DYNAMIC ADDRESS ALLOCATION PROTOCOLS FOR MOBILE AD HOC NETWORKS

A Thesis

by

PRAVEENA PATCHIPULUSU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2001

Major Subject: Computer Science

DYNAMIC ADDRESS ALLOCATION PROTOCOLS FOR MOBILE AD HOC NETWORKS

A Thesis

by

PRAVEENA PATCHIPULUSU

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Nitin H. Vaidya (Chair of Committee)	Riccardo Bettati (Member)
A. L. Narasimha Reddy (Member)	Wei Zhao (Head of Department)

August 2001

Major Subject: Computer Science

ABSTRACT

Dynamic Address Allocation Protocols for Mobile
Ad Hoc Networks. (August 2001)
Praveena Patchipulusu, B.E., Devi Ahilya Vishwa Vidyalaya;
Chair of Advisory Committee: Dr. Nitin H. Vaidya

Address allocation is an important issue in Mobile ad hoc networks. This thesis proposes solutions to assign unique IP addresses to nodes participating in Mobile ad hoc networks and evaluates the proposed solutions.

Address allocation protocols such as DHCP are centralized in nature. In Mobile ad hoc networks, due to the lack of a central server, such protocols are not very suitable. In order to truly be a network of peers, a MANET requires a distributed algorithm for address assignment. Therefore, mechanisms like DHCP are not sufficient to assign IP addresses in the MANET environment when nodes enter and leave the MANET group at their will. This thesis proposes a distributed algorithm "Dynamic Address Allocation Protocol". This protocol takes the concept of the central node allocating the IP address but the functionality of the central node is distributed among all the nodes. This protocol uses mac addresses to identify network merging. This thesis also proposes another solution in which unique identifiers are sent along with the routing messages, which help to distinguish between duplicate addresses, and thus provide a way to uniquely identify a node.

To mother, father, friends

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my advisor Dr. Vaidya for his guidance, comments and encouragement. Without his advice, enthusiasm and help, my graduate school experience would not have been so fulfilling.

I thank Dr.Bettati and Dr.Reddy for being part of my advisory committee.

I would like to thank my friends Mikin Macwan, Avinash Gupta and Shilpa Santhanam for their help during the course of work.

I would like to thank my parents for their constant support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Mobile ad hoc networks are networks composed of mobile nodes forming a temporary network without requiring the aid of any established infrastructure or centralized administration. Mobile users want to communicate in situations in which no fixed wired infrastructure is available either because it may not be economically practical or physically possible to provide the necessary infrastructure. Ad hoc networks are sometimes also called infrastructureless networks, since the mobile nodes in the network dynamically establish routing among themselves to form their own network "on the fly".

In order to deploy ad hoc networks, ad hoc routing protocols, such as DSR [1], DSDV [2], AODV [3], TORA [4], may be used. In mobile ad hoc networks, each node acts as host as well as router. For routing and other functions, the routing protocols assume that every node has a unique identifier. Therefore, address allocation is an important issue for the nodes to participate in mobile ad hoc networks.

A lot of research has been done in routing for mobile ad hoc networks, but not much has been done in the address allocation problem. In some systems, nodes in MANETs require IP addresses to be configured a priori, before they become a part of the network [5]. But in MANET, nodes should be able to enter and leave the network at their choice. So, static/manual configuration is inadequate in MANET because nodes may need to be configured with new parameters whenever they enter a new environment.

Centralized methods for allocating IP address in wired networks use DHCP state-

---

The journal model is *IEEE Transactions on Automatic Control.*

ful autoconfiguration [6],[7],[8]. But when a node in an ad hoc network wishes to obtain an IP address, it may be difficult or impossible to contact any central address allocation agency in the network. So, in ad hoc networks we need some distributed mechanism to allocate IP addresses. Some solutions have been proposed for allocating addresses in MANETs [9],[10],[11], but these solutions do not assign unique addresses in some specific conditions.

This thesis proposes a "Dynamic address allocation protocol" which assigns unique address to a node dynamically using a distributed algorithm. The proposed approach assumes that MAC addresses are unique or any other unique identifier can be used. This solution ensures assignment of unique IP addresses. This thesis also proposes "Duplicate address resolution algorithm" which detects duplicates and assigns unique IP addresses.

The remainder of the thesis is organized as follows. Chapter II describes the previous work related to this topic. Chapter III describes the proposed "Dynamic address allocation protocol" and "Duplicate address resolution algorithm". Chapter IV describes the details of "Dynamic address allocation protocol". Chapter V presents simulation results. Chapter VI gives the conclusion and possibilities for further work.

CHAPTER II

RELATED WORK

This thesis presents new protocols for allocating unique addresses to nodes in mobile ad hoc networks. This chapter discusses the related work.

A. Server-Based Schemes

A scheme for address binding, which extends DHCP [12] into new environemnts, for roaming users is discussed in [6]. This scheme assumes the node to have at least one physical interface onto the access network and each access network has at least one edge server. It only considers server based methods for registration and configuration of nodes within a single IP domain. It assumes that the network servers are themselves configured with information such as an IP address pool. Dynamic Registration and Configuration Protocol (DRCP) [13] is a lightweight dynamic configuration protocol, which heavily borrows from DHCP. This is also a server-based method, but here, DRCP server must configure its own interface using the configuration information for that subnet, which provides more flexibility and robustness than DHCP.

Dynamic configuration of IPv4 link local addresses [7] describes a method by which a host may automatically configure an interface with an IPv4 address in the 169.254/16 range that is valid for link-local communication on that interface. IPv4 link-local addresses are independent from any other IPv4 addresses that a host may have. When a host wishes to configure a link-local address, it selects an address at random, uniformly distributed in the range 169.254.1.0 to 169.254.254.255. After it has selected an address, the host must test to see if the address is already in use before beginning to use it. On a network such as Ethernet that supports ARP, this test is done using ARP probes.

Another solution is to increase the IP address space, which is described in Mobility support in IPv6 [14]. IPv6 addresses are all 128 bits long, instead of 32 bits as in IPv4. Within this huge address space, a tiny part is reserved for all current IPv4 addresses, and another tiny part is reserved for Link-Local Addresses, which are not routable but which are guaranteed to be unique on a link (LAN). Nodes on the same link can communicate with each other by using their link-local addresses. Nodes discover each other's presence, as well as each other's link-layer (MAC) addresses, by participating in the Neighbour Discovery Protocol [15] and IPv6 Stateless Address Allocation Protocol [16]. The IPv6 Neighbor Discovery protocol can be characterized as a much improved version of two IPv4 protocols, the Address Resolution Protocol (ARP) [17] and the ICMP Router Discovery Protocol [18]. IPv6 nodes also discover local routers and network prefixes by means of neighbour discovery.

IPv6 Stateless Address Autoconfiguration [16] specifies the steps a host takes in deciding how to autoconfigure its interfaces in IPv6. The autoconfiguration process includes creating a link-local address and verifying its uniqueness on a link, determining what information should be autoconfigured. This stateless mechanism allows a host to generate its own addresses using a combination of locally available information and information advertised by routers.

Dynamic Host Configuration Protocol for IPv6 enables DHCP servers to pass configuration parameters using extensions to IPv6 nodes. It offers the capability of automatic allocation of reusable network addresses and additional configuration flexibility. This protocol is a stateful version of "IPv6 Stateless Address Autoconfiguration". Both stateful and stateless autoconfiguration can be used concurrently in the same environment, leveraging the strengths of both mechanisms.

B. Schemes for Mobile Ad Hoc Networks

There are other solutions proposed particularly for mobile ad hoc networks.

1. Variable Length Addressing

Variable Length Addressing [10] describes a mechanism, in which two fields are added to the network layer packet header. These two fields are each four bits long and represent the ADDR_LEN (the number of nibbles currently being used for addressing) and HIGH_ADDR (the current highest address in use in the network). Each field requires only four bits, therefore requiring an extra byte be added to each packet transmitted.

In this mechanism, a node wishing to participate in a network, sends a join beacon. If no reply is received in a short amount of time, then the node retransmits the join beacon and waits for a longer period of time. This continues using an exponential back-off for a number of retries.

When a node already in the network hears the join beacon, this node becomes the joining node's attachment agent. This agent replies to the joining node informing it that an address request is pending and floods an address request message throughout the network requesting to use the address after the current highest known address. If any node in the network knows that the requested address is not available, they must reply to the attachment agent with a negative address request. This reply to the attachment agent can contain a recommended address. This process ends when the attachment agent floods an address request message and an appropriate period of time passes with no negative replies.

To avoid duplicate addresses, the attachment agent could have to require positive replies from all existing network nodes that the requested address is valid. Otherwise,

there would be no way to tell if a node did not reply, or had left the network. Two attachment agents at opposite sides of the network could easily allocate the same address. If duplicate addresses exist in a network routing tables may become invalid and routing algorithms may no longer guarantee loop freedom or accurate delivery.

2. DADHCP

Distributed dynamic host configuration protocol [11] presents a solution similar to that in [10]. New nodes entering a MANET request configuration information from neighbors who are already a part of network. One of these neighbours initiates a configuration allocation for the new node. The protocol requires all nodes in the network to agree upon an IP address currently not in use by the network and assign the same to the new node. But this algorithm also does not consider network partitioning and subsequent reconnection.

3. IP Address Autoconfiguration for Ad Hoc Networks

C.Perkins [9] proposed a solution in which a node in an ad hoc network may autoconfigure an IP address, which is unique throughout the connected portion of the ad hoc network. According to this specification, the node attempts to select a random address in the range 2048-65534 from 169.254/16, similar to the way that Autonet allocations are done in zeroconf [19]. Then, the node floods a Route Requests (RREQs) for that randomly selected address. If no Route Reply (RREP) for that randomly selected address is received within a timeout period, the node retries the RREQ upto RREQ_RETRIES times. If, after all retries, no RREP is still received, the node assumes that the address is not already in use, and assumes that the address can safely be taken for its own.

The specifications in [9] are only designed for use with ad hoc network protocols

that offer a mechanism for "route discovery". This mechanism also cannot detect why a node did not reply for all the RREQ_RETRIES times and risk of duplicate address allocation is still there. This mechanism also does not consider the problem of duplicate addresses when MANETs merge. Using this method, unique addresses can be assigned within a single MANET group but there is no duplicate detection when two or more MANETs merge.

CHAPTER III

PROPOSED SOLUTIONS

This thesis proposes a solution for assigning unique identifiers to the nodes participating in mobile ad hoc networks and a solution for duplicate address detection.

A. Dynamic Address Allocation Protocol

This solution provides a methodology to allocate unique IP addresses to all the participating nodes in mobile ad hoc networks. Allocation of IP addresses is done here through a distributed mechanism which provides the following capabilities:

- Network Initiate : allow a node or nodes to establish a network
- Network Join : allow nodes to join an existing network
- Network Separate : allow nodes to separate from or leave the network and form a new MANET group
- Network Partition : allow nodes to leave the network and join another already existing MANET network
- Network Combine : allow two previously disconnected MANETs to merge and form one MANET.

A new node broadcasts the join message. It receives the leader's IP address from other nodes. The new node then contacts the leader to get IP address. So, the new node gets IP address from the leader node in the network. The leader in this protocol is the node with the highest IP address. Only the leader is responsible for allocating the IP address to other nodes. The new node of the network after receiving the IP address becomes the new leader of the network.

Each node has a notion of 32-bit IP address and a unique identifier along with that. The 32-bit unique IP address is assigned according to the protocol. So, it is guaranteed that the nodes within a MANET group have unique IP addresses. Alongwith this, if two MANETs merge, there is unique identifier associated with each MANET group, which helps in detecting the merging. For this unique identifier, proposed protocol assumes that Mac addresses for each node are unique and assigns the Mac address of the first node in MANET as the unique identifier of the whole MANET group.

1. Node Initiate

A mobile ad hoc network can be initiated by any node. Whenever a new node wants to participate in a network, it will send a join beacon which specifies that it is a new node without IP address. If it does not get a reply back even after a certain amount of time, then the node retransmits the join beacon and waits a longer amount of time. This continues for a certain number of retries. If after a certain number of retries the node does not get a reply, then the node assumes that there is no MANET group existing in this area and it is the only node in the area wishing to participate. The procedure so far is similar to [10]. The node assigns an IP address, say i, for itself. Alongwith this, it also assigns itself a unique identifier which is the Mac address of this node. Now this node acts as a "leader" of this MANET.

Each node also stores the highest IP address which is the IP address of the leader. For the network initiate case, the node will store its own IP address in this field. This view of highest IP address may not be accurate but it does not affect the assigning policy of the algorithm. We will see this later in Node Join section [3.A.2]. Every node sends hello messages periodically to its neighbours so that each node is aware of its neighbours. In hello messages, a node also includes the unique identifier, which

is used in network partitioning section [3.A.4] and network merging section [3.A.5]

2. Node Join

Whenever a node wishes to participate in a network and broadcasts a join beacon and waits for a response, as we saw in section 3.A.1, if no reply is received, it assumes that it is the only node in the MANET group. However, if there are nodes already present, the following steps are taken.

The nodes, which hear the join beacon message, wait for a random period of time to avoid broadcast storm reply problem, and then send the address of leader according to their view by reversing the route in Route Request (RREQ). Even if the notion of leader of the sending nodes is not accurate, the new node will ultimately reach the correct leader following the leader links of the other nodes. Ultimately, the new node contacts the leader node in the network, which is responsible to assign IP address to the new node. The leader then assigns IP address to the new node and updates its own leader field to new IP address. This new node broadcasts the new leader address so that other nodes can update their leader pointers.

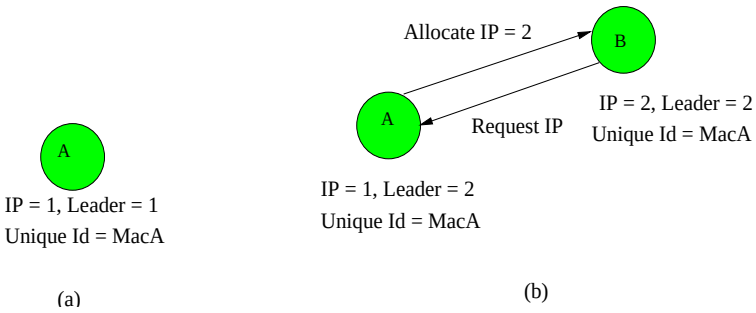

Fig. 1. Node Join with 2 Nodes

In Figure 1(a), A is the first node in this group and so has assigned itself an IP address of 1 and selected its own mac address MacA as a unique identifier and the highest address, which is the leader will also be 1. But when new node B comes in

figure 1(b), A being the highest address node, responds and assigns an IP address of 2 and unique identifier MacA to B and updates its own highest pointer to node 2 (which is B). Now, B broadcasts the highest address so that other nodes update their view of highest address. So, actually the last incoming node to the group acts as the central server assigning IP addresses.

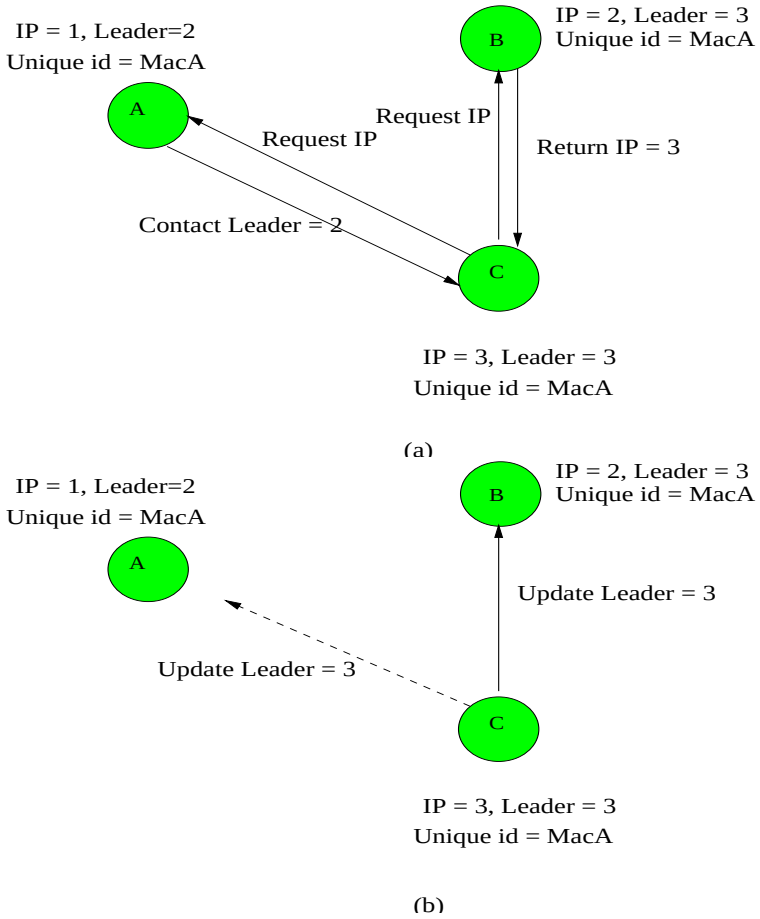

Fig. 2. Node Join with 3 Nodes

Similarly, in Figure 2(a), C is the next node entering the manet group and broadcasts a join beacon. Both A and B receive this join beacon from new node C. Node A is not the leader but has the leader's IP address. So, A sends a message containing leader's IP address to C. Node B being a leader assigns IP address to C.

Node C will get address (which will be 3) only from B and B updates its highest pointer to node C. C will broadcast updated leader IP address message saying it is the current highest node as in figure 2(b).

But, as in Figure 3(a), say A got a join beacon message from new node D before it updates the leader value and replies to new node D about its view of leader value which is B. Now, in figure 3(b), D contacts B, which further leads to C, as node B definitely has the correct information about leader node because it assigned address to C. Then ultimately, in figure 3 (c), D contacts the accurate leader C. Then, C assigns IP address to D as 4 and hence ensures assigning of unique address.

3. Node Separate

Whenever a node leaves the network willingly, it can inform the highest address node about its leaving so that its address can be reused for the new upcoming node.

But when a node is down or is out of range of this network, there is no way of detecting. The protocol proposes a notion of time-lease for address assignment. Whenever time-lease is about to expire, it can take the lease for another time period from the previous node. Each node i whose address is, say i, sends hello beacons to its neighboring nodes. Neighboring nodes in this context indicates the nodes, which have address as i-1 and i+1. If the nodes do not receive these periodic messages from their neighbors for a certain amount of time or for the time-lease period and does not receive any updating messages or join messages, the node itself can detect that node i itself is out of range of the network and now node i cannot use that address anymore.

Other nodes in the network, when they do not receive messages from the node i, but receive other messages from the network like update/join messages of address, they can wait for some more time and if the time lease of that node i expires, they

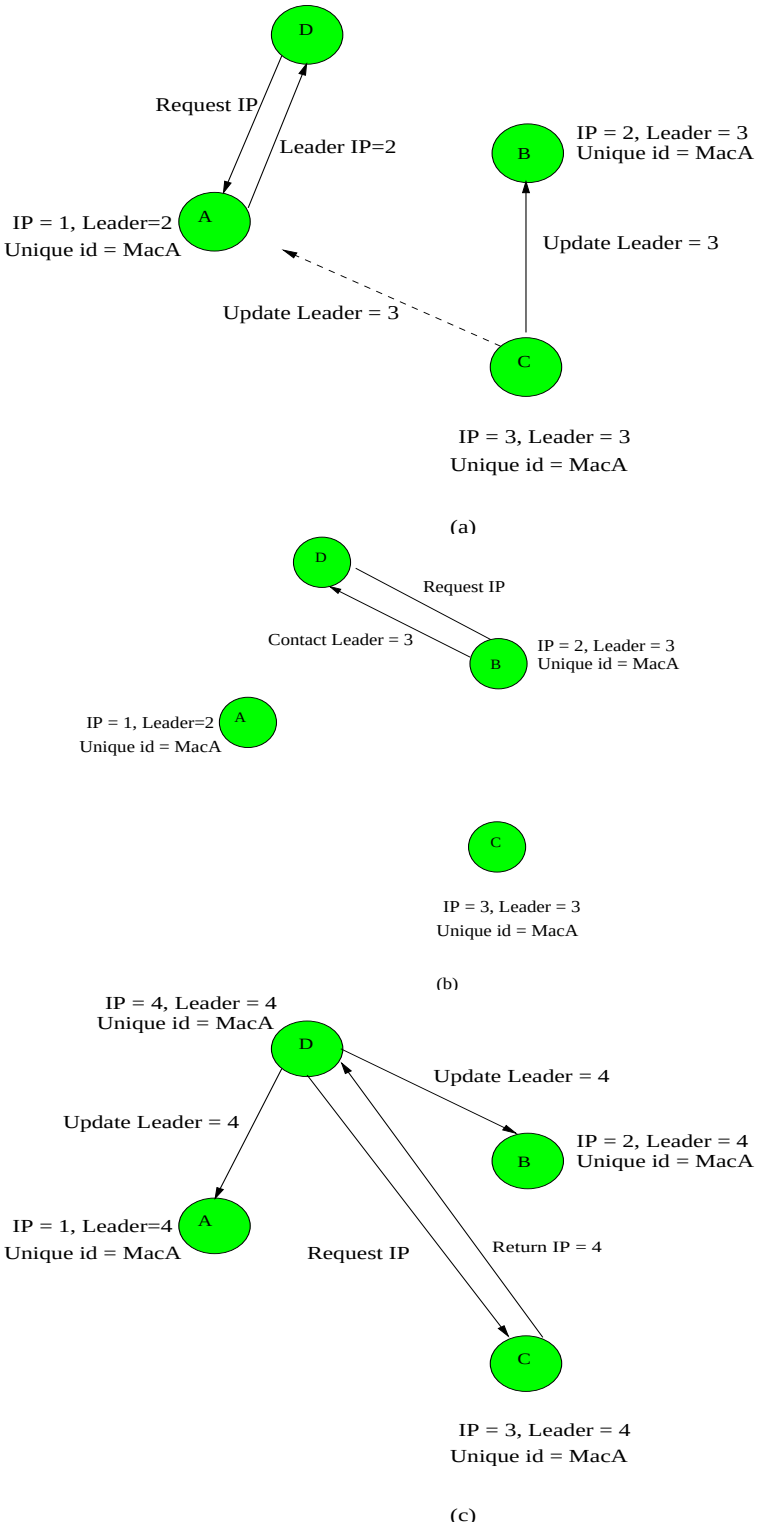

Fig. 3. Node Join with 4 Nodes

update their previous and next pointers and assume that node i is down. Whenever a node goes down and comes up, it should again start its process of getting the address.

Updation of next pointers is important because the view of leader address is important in address allocation. If the leader node h itself is out of range, the previous node h-1 will detect its absence after its time-lease and declare itself as the leader of the group. So, there is always a central node who is responsible for assigning the address.

4. Node Partition

Whenever a node i gets out of range from one network, as said previously, it detects about this situation. Now, if that node i is not in range of any other network, it will not receive any messages from any network. After its time lease expires, it will assign itself a new address i and the unique identifier as Maci, where Maci is the mac address of the node i. It assigns the leader address as i.

If the node is in out-of-range of the previous network but now is in range of new network, it can detect this situation when it receives periodic hello messages which have their unique identifier different from its own identifier. As the node is out-of-range of the previous network, it will not receive any messages from that network, so it will not receive any reply for leasing its address. So, now node i realizes that it is out of range of previous network and is in range of new network, so it resets its own address, unique identifier and leader address and broadcasts a join beacon to the new network and joins the new network.

5. Network Merge

One key challenges to mobile ad hoc networks is the resolution of addresses when two networks come into contact. To detect contact of two networks, the unique identifier

is used, which is sent alongwith hello messages.

Whenever a node alongwith receiving the hello messages of its own identifier, also receives hello messages, which has a different identifier than its own, it recognizes that there are two partitions, which are merging. It sends this message to the highest node in its partition. The partition that has less number of nodes will reassign IP addresses to its nodes.

B. Duplicate Address Resolution Algorithm

This solution proposed by Vaidya states that the IP address can be assigned by any method as described in [9], [10], [11]. The basic problem with the assignment policy arises when the addresses are not unique. So, to identify that there exists duplicate addresses, we can assign an additional unique identifier to each node which is unique. This identifier can be the node's mac address or some other unique identifier.

Now, each node stores the unique identifier alongwith the IP address in all the routing related control packets. The routing table also includes this unique identifier alongwith the IP address of the next hop/destination. So, there is one-to-one correspondence between each IP address and this unique identifier. The IP addreses are used in the data packets and the unique identifiers are included only in the routing related packets(such as link state updates).

Whenever a node A wants to send data to node B and sends the route control messages, it makes an entry in its routing table consisting of B's IP address and the corresponding unique identifier alongwith other information required for routing. Now, if A has another packet to be sent for C and C has the same IP address as that of node B, then while updating its routing table, node A realizes that duplicates addresses exist. At this point, node A informs node B and node C to reassign IP

addresses to themselves and can again call any of the above described address assignment algorithms repeatedly until unique addresses are assigned to nodes B and C. During this duplicate address resolution process, if any other messages are received by A whose destination is either B or C, node A can inform the sender about the duplicate addresses and once the duplicate address resolution is done, can inform the sender about the new addresses assigned to nodes B and C. This technique can be applied to a wide range of routing protocols; includes source routing, link state routing and distance vector protocols.

CHAPTER IV

DYNAMIC ADDRESS ALLOCATION PROTOCOL

This chapter deals with the details of the dynamic address allocation protocol. It gives the flow of program, protocol messages exchanged to assign unique addresses to nodes and the data structures used.

A. The Algorithm

The basic algorithm is the same as described in the previous section. Whenver a new node does not have an IP address, sends a broadcast join request with the IP field 0 indicating that it does not have an IP address. This message is broadcasted to the one-hop neighbours using TTL value of 1. The nodes in the vicinity hear this message and respond with their view of leader's IP address, which may not be accurate, but as explained in the previous section, will lead to the correct value. The new node does not have an IP address, so the nodes in the vicinity respond to this new node with the help of MAC address of the new node. The nodes in the vicinity, in this context, means the nodes which are one-hop from the new node. The nodes which are one-hop distance, jitter before responding with the leader's IP address value to avoid the reply-storm problem.

When the new node receives the leader's IP address, new node contacts the leader. The leader may be multiple hops far from the new node. For the message to reach the leader, any routing algorithm can be used to find the route to the leader. This message is similar to the route request message in DSR [1], with the difference that the source IP address is set to the ethernet address of the source. When the message reaches the intended destination, the destination node checks if it is correct leader.

If the destination is the correct leader, destination node sends an IP reply message which contains the new IP address for the new node, and updates its own leader field. For this message to reach the new node (which does not have an IP address), the one-hop neighbour of the new node uses the ethernet address of the new node. If the destination is not the correct leader, it sends a message which contains its view of leader's IP address.

In this proposed algorithm, the new node after receiving the IP address becomes the new leader of the network. When the new node receives the IP reply message, it updates its IP address and assigns itself as the leader of the MANET group. It broadcasts update-leader message to the MANET group informing that it is the new leader of the group.

The address which is assigned to the new node by the leader is assigned for a specific time period, as in DHCP. When this time-lease expires, the new node, which is new leader, has to extend the time lease for the IP address from the old leader. So, if the old leader does not receive message from new leader, the old leader detects that the new leader is down or leaves the network and old leader takes over the responsibilites of new leader.

Alongwith these messages, each node broadcasts hello messages to the logical neighbors. Logical neighbors of node with IP address i will be the nodes with the IP address i-1 and i+1. Each node has another field which stores the unique identifier for the MANET group. So, each hello message also contains this unique identifier. This unique identifier is used to detect network merging. These hello messages are broadcasted periodically to the logical neighbours.

These hello messages are used to detect if any node is down. If a node receives all the other messages from the MANET group but does not receive hello message from a particular logical neighbor, it detects failure of that particular node. But if a node

does not receive any messages from the MANET group, the node itself detects that it is out of range of the MANET. So, it forms a new MANET group. But if the node receives hello messages which carry different unique identifer from the node's own unique identifier, the node detects that it is entering the new MANET group. The node then resets its own IP address and sends the join message to the new MANET group to receive a new IP address.

Similarly, if a node receives hello messages from two MANET groups, it detects the merging of two MANET groups and informs the leader of its own MANET group about the merge. It also sends a message to the other group and finds the number of nodes in that network. Finding the number of nodes is not difficult as each node has the IP address of the last node entered the group, which gives the number of nodes in the network. The MANET group which has less number of nodes reassigns addresses to the nodes of its group.

The following steps are taken when a message is received at each node, the node checks for the following and takes appropriate steps.

- If the message is a join message and the node itself is the leader, the node sends IP address to the new node.
- If the message is a join message and the node is not a leader, the node sends the leader's IP address to the new node.
- If the node is new node and receives the leader's IP address, it contacts the leader.
- If the node is a new node and receives IP address, it assigns itself IP address and broadcasts a update leader message.
- For all the nodes, when they receive the update leader message, they update

their leader field.

- If the message is a hello message, each node just expires the hello timer and wait for the new hello message.

B. Protocol Messages Exchanged

The following messages are exchanged among nodes for allocating an address to a new node. The new node does not have an IP address, so other nodes should use its ethernet address to send a message to this node.

- *getIpForNode*: this message is sent by the new node. This is broadcast message querying the neighbors to either give the IP address if any of them is leader otherwise give the leader's IP address.

- *returnLeaderidToRequestor*: this message is sent by all the neighbors (except the one node which is leader if any present) to the new node (requestor) as a response to the new node's *getIpForNode* message. This message contains the leader's IP address according to the view of sending node and is unicast back to the layer 2 ethernet address of the requestor.

- *contactLeaderid*: this message is sent by the requestor to the leader id of the group to get the IP address. This message is unicast to the leader as the requestor knows the IP address of the leader now. The leader can be multiple hops from the requestor. The route to the leader can be found by any routing algorithm. This message is similar to the route request message of DSR. But here the source will put its layer 2 address instead of its IP address in the request packet. Other nodes on the way to the leader put their IP addresses in the request message packet.

- *returnIpToRequestor*: this message is sent by the leader of the group to the requestor. It just reverses the route in the request packet which it receives from the requestor with the message contactLeaderid. The new node gets the packet by the layer 2 address. This message contains the IP address of the new node. Now, the new node updates its IP address. The leader updates its leader id to the new IP address which it assigns.

- *updateLeaderid*: this message is sent by the new node after it receives the IP address from the old leader. Now, this node is the new leader of the manet group. This message is a broadcast message which contains the IP address of the new node, which is the current leader.

- *HelloPkt*: this message is sent by all the nodes to their logical neighbors periodically. This message contains the unique identifier of the manet group, which is the mac address of the first node in the group.

- *extendIPlease*: this message is sent by a node whose IP address is i to the node whose IP address is i-1 requesting it to extend the time lease on that IP address, indicating that the node is still alive and wants the IP address for more time.

- *approveIPlease*: this message is sent by the node whose IP address is i-1 to the node whose IP address is i in response to extendIPlease, approving the request of extension.

- *informNetworkMerge*: this message is sent by the node which detects the network merge to the leader of its network to which it belongs, informing that there is another network which is merging.

- *reassignIPAddress*: this message is sent by the leader of the network which is

reassigning the IP Addresses, to all the nodes in the network. This message is broadcasted by the leader to all the nodes.

C. Data Structures

The following data structures are maintained by each node in the MANET.

- *LeaderId*: each node stores some values which are required by the algorithm. Each node has a field which stores the leader's IP address, so that whenever a node receives a request for join from the new node, it replies to the new node with the value of leader's IP address, if the node itself is not the leader. If the node itself is the leader, it sends IP reply to the new node. This parameter gets updated when the new node after receiving IP address, broadcasts the new leader IP address.

- *unique_identifier*: Alongwith this, each node stores unique identifier which is unique to the MANET group. This is sent alongwith the hello messages which are periodically exchanged between nodes to check the existence of nodes.

- *no_of_retries*: each node after sending the request for IP address, waits for the response from the neighboring nodes to receive the leader's IP address or the leader to respond with the IP address for the new node. If the node does not get a response after a certain amount of time, it retries for another time and increments no_of_retries. If the no_of_retries reaches a certain number, the node initializes itself as the leader of the manet group.

- *join_reply_timer*: this timer is started by the node when it enters the MANET and sends a join message to the network. It waits for their neighboring nodes to reply with the leader's IP adress

- *hello_timer*: this timer is responsible for sending periodic hello messages between nodes in the manet group to show their existence.

- *IPlease_timer*: this timer is started by the node when the node receives a new IP address from the leader. Each new node is assigned an IP address for a certain amount of time-lease. After the time-lease expires, the new node has to again request the leader and extend the time-lease. So, when the IPlease_timer expires, the node requests for extension for time-lease.

CHAPTER V

SIMULATION DETAILS

A. Simulation Scenario and Parameters

This section describes the details of simulation scenario and parameters. Performance evaluation is performed using a modified version of ns-2 simulator [20]. ns-2 is a discrete event-driven network simulator with extensive support for simulation of TCP, routing and multicast protocols. The CMU extensions are also used for simulations. The ns-2 simulator includes a module to simulate "Dynamic Address Allocation protocol" in the routing layer. For simulations, the proposed algorithm uses DSR as the routing algorithm, although any routing algorithm can be used. The channel bandwidth is assumed to be 2 Mbps.

In the proposed simulation model, the total number of nodes considered are 20 nodes. To model the mobility of nodes, random waypoint model from CMU extensions is used. In each mobility scenario generated using this model, the 20 nodes are initially places in randomly chosen points in a 500*500 meters area. The nodes follow a randomly selected path, depending upon the speed specified. The following values are used for parameters.

- All flows are CBR traffic.
- The simulation duration is 30 seconds.
- All data packets are 512 bytes.
- Maximum packets set to 10,000.
- Packet interval rate is set to 0.0625.

- Routing protocol used is Dynamic Source Routing (DSR).
- NoofRetries - The number of retries a node makes before assigning IP address to itself, is set to 3.
- Hello_Timer - hello messages are sent periodically within 0.03 seconds.
- Join_Timer - A node after sending request for IP, waits for a join_timer period and if it does not receive a reply within this period, again transmits the request packet. This timer is set to 0.03 seconds.

The performance of the protocol is measured in terms of number of messages exchanged and latency under various network conditions.

- *Number of Messages exchanged*: Number of messages exchanged is defined as the number of messages exchanged by the protocol to assign unique IP addresses to all the nodes. The total number of messages exchanged by the protocol can be divided into two categories.

    1. The messages exchanged to assign IP addresses, *getIPforNode, returnLeaderIP, returnIPForNode*. These messages are the basic messages which are required by the protocol and which are specific to this protocol.
    2. The *hello* messages exchanged are the messages which are useful to detect the presence of neighboring nodes, to detect if some node is leaving, to detect paritioning, to detect network merging, etc. If some routing protocol already uses some kind of "Hello protocol", to detect the presence of neighbors, this protocol can be modified to meet the requirements of "Dynamic Address Allocation Protocol".

Simulations are carried out seperately to calculate total number of messages exchanged and basic messages exchanged. One part of simulation calculates the total number of messages exchanged, including both type of messages. Another part of simulations calculates the number of basic messages exchanged by the protocol. These simulations are carried out by varying number of nodes from 1 to 20 in regular intervals in a specific mobility pattern and then by keeping nodes constant at 20 and varying the mobility patterns.

- *Latency*: Latency is defined as the time taken by the protocol to assign unique IP address to a particular node. In other words, the total time taken by the protocol, from the point when the node entered and requested for IP address till the point when the protocol was successful to assign an IP address is defined as latency. In my simulations, the average amount of time taken by the nodes to receive an IP address is reported. The latency is measured by varying the number of nodes keeping other network conditions constant and then by keeping nodes constant and varying mobility patterns.

B. Simulation Results

1. Message Overhead

This section calculates the messages exchanged by the nodes to assign unique IP addresses to the nodes.

In Fig 4., the total number of messages exchanged are plotted by varying the number of nodes in the network. Speed of all the nodes is set up at 5m/s. Pause time of nodes is set to 1.0. As can be seen from the graph, as the number of nodes are increased, the number of messages exchanged also increases because now there are more nodes exchanging information. When a new node broadcasts a join message

asking for IP address, the neighboring nodes reply with the leader's IP address, so the number of messages increase. The hello messages exchanged between the nodes also increases as the number of nodes increases. So, overall effect is the increase in number of messages exchanged by increasing number of nodes. Each point on this graph is obtained by averaging over 20 simulation runs where the scenario and the number of connections in each run are changed.

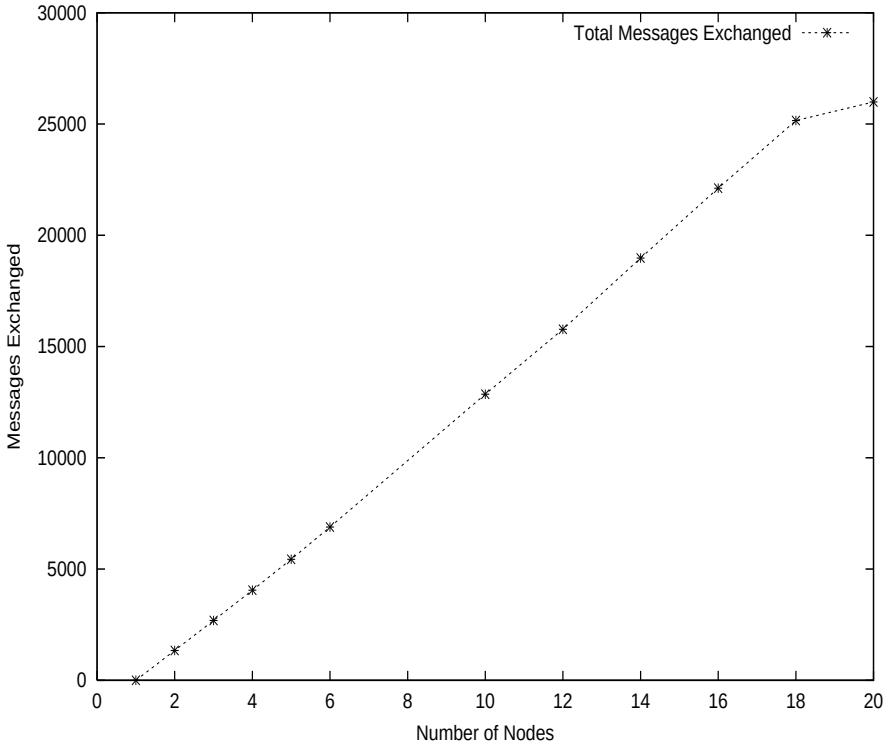

Fig. 4. Number of Nodes vs. Total Messages Exchanged

Fig 5. plots basic protocol messages exchanged as the number of nodes increases. Basic protocol messages also increase as the number of nodes increase, the reasoning being same as for the above graph (Fig 4). Here the simulation parameters are the same, number of nodes are varied from 1 to 20, speed of nodes is set at 5m/s, pause time is set to 1.0. Here as the area in which nodes can move is 500 * 500 meters there

are no partitions seen, so the addresses assigned to the nodes are unique. Each point on this graph is obtained by averaging over 20 simulation runs where the scenario and the number of connections in each run are changed.

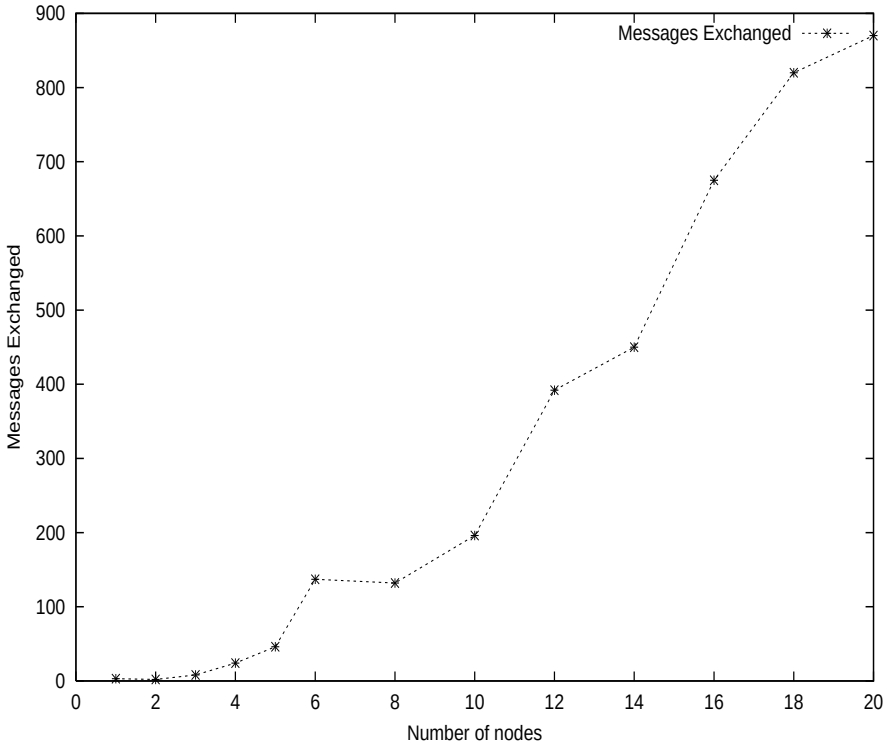

Fig. 5. Number of Nodes vs. Messages Exchanged

Fig 6 plots total number of messages exchanged by varying the speed of nodes. For this simulation, number of nodes is set at 10, pause time is set at 1.0s, number of connections in the network is set at 4. Speed is varied in the range of 2m/s to 30m/s in the intervals of 5m/s. As can be seen, the total number of messages exchanged decreases when the speed increases till 25m/s, where the total number of messages exchanged is the lowest. This is because as the nodes are moving fast, they are always close to each other, and even though if some routes fail, they are formed quickly because of fast movement before the message reaches the intended recipient and the

messages reach the destination as there is route from source to intended destination. But once the speed increases beyond a certain point, the message intended to reach the destination may not reach because the node must have moved from that position as the speed is very high. So, the messages have to be retrasmitted and takes more messages. Each point on this graph is obtained by averaging over 20 simulation runs where the scenario in each run are changed.

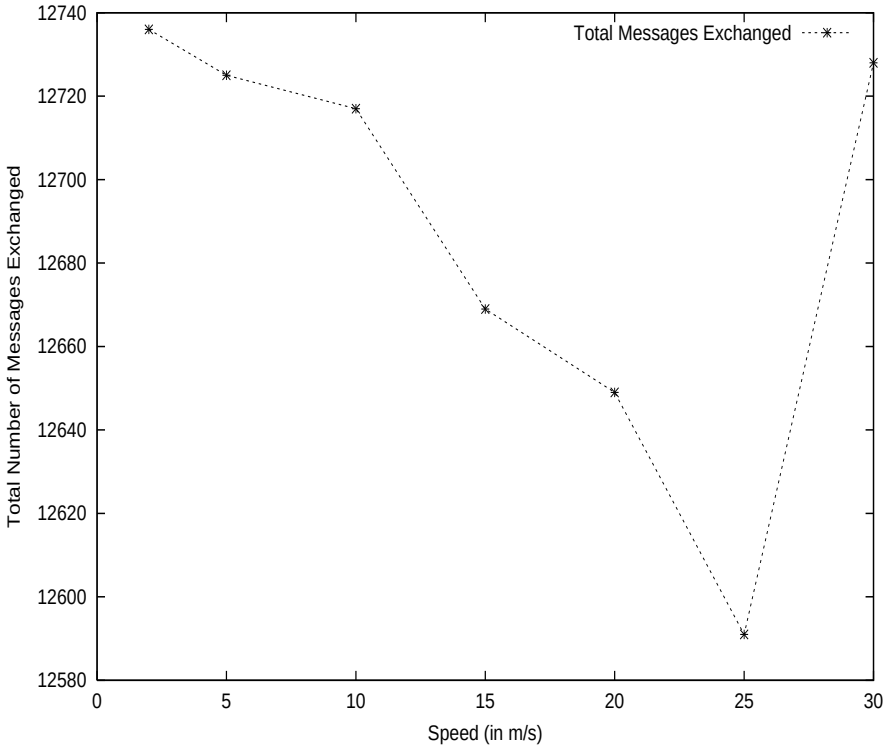

Fig. 6. Speed vs. Total Messages Exchanged

Fig 7. plots the basic prototcol messages exchanged as the speed is varied from 2m/s to 30m/s as the above graph. Number of nodes is set at 10, pause time is set at 1.0s. It shows the same results as the above graph with similar reasoning.

Fig 8. plots the total messages exchanged as the pause time is varied from 0.0

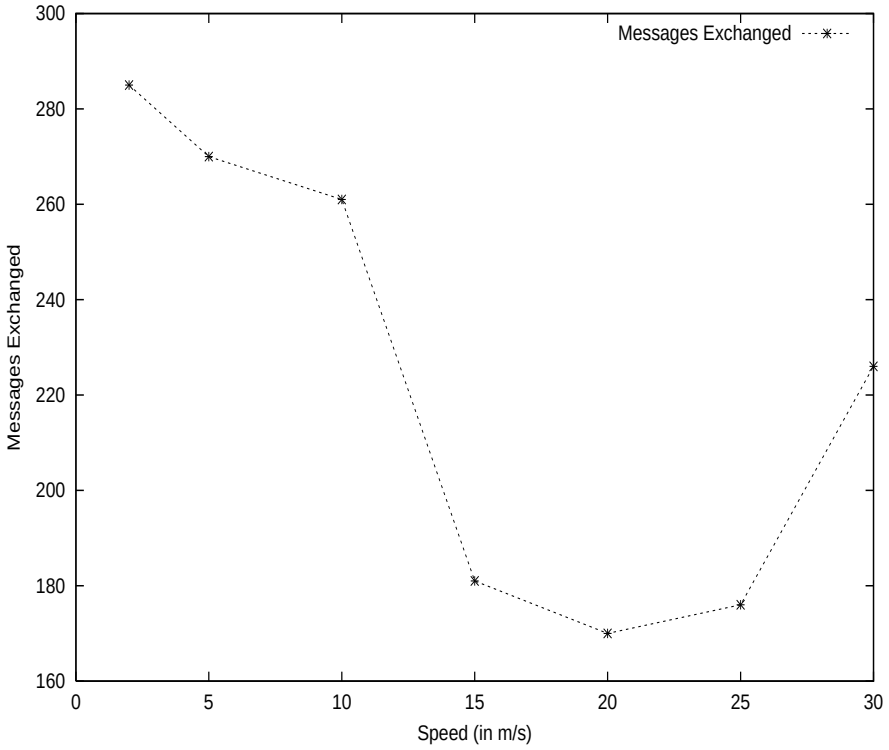

Fig. 7. Speed vs. Messages Exchanged

to 3.5 in the interval of 0.5s. Number of nodes is set to 10 and speed is set to 25m/s. As seen from the figure, total number of messages decreases in the beginning till the pause time reaches 1.5. This is because when the pause time increases from 0 to 1.5, the movement of the nodes decreases and so messages reach the intended destination as required without much retransmissions due to which the total number of messages decreases. But when the pause time increases beyond a point at 1.5, the total number of messages increases again because if the nodes are initially far from each other and there are multiple hops for a message to pass through, the sender times out and retransmits. As the pause time increases, the message takes the same time to reach as the nodes dont move for a long time and dont come closer, which increases the messages exchanged.

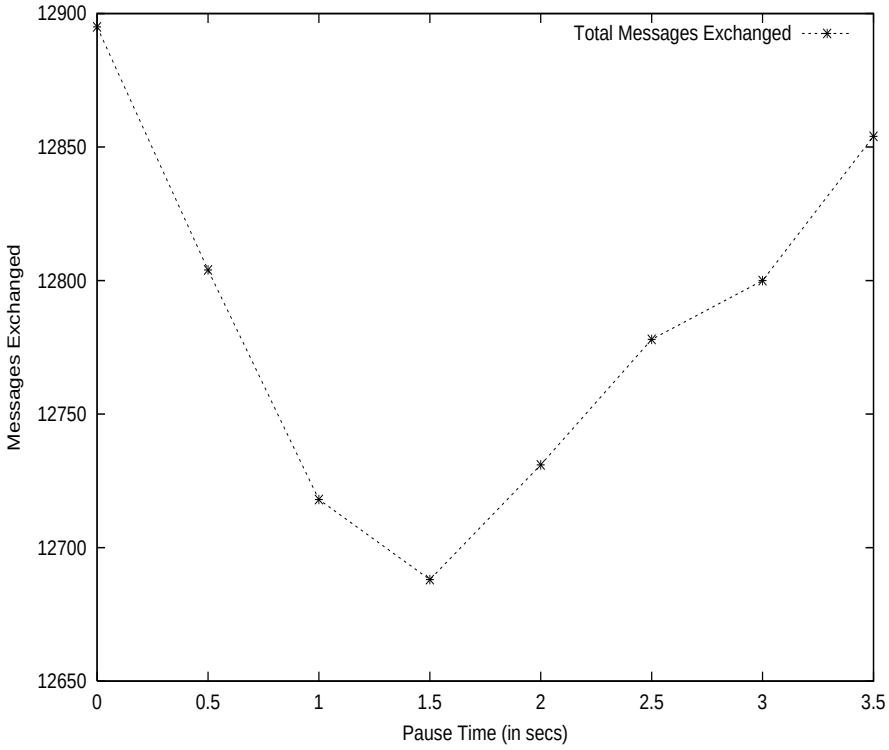

Fig. 8. Pause Time vs. Total Messages Exchanged

Fig 9. plots the basic protocol messages exchanged as the pause time is varied from 0.0 tp 3.5 in the interval of 0.5s. Number of nodes is set to 10 and speed is set to 25m/s. The graph is similar to the above graph with the same reasoning.

In a network where a large percentage of packets get delivered by the routing protocol, the number of messages exchanged by the dynamic address allocation protocol will reduce because number of messages lost will be less and there will be no duplicate messages sent due to this.

2. Latency

Latency is plotted by varying number of nodes, speed, pause time, the mobile network area in which nodes can move, and the number of connections present in the network.

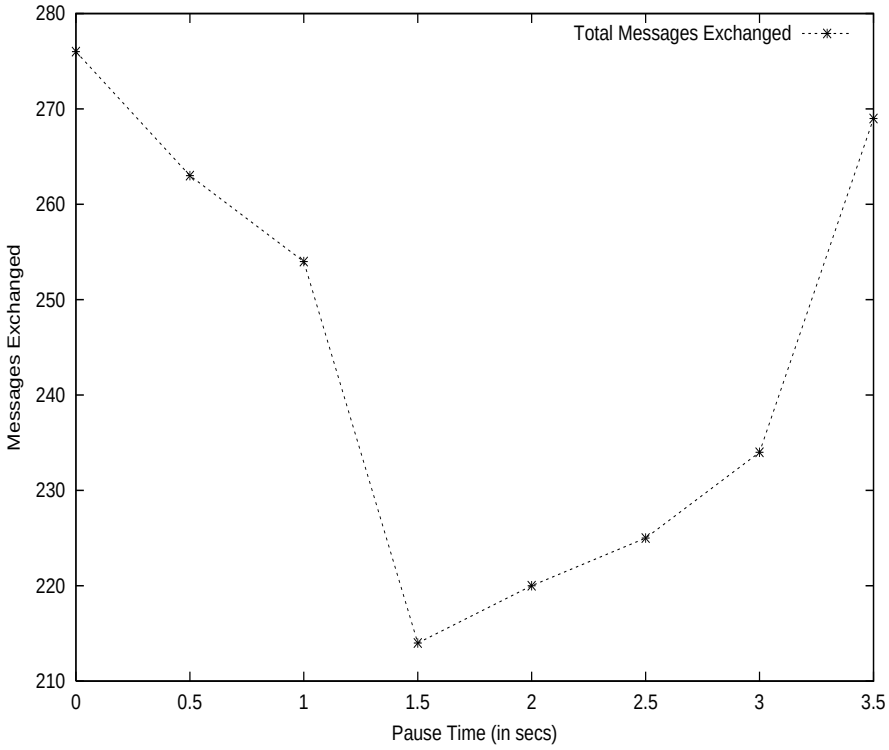

Fig. 9. Pause Time vs. Messages Exchanged

Fig 10 plots latency by varying number of nodes. Number of nodes are varied from 1 to 20. Pause time is set to 1.0 and speed is set to 25m/s. As seen from the figure, latency increases as the number of nodes increase. As the number of nodes increase, the number of messages exchanged between nodes increases and the messages get lost, due to which new nodes timeout and retransmit the messages, which increases the latency. Each point on this graph is obtained by averaging over 20 simulation runs where the scenario and the number of connections in each run are changed.

Fig 11. Latency is measured by varying the rectangular space in which nodes can move. It is varied from 1000*1000 to 3500*3500. Speed of nodes is set to 5m/s. Number of nodes are set to 10. Pause time is set to 1.0. Number of connections are set to 4. Graph shows that as the mobile network area increases, latency increases.

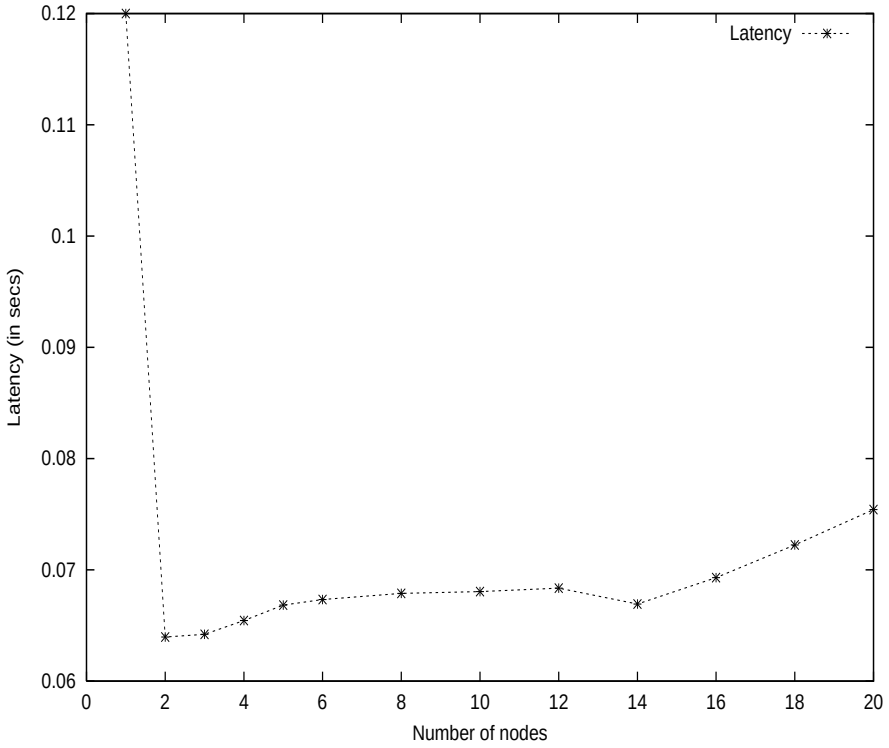

Fig. 10. Number of Nodes vs. Latency

This is because as the network area in which nodes can move increases, the nodes move far apart, and messages take more time to reach the destination as they have to travel more hops before reaching destination. This increase in latency of messages causes nodes to time out and retransmit messages, which again increases latency. The increase in network area also results in network partitions and the new node times out and retransmits until it assigns IP address to itself, which takes more time, increasing latency. And when again if nodes come closer, networks can merge, in which case, reassignment of IP addresses takes place and increases latency. As shown in Fig 12., number of partitions and corresponding merges also increase as the network area increases. Each point on this graph is obtained by averaging over 20 simulation runs where the scenario in each run is changed.

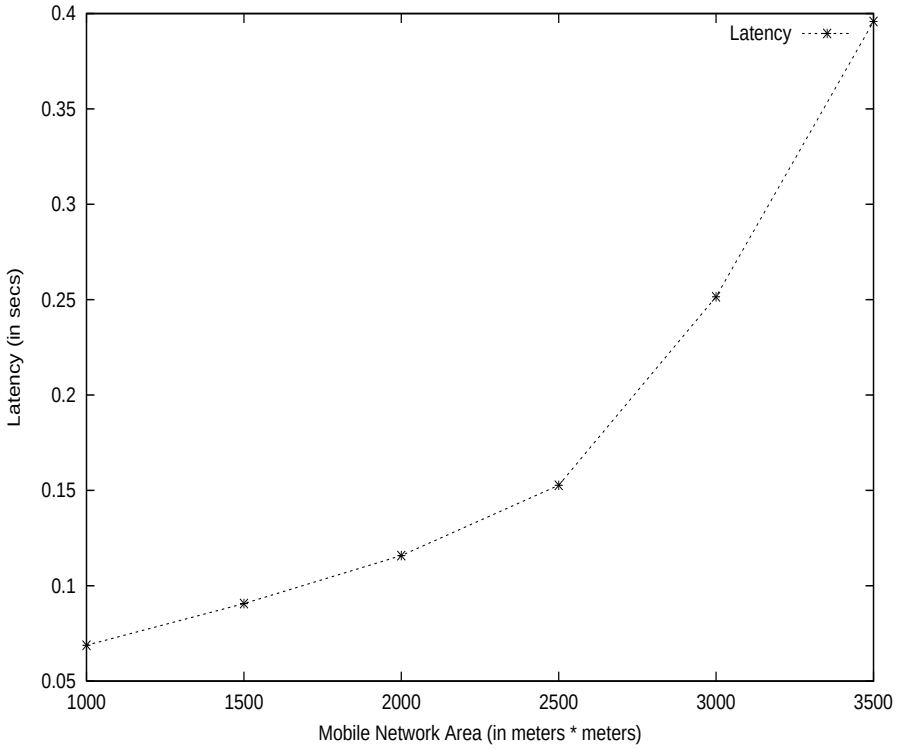

Fig. 11. Mobile Network Area vs. Latency

Fig 13 plots latency against number of connections. Here number of connections between nodes, which is responsible for network traffic, is varied from 2 to 8 in steps of 1. Speed is set to 5m/s, pause is set to 1.0 and number of nodes are set to 10. As the number of connections are increased, the number of messages exchanged between nodes also increases, due to which IP assignment messages get lost or may take more time, in which case node times out and retransmits the message. This increases latency as the number of connections are increased.

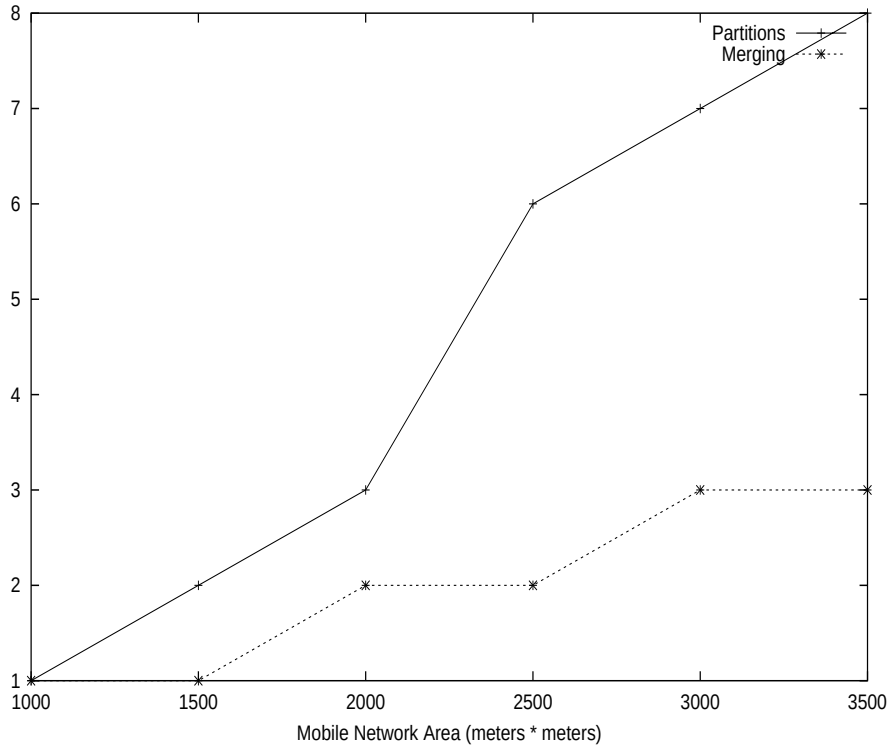

Fig. 12. Mobile Network Area vs. Network Partitions, Mobile Network Area vs. Network Merging

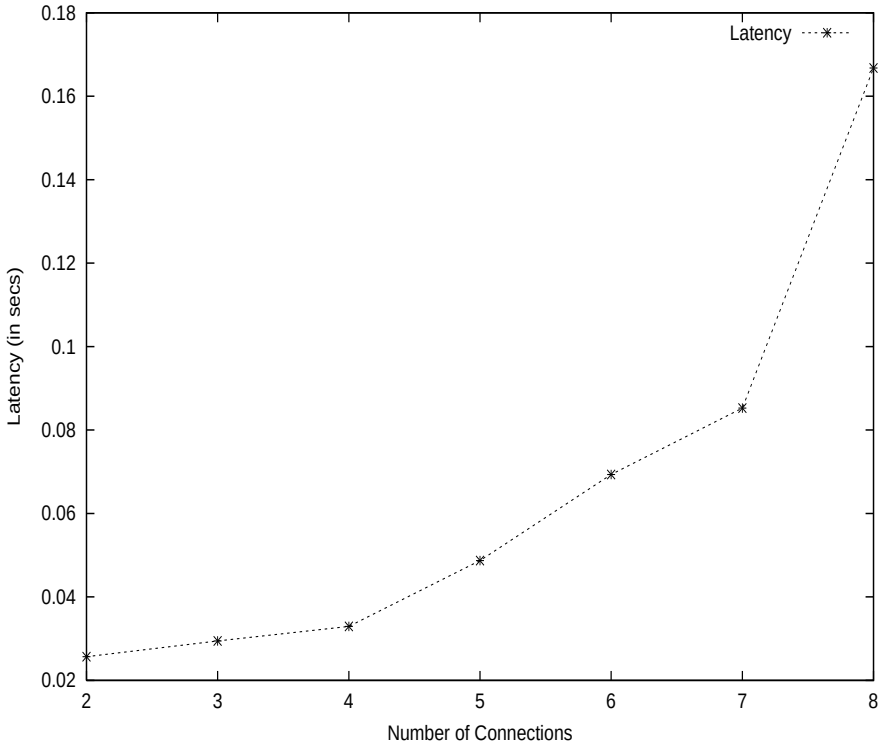

Fig. 13. Latency vs. Number of Connections

CHAPTER VI

CONCLUSION

This thesis proposes solution for assigning unique IP addresses to nodes in mobile ad hoc networks and solution for duplicate address detection. This thesis presents a distributed and dynamic address assignment protocol for IP address assignment. It guarantees uniqueness under all network conditions. The proposed solution for assigning unique IP addresses to nodes works for network partitions and network merging as well. Hence, there is no need of manual address configuration of nodes in a MANET.

Simulations show that "Dynamic Address Allocation Protocol" requires the exchange of finite number of messages. The proposed solution has a bounded latency, which means, each new node acquires IP address within a finite time. The protocol works correctly even in the presence of protocol message losses.

In the proposed protocol, there is no specifications for subnets. Future work can focus on this area. Although there is no requirement for a common prefix for IPv4 forwarding between nodes within the MANET, but from outside, the MANET must present itself to the rest of the internet as a set of one or more aggregatable IPv4 prefixes. It is possible for two MANETs to overlap, in which case the nodes common to both MANET's need to have both prefixes. If a MANET is partitioned, each connected component should be reconfigured so that the different components use different network prefixes. In order to accomplish this, some sort of subnet convergence should occur, maybe through some gateway/border router that is selected by the home/foreign agent, such that one or more logical IPv4 subnets appear to reside within the MANET when viewed from the outside.

REFERENCES

[1] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks" in *Mobile Computing*, T. Imielinski and H. Korht, editors. Dallas, TX : Kluwer Academic Publishers, 1996, pp. 153-181.

[2] C. Perkins and P.Bhagwat, "Routing over Multihop Wireless Network of Mobile Computers." in SIGCOMM'94: *Journal on Selected Areas in Communications,* Vol. 17 No. 8, pp. 1395-1414, August 1999.

[3] Charles Perkins and Elizabeth Royer, "Ad Hoc On-Demand Distance Vector Routing." in *2nd IEEE Workshop on Selected Areas in Communication,* pages 234-244, New Orleans, October 1994.

[4] 1 V. D. Park and M. Scott Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks." in *IEEE Conference on Computer Communications* (Infocom'97),1997.

[5] David A. Maltz, Josh Broch, and David B. Johnson, "Quantitative Lessons From a Full-Scale Multi-Hop Wireless Ad Hoc Network Testbed." in *Proceedings of the IEEE Wireless Communications and Networking Conference*, Chicago, IL, 8 September, 2000, pp. 134-144.

[6] Anthony McAuley, Subir Das, Shinichi Baba and Yasuro Shobatake, Requirements for Extending DHCP into New Environments. Internet Draft draft-ietf-dhc-enchance-requirements-00.txt, 8 March, 2000

[7] Stuart Cheshire, Dynamic Configuration of IPv4 L21ink-Local Addresses. Internet Draft draft-ietf-zeroconf-ipv4-linklocal-00.txt, 8 October, 2000

[8] J. Bound, M. Carney and C. Perkins, Dynamic Host Configuration Protocol for IPv6 (DHCPv6). Internet Draft draft-ietf-dhc-dhcpv6-15.txt, 5 May, 2000

[9] Charles E. Perkins, Elizabeth M. Royer and Samir R. Das, IP Address Autoconfiguration for Ad Hoc Networks. Internet Draft draft-ietf-manet-autoconf-00.txt, 10 July, 2000

[10] Jeff Boleng, *"Efficient Network Layer Addressing for Mobile Ad Hoc Networks"*, Colorado School of Mines, Golden, CO. April 3, 2000

[11] Sanket Nesargi and Ravi Prakash, "DADHCP : Distributed Dynamic Configuration of Hosts in a Mobile Ad Hoc Network", University of Texas at Dallas, TX. Work in Progress, May 2001.

[12] R. Droms, "Dynamic Host Configuration Protocol." Network Working Group - RFC 2131, March 1997.

[13] Anthony McAuley, Subir Das, Sunil Madhani, Shinichi Baba and Yasuro Shobatake, Dynamic Registration and Configuration Protocol (DRCP). Internet Draft draft-itsumo-drcp-01.txt, 14 July, 2000

[14] Charles E. Perkins and David B. Johnson, "Mobility Support in IPv6." in *Proceedings of The Second Annual International Conference on Mobile Computing and Networking - Mobicom'96* November,1996.

[15] Thomas Narten, Erik Nordmark, and William Allen Simpson, Neighbor Discovery for IP version 6 (IPv6). Internet Request for Comments RFC 1970, August 1996.

[16] S. Thomas and T. Narten, "IPv6 Stateless Address Autoconfiguration." Internet Request for Comments RFC 2462, December 1998.

[17] David C.Plummer, "An Ethernet Address Allocation Resolution Protocol" Internet Request for Comments RFC 826, November 1982.

[18] Stephen E. Deering, ICMP Router Discovery Messages. Internet Request for Comments RFC 1256, September 1991.

[19] M. Hattig, Zeroconf Requiremenets. Internet Draft draft-ietf-zeroconf-reqts-05.txt, 18 September, 2000.

[20] K. Fall and K. Varadhan, "ns notes and documentation", Technical Report, VINT Project, University of California, Berkeley and Lawrence Berkeley National Laboratory(LBNL), 1997.

VITA

Praveena Patchipulusu received her Bachelor of Engineering Degree from Shri Vaishnav Institute of Technology and Science, Devi Ahilya Vishwa Vidyalaya, Indore, India in June 1999. She joined the graduate program in Computer Science at Texas A&M University, in September 1999. Her research has been in mobile computing and dynamic address allocation in mobile ad hoc networks. Her address is Department of Computer Science, Texas A&M University, College Station, TX 77843-3112.

The typist for this thesis was Praveena Patchipulusu.