivity during the movement. The basic idea is to elect a movement coordinator in the network to direct the movement of the other nodes. The key concept behind the movement is that nodes move in such a fashion that their distance from the coordinator does not increase a predefined maximum value. This maximum value will ensure that if two particular nodes were neighbouring nodes at the beginning of the movement, they will continue to be so during the movement

---

S S Basu is with Price Water House Coopers Pvt Ltd DA, 83 Saltlake Sector 1, Calcutta - 700 064 and Prof A Chaudhuri is with the Department of Computer Science & Engineering, Jadavpur University, Kolkata 700 032.

This paper was received on September 14,2001. Written discussion on the paper will be received till September 30, 2003.

also. As a result, routes from one node to another will not change throughout the movement and the routing overhead can be overcome.

Before starting movement, mobile nodes elect a node as the movement coordinator, responsible for maintaining the network connectivity. Each node sends a periodic hello message to the coordinator. After analyzing hello message received, coordinator can keep track of the node which tends to move out of the permissible distance and takes necessary action. Coordinator can issue start, stop or rush signal to the nodes.

BACKGROUND CONCEPTS

Each individual node in MANET is a mobile transceiver. In the present approach, a node is not only transceiver but can also vary the transmission range stepwise whenever required. The algorithm intends to use three transmission ranges, the lower two for maintaining connectivity and sending control messages and the highest range to send actual message packets. The middle communication range is the distance limit that each node has to abide by from the coordinator.

The algorithm assumes that the nodes of the network are moving in same direction and have a predefined maximum velocity. Four kinds of control messages are defined in this algorithm to control node movements in the network, *ie*, Hello-Message, Start-Message, Stop- Message and Rush-Message. Each individual node sends Hello- Message to the coordinator in a predefined interval using the shortest communication range. Other three messages are sent by the coordinator to the nodes to control their movements. Start-Message indicates nodes can move with their preferred velocity within the predefined maximum whereas Rush-Message indicates that the node should move with the predefined maximum velocity.

The algorithm is based on a start and stop paradigm. Initially, a node whose distance from all other nodes in the network are within the lowest communication range is elected as the coordinator and after determining the coordinator the network starts moving. Coordinator listens to Hello-Message from all other nodes periodically. On not listening to hello from another node the coordinator detects the distance increment and then decides on the network movement. Again the coordinator can send a Start-Message to all other nodes when their distances from coordinator happen to be within the lowest communication range.

Now it is important to determine the middle communication range, the Hello-Message interval and establish the start and stop paradigm.

Lemma 1

If two neighbouring nodes move in such a way that they are at the most within 50% of the full communication range distance from the coordinator, they will continue to be neighbours throughout the movement.

Proof

The responsibility of the coordinator is to maintain the neighbourhood topology of the network. During the movement, the coordinator and two other nodes can follow any pattern. In the worst case the coordinator and two neighbouring nodes can fall in the same straight line, coordinator being in the middle. In this case, each individual node can be at the most at 50% of their maximum communication range from the coordinator to be able to communicate with each other.

As in the worst case, it can be concluded that if two neighbouring nodes move in such a way that their distance from the coordinator is at the most 50% of the maximum communication range, the nodes will be neighbours throughout the movement.

Lemma 2

The hello-message interval should be half of the minimum time that a particular node takes to travel the difference between the shortest and the medium range.

Proof

The hello-message is meant for maintaining network connectivity. Before starting movement all the nodes are within the shortest communication range from the coordinator. The responsibility of the coordinator is to see that the distance of each node does not increase more than the middle communication range, *ie*, 50% of the maximum range.

All nodes periodically broadcasts hello to the coordinator. On missing a hello from a particular node the coordinator detects whether the particular node has moved away from the lowest communication range. This can happen when the node moves ahead of the coordinator or the coordinator moves ahead of the node.

In the first case the node has to stop and coordinator should move and catch up the node whereas in the second case the coordinator should stop. While moving since the actual position is not known to the coordinator, it is difficult to determine who should stop. To overcome this confusion the message interval should be chosen appropriately.

Without loss of generality, as a terminal case let it be assumed when the coordinator received last hello from the node, it was in the boundary, *ie*, the distance between them was exactly same as the lowest range. Now, if the velocity of the node has a maximum value, the time it takes to travel a distance equal to the difference between the shortest and middle range can be calculated. This time is referred to as T hence forth.

If the Hello-Message interval is fixed to be $T/2$ the coordinator can dynamically prevail over the above-mentioned confusion. First it can arbitrarily decide to stop or stop the node. If the coordinator receives the next hello it is OK otherwise it assumes that the stopped node has moved behind and asks it to rush stopping the other node.

Thus the Hello-Message interval should be half of the minimum time a particular node takes to travel the difference between the shortest and the medium range. The minimum time can easily be calculated if the maximum velocity of each node in the network is defined.

Lemma 3

On missing a hello-message from any node, a coordinator can ensure that the node will not move away from the permissible distance either by stopping itself first or making the other node stop.

Proof

Initially all nodes are within the shortest communication range from their coordinator. On missing a Hello-Message from any node, a coordinator can sense that the node moved out of the shortest communication range. As a border line case suppose it is assumed that the node was just at the shortest communication range distance from the coordinator when the coordinator received the last Hello-Message from it.

Since the algorithm assumes that both the nodes were moving in the same direction, the relative velocity of the other node with respect to the coordinator cannot be equal to the predefined maximum velocity of the network. Thus, the distance of the node will be less than the shortest communication range + half of the difference between the shortest and middle communication range from the coordinator, taking into consideration the hello interval as determined in the above lemma.

If the other node were behind the coordinator, its distance from coordinator will decrease if it moves and the coordinator stops first. If the node is within the shortest communication range during the next hello interval, coordinator will receive the next hello-message from it. In case the coordinator does not receive hello-message, it will make the node stop and rush. In this case, the distance between the coordinator and the other node will increase, but will not be more than the middle communication range during the next hello interval. The stop and rush continues.

On the other hand, if the other node be ahead of the coordinator, its distance will increase on moving and the coordinator stops first. Coordinator misses the next hello from the node and starts rushing. The distance of the node from the coordinator will be less than or equal to the middle range when the coordinator starts rushing. Again stop and rush continues.

Similar argument holds good if the coordinator makes the other node stop first rather than stopping itself. Thus, a coordinator can maintain connectivity either by stopping itself first or asking the member node to stop first on missing a hello-message from a member node.

ALGORITHM

This algorithm enables maintaining neighbourhood topology in a MANET through restricted movement by message communication. Basically it can be divided into two parts, (i) coordinator election, and (ii) movement.

Assumptions

The algorithm poses some assumptions on the network. They are :

- The algorithm is applicable for those networks, the nodes of which move in the same direction, *eg*, a group of mountain climbers.
- All nodes should have a maximum velocity. As the algorithm is based on a stop start paradigm, it is not possible to calculate control message sending periodicity without having this knowledge. However nodes can move with their own velocity when the network is stable.
- Accelerating or decelerating of the velocities of the nodes is considered instantaneous for the sake of simplicity in calculating the interval between hello messages.
- Control messages sent from a source node to a destination node is lost in transit only if the destination node moves out of the transmission range from the source node.
- At the beginning there should be at least one node in the network, whose distance from the other nodes is within the shortest communication range. This node will be elected as the coordinator.
- All nodes in the network should know the total number of nodes in the network.

Coordinator Election Algorithm

Any node, whose distance from all other nodes in the network is within the shortest communication range, is automatically elected as the coordinator before the network starts moving. To facilitate the election process all the nodes are given a unique number, and the node having the least number has the priority in case of any conflict. Two extra messages, *eg*, Start-Request and Wait are defined to elect the coordinator.

Initially, all nodes broadcast a Start-Request-Message to all other nodes using the shortest communication range. A node that receives start-request from all other nodes becomes eligible for being the coordinator. It may happen that multiple nodes are eligible for being the coordinator. Node with the least number is elected as coordinator to resolve the conflict in that case.

All nodes, eligible for being a coordinator, broadcast Wait- Message to all other nodes with a unique number, and thereby all nodes know about the eligibility of the other nodes. An eligible coordinator on finding that there are no other eligible coordinators declares itself as the coordinator and sends the Start-message to all other nodes to start network movement. All other nodes update their coordinator information. The following example illustrates the process.

Example

Suppose in the example scenario, the network consists of six nodes. Nodes are numbered from 1 to 6. According to the above mentioned convention, node numbered '*i*' will have greater priority than '*i*'+ 1 in coordinator election stage.

In the beginning the nodes flood the network with a Start-Request message. Let it be assumed that the network positions are such that node 2 and 4 receive message from all other nodes. Nodes 2 and 4 become eligible to be coordinator. They send Wait-Message to all other nodes in the network. On receiving Wait-Message from node 2, node 4 leaves its claim. On the other hand node 2 receives Wait from node 4, that has a higher number than it. Node 2 declares itself the coordinator and sends the Start-Message to the remaining five nodes.

Movement Algorithm

The algorithm defines four types of control messages namely Hello-Message, Start-Message, Stop-Message and Rush-Message. Each node destined to the coordinator in a predefined interval, using the shortest communication range, sends hello-message. The coordinator, using the medium communication range sends other three messages to other nodes. On receiving Start Message, a node can decide to move with it's own velocity.

Rush message to a node indicates that the node must move with the maximum defined velocity. Evidently 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 pattern is as follows :

- The coordinator listens to hello message from other nodes. On missing hello from some nodes, it detects whether those nodes have moved out of the shortest communication range from the coordinator.
- On the assumption that the nodes have moved ahead, the coordinator first stops those nodes, and keeps the other nodes which are moving within the shortest range.
- If the coordinator receives the next hello 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 some of the stopped nodes, coordinator detects that these nodes are not ahead, but have been left behind, so sends a rush message to each of them and sends stop message to all other well-connected nodes. The coordinator itself also stops.
- At this point, the network virtually becomes divided into two parts. Coordinator and the nodes that are well connected to it are stopped. Nodes that are not well connected to the coordinator are moving.
- This stop and rush protocol goes on until all the nodes become well connected to the coordinator again.

Table 1 Movement showing snap after each hello message internal

Time in Minute	Distance Between Node 1 and 2 in km	Distance Between Node 1 and 3 in km	Coordinator Actions with Remarks
0	40	40	Coordinator sends start signal to the other two nodes.
5	40.41667	40.41667	Coordinator detects both node 2 and 3 moved out of shortest range. It sends stops signal to both the nodes. Coordinator moves with its original velocity.
10	36.25000	44.58333	Node 3 is still out of shortest range. So the coordinator sends a rush message to it. As node 2 is now well connected, coordinator stops node 2 with itself.
15	36.25000	39.58333	Each node is well connected to the coordinator so, the coordinator starts and sends start signal to the other two nodes. Node 2 and 3 can now move with their own velocity along with the coordinator.

Table 2 Sample movement for preferred velocity of 55 km/hr for node 3 (sample 1)

Time in Minutes	D (1, 2) in km	D (1, 3) in km	D (1, 4) in km	D (1, 5) in km	Actions of the Coordinator
0	40	40	40	40	Coordinator sends a start message to every node.
5	40	40.41667	40	40	Coordinator requests 3 to stop.
10	40	36.25	40	40	Coordinator sends a start message to node 3.
15	40	36.66667	40	40	None
20	40	37.08333	40	40	None
25	40	37.5	40	40	None
30	40	37.91667	40	40	None
35	40	38.33333	40	40	None
40	40	38.75	40	40	None
45	40	39.16667	40	40	None
50	40	39.58333	40	40	None
55	40	40	40	40	None
60	40	40.41667	40	40	Coordinator requests 3 to stop.
65	40	36.25	40	40	Coordinator sends a start message to node 3.
70	40	36.66667	40	40	None
75	40	37.08333	40	40	None
80	40	37.5	40	40	None
85	40	37.91667	40	40	None
90	40	38.33333	40	40	None
95	40	38.75	40	40	None
100	40	39.16667	40	40	None
105	40	39.58333	40	40	None
110	40	40	40	40	None
115	40	40.41667	40	40	Coordinator requests 3 to stop.
120	40	36.25	40	40	Coordinator sends a start message to node 3.

In between the stop and rush, coordinator might miss a hello from some other node, which was well-connected previously. According to the normal convention described earlier coordinator should ask the node to stop. At the same time it may so happen that the coordinator has to stop along with other well connected nodes to connect the previous set of non-connected nodes. Thus the node that has just moved out of range cannot be stopped. Instead the node is asked to rush.

If coordinator misses the next hello from the same node, surely the node is ahead of the coordinator as otherwise the node will be within the shortest communication range and the hello message cannot be missed. Hence on missing the next hello from the same node, coordinator rushes with all well-connected nodes and asks that node to stop. Then stop and rush again continue. The following example illustrates the movement algorithm.

Example

The coordinator maintains the connectivity using the movement algorithm. Shortest communication range and maximum velocity can be chosen to optimize the hello-message interval.

In this example, there are three nodes in the network. The largest communication range is 100 km and the middle range is 50 km. Let the shortest range be assumed to be 40 km. The maximum velocity of the nodes is 60 km/hr, so the hello message interval is 5 min. Nodes are labelled as 1, 2 and 3, 1 being the coordinator. As a boundary case, let the network start moving with all the nodes in a single line and the coordinator at the middle. Let the velocities of the nodes be 50, 55 and 45 km/hr respectively. Table 1 illustrates the movement showing the snap after each hello message interval.

SIMULATION

In the simulation environment the largest communication range is 100 km, the middle range is 50 km and the shortest range is 40 km. The maximum velocity of the nodes is 60 km/hr, and hence the hello message interval is 5 mins. The node number denotes individual nodes, *ie* node i is denoted by i; $D(i, j)$, the distance from node i to node j.

The algorithm is run with a few sample networks. For each sample, movement snaps are shown for 2 hours each in 5 min interval. Each sample consists of initial network positions description and sample movement.

Table 3 Sample movement for preferred velocity of 45 km/hr for node 3 (sample 2)

Time in Minutes	D (1, 2) in km	D (1, 3) in km	D (1, 4) in km	D (1, 5) in km	Actions of the Coordinator
0	40	40	40	40	Coordinator sends a start message to every node.
5	40	40	40	40.41667	Coordinator requests 5 to stop.
10	40	40	40	44.58333	Coordinator stops and sends stop message to 2, 3, 4. Sends rush message to 5.
15	40	40	40	39.58333	Coordinator starts and sends start messages to every node.
20	40	40	40	40	None
25	40	40	40	40.41667	Coordinator requests 5 to stop.
30	40	40	40	44.58333	Coordinator stops and sends stop message to 2, 3, 4. Sends rush message to 5.
35	40	40	40	39.58333	Coordinator starts an sends start messages to every node.
40	40	40	40	40	None
45	40	40	40	40.41667	Coordinator requests 5 to stop.
50	40	40	40	44.58333	Coordinator stops and sends stop message to 2, 3, 4. Sends rush message to 5.
55	40	40	40	39.58333	Coordinator starts and sends start messages to every node.
60	40	40	40	40	None
65	40	40	40	40.41667	Coordinator requests 5 to stop.
70	40	40	40	44.58333	Coordinator stops and sends stop message to 2, 3, 4. Sends rush message to 5.
75	40	40	40	39.58333	Coordinator starts and sends start messages to every node.
80	40	40	40	40	None
85	40	40	40	40.41667	Coordinator requests 5 to stop.
90	40	40	40	44.58333	Coordinator stops and sends stop messages to 2, 3, 4. Sends rush message to 5.
95	40	40	40	39.58333	Coordinator starts and sends start messages to every node.
100	40	40	40	40	None
105	40	40	40	40.41667	Coordinator requests 5 to stop.
110	40	40	40	44.58333	Coordinator stops and sends stop message to 2, 3, 4. Sends rush message to 5.
115	40	40	40	39.58333	Coordinator starts and sends start messages to every node.
120	40	40	40	40	None

Sample 1

The network consists of five nodes, node 1 being the coordinator. Keeping coordinator in (0, 0) the coordinates for other nodes are (0, 40), (40, 0), (0, – 40) and (– 40, 0) respectively. The unit is 1 km. The preferred velocity for node 3 is 55 km/hr, for others it is 50 km/hr. Table 2 shows the movement.

Sample 2

The network consists of five nodes, node 1 being the coordinator. Keeping coordinator in (0, 0) the coordinates for other nodes are (0, 40), (40, 0), (0, – 40) and (– 40, 0) respectively. The unit is 1 km. The preferred velocity for node 5 is 45 km/hr, for others it is 50 km/hr. Table 3 shows the movement.

Sample 3

The network consists of five nodes, node 1 being the coordinator. Keeping coordinator in (0. 0) the coordinates for other nodes are (25, 25), (– 25, 25), (– 25, – 25) and (25, – 25) respectively. The unit is 1 km. The preferred velocities of the nodes are 50, 55, 60, 45, 40 km/hr respectively. Table 4 shows the movement.

From the sample results it is found that no node is moving out of the 50% range from the coordinator. The lowest range is chosen as 40% to increase hello message interval and network stability. In the first two cases network gets stabilized after some stop start, but in the third case after certain time the stop start continues by virtue of the restricted movement, the neighbourhood for each node is maintained throughout.

CONCLUSION

This paper describes an adaptive movement algorithm of mobile nodes in a wireless ad-hoc network to maintain neighbourhood topology. The algorithm defines some control traffic but results in removal of routing overhead. Virtually through a restricted movement pattern the network behaves as a wireless LAN. The algorithm has been applied to few sample cases. In future fault tolerance can be included in the algorithm to make it more rugged.

Table 4 Sample movement for preferred velocities of 50, 55, 60, 45, 40 km/hr (Sample 3)

Time in Minutes	D (1, 2) in km	D (1, 3) in km	D (1, 4) in km	D (1, 5) in km	Actions of the Coordinator
0	35.35534	35.35534	35.35534	35.35534	Coordinator sends a start message to every node.
5	35.65118	35.94942	35.65118	35.94942	None
10	35.94942	36.55285	35.94942	36.55284	None
15	36.25	37.16517	36.25	37.16517	None
20	36.55285	37.78595	36.55285	37.78594	None
25	36.84793	38.41477	36.85793	38.41417	None
30	37.16517	39.05125	37.16517	39.05124	None
35	37.47453	39.69502	37.47453	39.69501	None
40	37.78595	40.34573	37.78595	40.34572	Coordinator sends stop message to 3 and 5.
45	38.09938	37.16517	38.09938	43.69242	Coordinator stops and sends stop message to 2, 3, 4 and rush message to 5.
50	38.09938	37.16517	38.09938	38.92335	Coordinator starts and sends start messages to every node.
55	38.41477	37.78595	38.41477	39.56569	None
60	38.73207	38.41477	38.73207	40.21504	Coordinator sends stop message to 5.
65	39.05125	39.05125	39.05125	43.55583	Coordinator stops and sends stop message to 2, 3, 4 and rush message to 5.
70	39.05125	39.05125	39.05125	39.56569	Coordinator starts and sends start messages to every node.
75	39.37224	39.69502	39.37224	40.21504	Coordinator sends stop message to 5.
80	39.69502	40.34573	39.69502	43.55583	Coordinator stops and sends stop message to 2, 4 and rush message to 5 and 3.
85	39.69502	44.37842	39.69502	39.56569	Coordinator sends stop message to 3 and rush message to 2, 4, 5 with itself rushing.
90	39.69502	40.34573	39.69502	39.56569	Coordinator stops and sends stop message to 2, 4, 5 and rush message to 3.
95	39.69502	44.37842	39.69502	39.56569	Coordinator sends stop message to 3 and rush message to 2, 4, 5 with itself rushing.
100	39.69502	40.34573	39.69502	39.56569	Coordinator stops and sends stop message to 2, 4, 5 and rush message to 3.
105	39.69502	44.37842	39.69502	39.56569	Coordinator sends stop message to 3 and rush message to 2, 4, 5 with itself rushing.
110	39.69502	40.34573	39.69502	39.56569	Coordinator stops and sends stop message to 2, 4, 5 and rush message to 3.
115	39.69502	44.37842	39.69502	39.56569	Coordinator sends stop message to 3 and rush message to 2, 4, 5 with itself rushing.
120	39.69502	40.34573	39.69502	39.56569	Coordinator stops and sends stop message to 2, 4, 5 and rush message to 3.

REFERENCE

1. E C Prem. 'Wireless Local Area Networks.'

2. P Misra. 'Routing Protocols for Ad Hoc Mobile Wireless Networks.'

3. A Bruce McDonald and T Znati. 'A Mobility based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks.' *IEEE Journal on Selected Areas in Communications*, vol 17, no 8, August 1999.

4. M Gerla, G Pei and S J Lee. 'Wireless, Mobile Ad-Hoc Network Routing.' Computer Science Department, University of California, Los Angeles. *Presented at IEEE/ACM Focus'99*, New Brunswick, NJ, May 1999.

5. R Perlman. 'Fault-tolerant Broadcast of Routing Information.' *Computer Networks*, vol 7, 1983, pp. 395-405.

6. 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.

7. C Perkins and E M Royer. 'Ad Hoc on Demand Distance Vector Routing.' *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp 90-100.

8. C K Toh. 'Long-lived Ad-Hoc Routing based on the Concept of Associativity.' *IETF Draft*, March 1999.

9. C K Toh. 'A Novel Distributed Routing Protocol to Support Ad hoc Mobile Computing.' *Proceedings of 1996 IEEE 15th Annual International Phoenix Conference on Computing and Communications*, March 1996, pp 480-486.

10. R Dube, *et al*. 'Signal Stability based Adaptive Routing for Ad Hoc Mobile Networks.' *IEEE Pers Comm*, February 1997, pp 36-45.

11. V D Park and M S Corsion. 'A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks.' *Proceedings of INFOCOM'97*, April 1997.

12. S S Basu, A Nag, S Choudhury and A Chaudhuri. 'Modified AODV Algorithm for Ad Hoc Mobile Network for Increased Throughput.' *Proceedings of the Second Workshop on Distributed Computing*, Jadavpur University, Calcutta, December 28-30, 2000, pp 83-87.

13. M Jiang, J J Li and Y C Tay. 'Cluster Based Routing Protocol.' *IETF Draft*, August 1999.

14. C R Lin and M Gerla. 'Adaptive Clustering for Mobile Wireless Networks.' *IEEE Journal on Selected Areas in Communications*, vol 15, no 7, September 1997.

15. M Gerla and J T Tsai. 'Multicluster, Mobile Multimedia Radio Network.' *Wireless Networks*, vol 1, October 1995.

16. N H Vaidya, P Krishna, M Chatterjee and D K Pradhan. 'A Cluster-based Approach for Routing in Dynamic Networks.' *ACM Computer Communications Review*, vol 27, no 2, April 1997.

17. R Ramanthan and M Steenstrup. 'Hierarchically Organized Multihop Mobile Wireless Networks for Quality of Service Support.' *Mobile Networks and Applications*, vol 3, 1998.