Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.1387 – 1392

RESEARCH ARTICLE

Host Node Based Collision Free Address Assignment in Ad hoc Networks

Ms. Sumisha Samuel[1], Mr. Avanish Kumar Singh[2]

[1]Department of Computer Science and Engineering
Nehru College of Engineering and Research Center, Pampady, Thrissur, Kerala

[2]Department of Computer Science and Engineering
Nehru College of Engineering and Research Center, Pampady, Thrissur, Kerala

[1] sumishasamuel@gmail.com; [2] avnpt@rediffmail.com

Abstract— In a mobile Ad hoc network all nodes will be moving from time to time. In such a dynamic environment the address assignment is a key issue. In this paper we propose a host node based addressing technique used for assigning network addresses to the nodes in a network. Here the addresses are stored in address filter and the filter will be stored by the leader of the network. Furthermore the leader handles all joining node procedures, partition merging procedures and leaving node procedures.

Keywords— Mobile Ad hoc network; Leader; Nodes; Partition merging

I. INTRODUCTION

In Mobile ad hoc network there is no infrastructure or there is no node like server to assign the addresses for the nodes to communicate each other. There should be a unique address. But in the dynamic environment it is difficult to maintain a unique address. So we need some method to assign collision free addresses. The MANETs main application lies in the field of sensing, disaster recovery etc.

The centralized mechanisms like DHCP, NAT can't be used in the case of MANETs due to their dynamic nature. In dynamic host configuration protocol or DHCP there will be a DHCP server and when a new node comes to join the network it asks for the DHCP server to allocate an address to it. Then the DHCP server will give an address to the joining node. But this method can't be used in ad hoc network since there is no node acting as a server. In network address translation mechanism a single address is used for a number of devices in the same end. So this method is also not possible in the case of MANETs. In this paper we propose a host node based addressing protocol. Here the host node will be referred to as the leader node. The leader

node will be storing the currently allocated addresses in the address filter. The hash of the filter which is called partition identifier will be used to recognize the partitions. Furthermore, the leader node will be handling all address requests further. The leader node also deals with the different situations like partition merging and leaving node procedures. Since the address filter is stored by the leader node alone the storage requirements can be simplified.

The remainder of the paper is structured as follows. The related works are given in section II. The proposed method is given in section III and section IV concludes the paper.

II. RELATED WORKS

In the Duplicate Address Detection (DAD) scheme the nodes randomly select an IP address from the network prefix defined. Later this, the node floods the address request message which indicates the chosen address. If any other node is using the same address as chosen by the new node it sends address reply message indicating the duplication of the address. Then the joining node selects another address and the same flooding procedure is repeated. This method is the simplest method used for dynamic addressing. But it has some limitations. The method is considered and suits only in the initialization state. MANET is a dynamic network. So this method can cause increase in the control overhead. In this method the network partition and merging events are not considered [1].

In the MANETconf protocol, each and every node maintains two sets, the set of IP addresses already assigned and the set of IP addresses pending assignment. When a new node wishes to connect to the network, it floods a message to its neighbors to choose an agent. The agent selects an available IP that is neither assigned nor pending assignment, adds it to the set of IP addresses pending assignment, floods the IP to all other nodes, and starts a timer. If a recipient node finds this IP in one of its sets, it gives a negative reply. Otherwise it adds it to its set of IP addresses pending assignment list and sends a positive reply. If all nodes reply, and all replies are in favour, the agent assigns this IP to the new node, adds the IP to its assigned set, and floods this information in the MANET so all nodes add this IP to their sets of assigned IPs. If at least one reply is not in favor, the agent selects another IP and the procedure is repeated. If at least one node does not reply, the agent sends the message again to them [2].

In the Prophet method, the authors designed a function that produces a sequence of IPs. The initial state of the function is known as the seed. Different seeds produce different sequences. This protocol is as follows: The initial node in the MANET chooses a random number as its IP and uses a random state value or a default state value as the seed for its function. When a new node wants to join the MANET, it chooses a host to get a free IP. The host uses the function to generate an IP which is sent to the new node with the agent state. The new node uses the host state as the seed for its function. The host then updates its state accordingly. The advantages of this method are low latency time and low communications overhead. This method has some disadvantages like the function used cannot avoid address conflicts as each node produces a sequence in the same domain and two or more of the same numbers can be generated from the multiple sequences [3].

In the Weak DAD approach a unique key is used by every node along with the IP address. This key is generated at the starting phase. The node also sends the unique key in addition to the IP address. Each node in the network maintains the key along with the IP address in the routing table. When a node receives the message it first checks the IP address is present in its table or not. If it is not present then it assumes that the IP is unallocated and can be assigned. If the IP address is present in the table then it checks the key of the node. If the key is dissimilar the node concludes that address duplication has occurred. This method is more suitable than DAD method but it has the main drawback of dependency on routing protocol. The new modified address is used which is a combination of IP address and the unique key [4].

The IETF Zeroconf working group proposed a method where IPv6 addresses are assigned based on MAC address of the machine to the nodes. The MAC address is 48 bit long and IPV6 address is 128 bit long. This method suits well for the addresses IPV6 addresses. But most of the nodes still use IPV4 addresses and in these addresses the no of bits are smaller than the MAC address. The technique used is hashing the MAC address to fit in the address suffix. But this also method also makes use of random choice of address and does not guarantee a collision free address allocation in the network to advertise the address [5].

In Filter based addressing protocol method the addresses are stored in the filters and the filter will be stored by all nodes. A node can receive the HELLO message from any node. Partition merging procedure can be handled by any nodes in the network. Also a node can inform to any node in that network and leave the network [6].

This method aims to reduce the storage requirements and reduce the address allocation delay. Since there is a leader, the address requests without confusion can be send only to the leader node. Since the leader only stores the currently allocated addresses the storage requirements can be reduced. The partition identifier, which is used to identify the partitions is found by taking the hash of the filter. The filter stores the current set of addresses. When the stored set of addresses changes the partition identifier also changes. So the partition identifier can be used to identify the partitions.

III. HOST NODE BASED ADDRESS ASSIGNMENT

The proposed method aims to assign network address to the nodes in a dynamic ad hoc network. This method can be used in any procedures like joining node events, partition merging events and leaving node events. Here we are using a type of filter based approach where the filter stores the current set of addresses. Filter will be stored only by the leader nodes and it will be handled by the leader nodes. The leader node will manage the different events in the network. Since the filter is stored at the leader node alone the storage requirements can be reduced. The filter signature which is the hash function of the filter is used to identify the partitions. As the addresses in the filter changes the filter signature also changes.

A. Procedures of Host Node Based Approach

1) System Initialization: This is the initialization stage of the MANET. There are two situations which can happen during a MANET initialization: gradual and abrupt initialization. In gradual initialization each nodes join the network one after the other with some delay in between. In abrupt initialization the nodes comes to join the network at the same moment. The Filter Based Addressing protocol is appropriate for both gradual and abrupt initializations.

The host node based address assignment method uses two messages namely Hello and AREQ. Hello message is used for a node to show its current association status and partition identifier. AREQ message is used to show the message that previously available address is allocated now. The AREQ message has got an identifier number to distinguish between the AREQ messages generated by different nodes but with the same addresses. In the initialization phase at times there may exist a network or there may not exist a network. In the case when a network exists the node behaves as a joining node else if there is no network exist the node behaves as an initiator node. Then the initiator node behaves as the leader of the node. If the node is an initiator node, then the node selects an address based on the network prefixes and it creates an empty address filter. So it will be acting as the leader and this node alone stores the address filter. The leader node then floods the network by AREQ message. After that the leader node begins to broadcast the Hello message.

After the initialization all the nodes have selected unique addresses and all nodes addresses will be updated in the address filter and the leader node alone stores the address filter.

2) Joining Node and Partition Merging Management: After the initialization phase leader node broadcasts Hello message. The Hello message has the association status and partition identifier. So when a node gets a Hello message it can identify whether it is a partition merging event or not. Hello message can also be received by a joining node. And the joining node can join the network. The joining node method is described in the figure below.

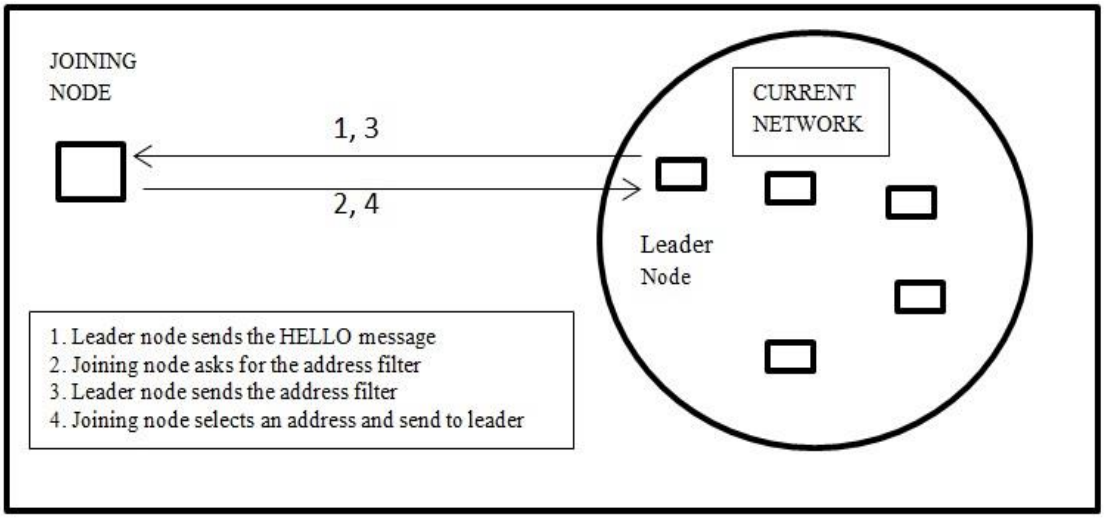

Fig. 1 Joining Node procedure

When a node turns on, it listens for receiving a Hello message from a leader node. If it receives a Hello message it can make sure that a network already exists and it will be joining node. Then the joining node requests for the address filter to the leader node. When the leader node obtains the address filter message it checks the *I* bit. If the *I* bit is 1 it is a node joining procedure else if the *I* bit is 0 it will be a partition merging process. Then the leader node will send the address filter to the joining node by making the *R* bit to 1. The *R* bit 1 indicates that the Address filter is an answer to the previous request. When the joining node gets the AF it chooses a random address which is not used. Then the joining node will send an AREQ message to the leader node to show that the address is chosen. When the leader node receives this AREQ each nodes updates its address filter and partition identifier is updated based on the current addresses in the address filter.
The partition merging events are shown in the figure below.

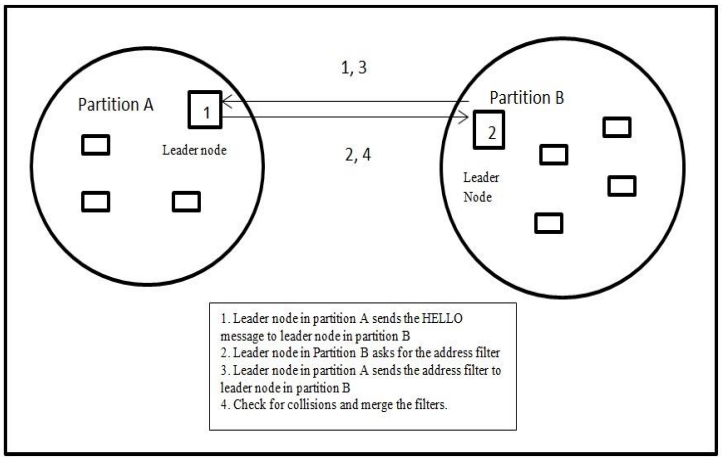

Fig. 2 Partition merging procedure

Nodes in different partitions may select addresses based on their partitions only. So sometimes in different partitions there may be identical address. This may cause address collisions when the two partitions join. The two partitions join when they come together. Such a situation is identified when a leader node gets a Hello message with a different partition identifier. Then the leader node will detect that it is a partition merging event. The two partitions will be having two different address filters and two different set of addresses and sometimes there may cause address collisions.

So when receiving the Hello message the leader node in the other partition will request for the address filter to the leader node in the other partition. Then the other leader node will sent the address filter to the requested leader node. The requested

leader node then stores this received address filter and sends its address filter to the other leader. Then both the leader nodes in the two partitions will flood the network with the partition message. After getting the partition message the leader nodes will check the M bit, where the M bit denotes the lowest priority partition. Then the lowest priority partition will check whether there occurred any address collisions while joining the two address filters. If there occurred any address collision in case, then the lowest priority partition will take a new address and floods the network with an AREQ message to the other leader node. Later the two address filters are merged and thus creating a new address filter and the partition identifier of the address filter is updated.

Now there are two leaders in that partition. So we have to select a leader for storing the addresses. So for that the network goes through leader election process. In the leader election process, every node in the partition has the right to vote. Every node votes to the node whom they want to be the leader. The node which gets the maximum number of votes will become the leader of that partition and the leader node will alone stores the address filter. Also the leader node will deal with the joining events, partition merging events and leaving events.

3) Leaving Node Management: When a node leaves a network its address should be detached from the filter. So that its address can be used by some other nodes in the network in future. If not this address cannot be used by any nodes in future and this may result in the address scarcity in future. In order to avoid the scarcity of address when a node leaves the network its address is deleted from the filter.

When a node wants to leave the network it should inform the leader node to remove its address from the address filter. Then the leader node removes that nodes address from the address filter so that address can be used by some other nodes in future.

In some case the leader node may leave the network. In that case the leader node should select a new leader and give the address filter to the new leader node. Then the old leader can leave the network.

If the node abruptly shuts down it may not notify the network. In such situations we make the idea that the address filters should be continuously updated. So during the updation, if the node gets sudden shut down we can eliminate its address from the filter. A node can leave the network during any of the procedures like joining node procedure, partition merging procedure. In order to avoid difficulties use timers to update the address filters.

4) Leader election management: In this phase a new leader is elected when wanted. So every node can vote to a node whom they wants to be the leader. A node can only vote to a node in the same partition only. Finally all the votes are counted and the node with maximum number of votes is made as the leader of the node. And the leader will alone store the address filter of the partition and leader node will alone send the Hello message to other nodes. The leader node handles the joining node management, partition merging events and leaving node management.

Fig. 3 shows the System Architecture, which shows the procedures during joining node, partition merging events and node departures.

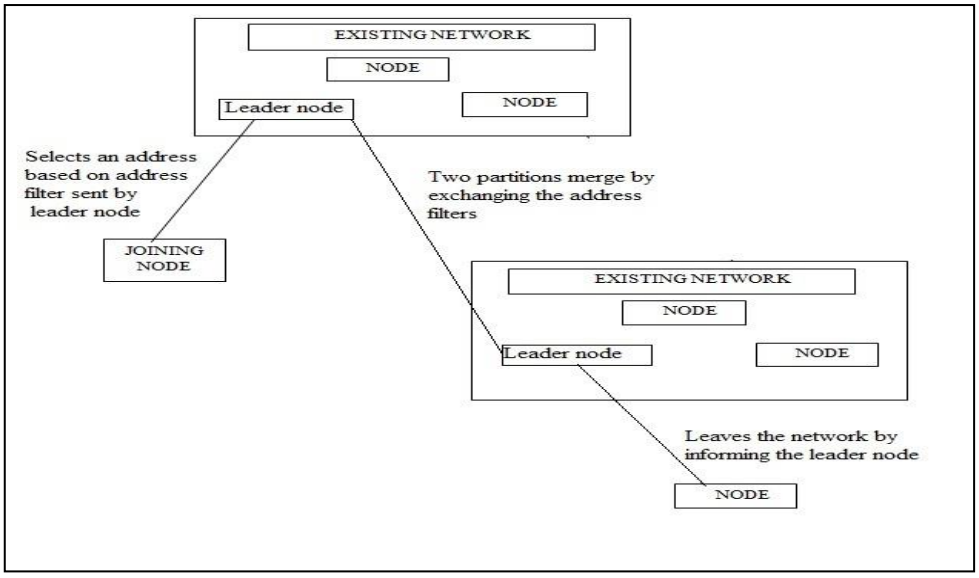

Fig. 3 System Architecture

IV. RESULTS AND DISCUSSION

The graph below shows the storage requirements needed in the proposed method that is the leader method and in the previous filter based addressing method. We consider node 1, node 2 and node 3 in a partition and node 4 and node 5 in a partition. We also consider node 1 and node 4 are leaders in that partition. In filter based addressing method each node has to store the address filter which has the number of nodes equal to the number of nodes in that partition which increases the storage requirements. But in leader based proposed method only leaders have to store the address filter and hence the storage requirements can be reduced. The results are shown in Figure 4.

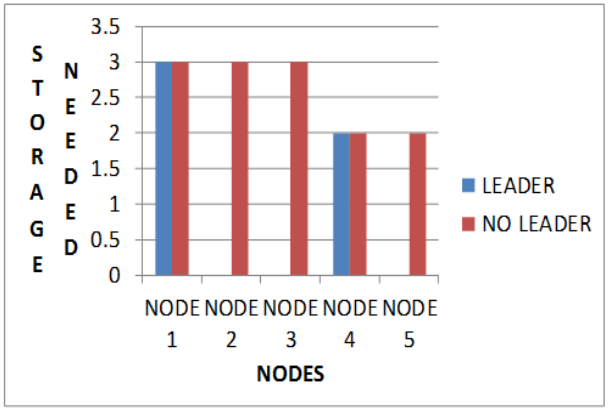

Fig. 4 Storage Requirement analysis

V. CONCLUSION

In this paper we proposed a host node based addressing method used for assigning addresses to the nodes in a MANET. Here we are making a node as a leader and the leader node alone stores the address filter. The leader node deals with the joining node, partition merging events and leaving node management. Here the addresses are stored in address filters by the leader node alone, which reduce the storage requirements and reduce the time needed for allocating address. The hash of the filter is used for identifying the partitions. This hash function helps in partition merging events.

REFERENCES

[1] C. E. Perkins, E. M. Royers, and S. R. Das, "IP address autoconfiguration for ad hoc networks," Internet draft, 2000.

[2] S. Nesargi and R. Prakash, "MANETconf: Configuration of hosts in a mobile ad hoc network," in Proc. 21st Annu. IEEE INFOCOM, Jun. 2002, vol. 2, pp. 1059–1068.

[3] H. Zhou, L. Ni, and M. Mutka, "Prophet address allocation for large scale MANETs," in Proc. 22nd Annu. IEEE INFOCOM, Mar. 2003, vol. 2, pp. 1304–1311.

[4] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in Proc. 3rd ACM MobiHoc, 2002, pp. 206–216.

[5] S. Thomson and T. Narten, "IPv6 stateless address autoconfiguration," RFC 2462, 1998.

[6] Fernandes N.C, Moreira M.D and Duarte M.B, "An efficient and robust addressing protocol for node autoconfiguration in ad hoc networks," in IEEE/ACM transactions, 2012 July.

