ISSN:0975-9646

Efficient Node Address Auto configuration in MANET

Mr.**Vikas Subhashrao Shinde**
Department of Computer Engineering
DEOGIRI INSTITUTE OF ENGINEERING AND MANAGEMENT STUDIES,
City: AURANGABAD. Country: Maharashtra

Mr. **K.Vishal Reddy**
Department of Computer Engineering
DEOGIRI INSTITUTE OF ENGINEERING AND MANAGEMENT STUDIES,
City: AURANGABAD. Country: Maharashtra

Abstract— **Mobile ad hoc networks (MANET) is used for many distributed network, the lack of a centralized administration makes these networks attractive for several distributed applications such as sensing, Internet access to deprived communities and disaster recovering. Mobility feature of the Ad hoc Network produce many problems in the network, due to this feature ad hoc network does not maintain any infrastructure, it also address, affects address assignment of the nodes in network. In this paper, we propose an efficient node address auto configuration protocol that automatically configures a network by assigning unique IP addresses to all nodes with a very low overhead and minimal cost. Evenly distributed Duplicate-IP address Detection Servers are used to ensure the uniqueness of an IP address during IP address assignment session. In contrast to some other solutions, the proposed protocol does not exhibit any problems pertaining to leader election or centralized server-based solutions. Furthermore, grid based hierarchy is used for efficient geographic forwarding as well as for selecting Duplicate-IP address Detection Servers. Through simulation results we demonstrate scalability, robustness, low latency, fault tolerance and some other important aspects of our protocol.**

Keywords— Ad hoc network, FAP Duplicate address detection (DAD), Duplicate-IP detection server (DDS), IP address auto configuration, MANET.

I. INTRODUCTION

Address auto configuration is one of the fundamental issues in MANET. A node must need some form of identity before participating in any sort of communication. So each host in a MANET needs to be uniquely addressed so that the packets can be relayed hop-by-hop and delivered ultimately to the desired destination. Moreover, nodes in the MANET are free to move and organize themselves in an arbitrary fashion. Therefore any fixed infrastructure based solution for assigning identity (i.e. IP address) is not directly applicable to MANET. Under this infrastructure less and sporadic nature of the mobile nodes, several protocols of address auto configuration in the MANET have been proposed. Although some of these protocols perform decently in sparse and small networks, but exhibit poor performance (e.g., single point of failure, storage limitation, large protocol overhead and so on) when the network is either dense or very large.

In this paper, we propose and analyze an efficient Approach called Filter-based Addressing Protocol (FAP) [2]. The proposed protocol maintains a distributed database stored in filters containing the currently allocated addresses in a compact fashion. We consider both the Bloom filter and a proposed filter called Sequence filter to design a filter-based protocol that assures both the univocal address configuration of the nodes joining the network and the detection of address collisions, Address Filter merging partitions. Our filter-based approach simplifies the univocal address allocation and the detection of address collisions because every node can easily check whether an address is already assigned or not. We also propose to use the hash of this filter as a partition identifier, providing an important feature for an easy detection of network partitions. Hence, we introduce the filters to store the allocated addresses without incurring in high storage overhead. The filters are distributed maintained by exchanging the hash of the filters among neighbors. This allows nodes to detect with a small control overhead neighbors using different filters, which could cause address collisions. Hence, our proposal is a robust addressing scheme because it guarantees that all nodes share the same allocated list.

Many existing works in the address assignment has the overhead and it does not gives the proper solution to the network partition merge. The best filter for FAP depends on network characteristics such as the estimated number of nodes in the network and the number of available addresses. It also depends on the false-positive and false-negative rates of the filter. Bloom filters do not present false negatives, which mean that a membership test of an element that was inserted into the filter is always positive.

II. RELATED WORK

Address auto configuration proposals that do not store the list of allocated addresses are typically based on a distributed protocol called Duplicate Address Detection (DAD) [3].The lack of servers hinders the use of centralized addressing schemes in ad hoc networks. In simple distributed addressing schemes, however, it is hard to avoid duplicated addresses because a random choice of an address by each node would result in a high collision probability, as demonstrated by the birthday paradox [4].

In this protocol, every joining node randomly chooses an address and floods the network with an Address Request message (AREQ) for a number of times to guarantee that all nodes receive the new allocated address. If the randomly chosen address is already allocated to another node, this node advertises the duplication to the joining node sending an Address Reply message (AREP). When the joining node receives an AREP, it randomly chooses another address and repeats the flooding process. Otherwise, it allocates the chosen address. This proposal does not take into account network partitions and is not suitable for ad hoc networks.

Other proposals use routing information to work around the addressing problem. Weak DAD [5], for instance, routes packets correctly even if there is an address collision. In this protocol, every node is identified by its address and a key. DAD is executed on the 1-hop neighborhood, and collisions with the other nodes are identified by information from the routing protocol. If some nodes choose the same address and key, the collision is not detected. Moreover, Weak DAD depends on modifying the routing protocols.

Prophet [6] allocates addresses based on a pseudo-random function with high entropy. The first node in the network, called prophet, chooses a seed for a random sequence and assigns addresses to any joining node that contacts it. The joining nodes start to assign addresses to other nodes from different points of the random sequence, constructing an address assignment tree. Prophet does not flood the network and, as a consequence generates a low control load. The protocol, requires an address range much larger than the previous protocols to support the same number of nodes in the network. Moreover, it depends on the quality of the pseudo-random generator to avoid duplicated addresses. Therefore, it needs a mechanism, like DAD, to detect duplicated addresses, which increases the protocol complexity and eliminates the advantage of a low control message overhead.

Our proposal aims to reduce the control load and to improve partition merging detections without requiring high storage capacity. These objectives are achieved through small filters and an accurate distributed mechanism to update the states in nodes. Furthermore, we propose the use of the filter signature (i.e. a hash of the filter) as a partition identifier instead of random numbers. The filter signature represents the set of all the nodes within the partition. Therefore, if the set of assigned addresses changes, the filter signature also changes. Actually, when using random numbers to identify the partition instead of hash of the filter, the identifier does not change with the set of assigned addresses. Therefore, filter signatures improves the ability to correctly detect and merge partitions.

III FAP

The proposed system proposes the protocol called FAP and it reduces Collision of address in the ad hoc network. FAP uses a distributed compact filter to represent the current set of allocated addresses. Every node has filtered to simplify frequent node joining events it also reduce the control over head in the address collisions. Signature of the Filter is important feature in the proposed system. So that it can easily detected that network merging events. In which address conflicts may occurs. For easy detection of the collision of address in the network it uses the filter signature the proposed system proposes the use of two different filters, depending on the scenario the Bloom filter, which is based on hash functions and the Sequence filter proposed in this proposed system, which compresses data based on the address sequence.

A. Bloom Filters

Another natural way to represent a set is to use hashing [7]. Each item of the set can be hashed into (-) *(*log *n)* bits and a (sorted) list of hash values then represent the set. This approach yields very small error probabilities. For example, using 2 log2 *n* bits per set element, the probability that two distinct elements obtain the same hash value is 1/*n*2. Hence the probability that any element not in the set matches some hash value in the set is at most *n*/*n*2 = 1/*n* by the standard union bound.

Bloom filters can be interpreted as a natural generalization of hashing that allows more interesting tradeoffs between the number of bits used per set element and the probability of false positives. (Indeed, a Bloom filter with just one hash function is equivalent to hashing.) Bloom filters yield a constant false positive probability even if a constant number of bits are used per set element. For example, when m = 8*n*, the false positive probability is just over 0.02. For most theoretical analyses, this tradeoff is not interesting, using hashing yields an asymptotically vanishing probability of error with only (-) (log n) bits per element. Bloom filters have therefore received little attention in the theoretical community. In contrast, for practical applications the price of a constant false positive probability may well be worthwhile to reduce the necessary space.

B. Sequence Filter

The other filter structure that we propose is called Sequence filter and it stores and compacts addresses based on the sequence of addresses.

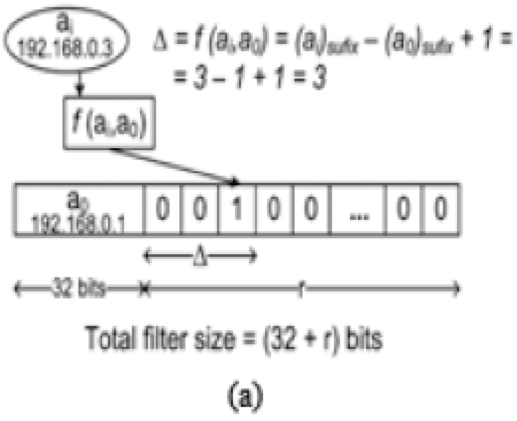

a_i
192.168.0.3
Δ = f (a_i,a_0) = (a_i)_suffix − (a_0)_suffix + 1 =
= 3 − 1 + 1 = 3
f (a_i,a_0)
a_0
192.168.0.1
0 0 1 0 0 ... 0 0
Δ
32 bits
r
Total filter size = (32 + r) bits
(a)

Fig.1. Insertion procedure of the address element

ai = 192.168.0.3 in the filters used with FAP. For the sequence filter, the address range, whose size is, goes from *a0* = 192.168.0.1 to *ar*-1 = 192.168.0.254 with a 192.168.0.0/24 sub network. (a) Sequence filter, assuming an address range of *r*=254 addresses.

This filter is created by the concatenation of the first address of the address sequence, which we call initial element (a0), with an r-bit vector, where is the address range size. In this filter, each address suffix is represented by one bit, indexed by which gives the distance between the initial element suffix (*a0suffix*) and the current element suffix (*aisuffix*). If a bit is in 1, then the address with the given suffix is considered as inserted into the filter, otherwise, the bit in 0 indicates that the address does not belong to the filter. Therefore, there are neither false positives nor false negatives in the Sequence filter because each available address is deterministically represented by its respective bit. The Sequence filter and the procedure to insert an element into the filter are illustrated in Fig. 1(a).

IV. MODULE DESCRIPTION

1) *Network Initialization:*

There are two kinds of initializations in the networks.

Abrupt initialization: joining of the nodes at the same time is called abrupt Initialization.

Gradual initialization: joining of the node Address Filter some interval and is considered as the gradual initialization.

Initially a node waits to join or try to join in the network for that it listens to the medium for a particular period (Tl). If the node does not receive the hello message with in the listening period it will act as the initiator node. The Initiator node starts the network alone or with other initiator nodes. Otherwise it acts as the joining node with the network already exists. Hello messages used in the initialization for a node to advertise its current association status and partition identifier, which is signature of the filter of each node it contains. From fig. 2.AREQ message is used to indicate that previously available address is now allocated. Each AREQ has an identifier number, which is used to differentiate AREQ messages generated by different nodes, but with the same address.

An initiator randomly chooses an address and it also creates an empty filter and starts the network initialization phase. Address Filter that the node floods the AREQ messages Nf times in the network. If there is other imitator node in the network that also send the AREQ floods messages Nf times to increase the reception of the AREQ messages by all the nodes present in the network. It is for a node randomly chooses the address. Address Filter particular time of waiting period the node does not waits for the AREQ message the node leaves the initialization phase, insert the address in its filter, the address received.

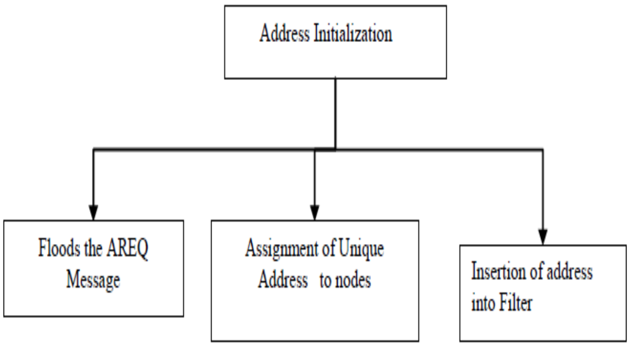

Address Initialization
Floods the AREQ Message
Assignment of Unique Address to nodes
Insertion of address into Filter

Fig.2. Network initialization of FAP

By the AREQ messages from each node and then the node starts to send the Hello messages with filter signature which is the hash value of the address filter. The signature plays the important role in the partition events. If the initiator node receives the same address with different identifier. The node finds there is the address collision. In this situation the node wait for particular time and choose another available address. It is to be continued until each node allocates the unique address to it. During the wait period it receives the many AREQ messages and check for the address collision. Therefore, Address Filter the node knows a more complete list of allocated address, which decreases the probability of choosing a used address. Hence, the period decreases the probability of collisions and, consequently reduces network control load.

2) *Node Ingress (or) Joining of Node:*

During the node joining the Host node checks the messages whether for the joining procedure or for partition procedure in Fig. 3. After the initialization the node ask for send the Hello message and after the Hello message send by the host node, the node sends the Address Filter message AF now the host node checks for the I bit is set to be 1 or 0, it is indicate whether the messages for joining procedure or the partition procedure. If the message came from a joining node then, the host node answers the request with another AF with bit set to 1, indicating that the Address Filter is an answer to a previous filter request. When the joining node receives the Address request reply message, it stores the address filter, chooses a random available address and floods the network with an AREQ to allocate the new address. When the other nodes receive the AREQ they insert the new address in their filters and update their filter signatures with the hash of the updated filter.

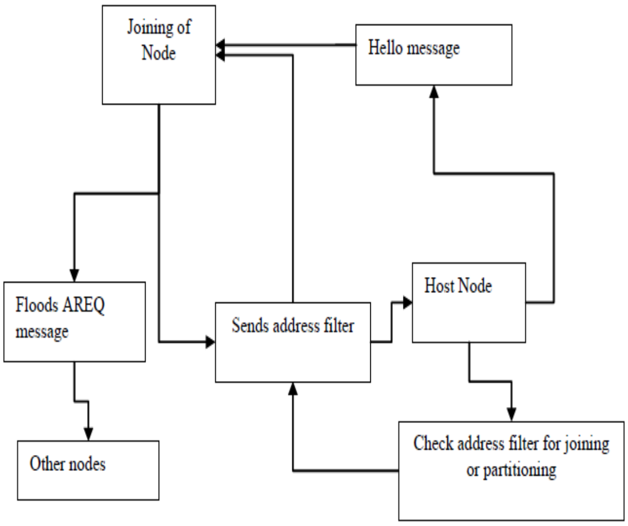


Fig.3. Node Ingress (or) Joining of Node of FAP

3) Partition Merge Events:

Merging events are also detected based on Hello and AF messages. Nodes in different partitions choose their address based only on the set of addresses of their partition. Hence, nodes in different partitions can select the same address, which may cause collisions after the partitions merged. If Address filter received from the node in that I bit indicates 0 that is partition to be done in Fig.4. The filter signature of the different partition differ in the signature, from that it is to identified that node contain the different group of address. In this both node distribute filter of its two partitions, each node on the lowest priority partition must check whether its address is on the other partition filter to detect collisions. If there is a collision, the node randomly chooses an available address in both filters and floods the network with an AREQ to allocate the new address. If the node receives an AREQ with the same address that it has chosen, but with a different sequence number, it chooses another address because another node has also chosen the same address.

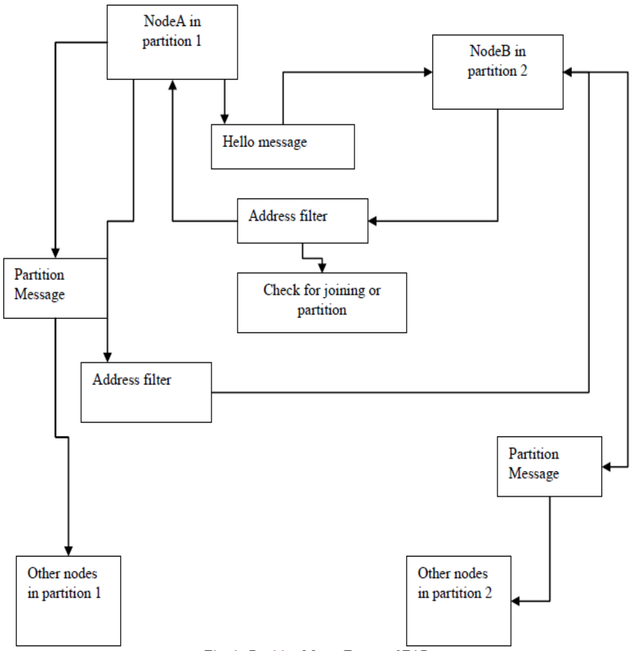


Fig.4. Partition Merge Events of FAP

Finally, all the nodes merge the other partition filter with its own filter, insert the addresses received in the AREQs into the new filter and update the filter signature.

4) Node Departure:

When node leaves the network and it floods the notification message in the network to remove the address from the address filter to perform the proper shutdown. The departure of the node is indicated by the fraction of the filter. So each time every node verifies that its filter fraction bit to check or to know the departure of node. Therefore, every node verifies this fraction in their address filters every time the filter is updated. If this fraction reaches a threshold that indicates that the filter is full or almost full, all the nodes reset their address filters and returns to the network initialization.

V. Experimental Results

Performance of the protocol is evaluated using java.netbin simulation software.

A] It shows the node details.

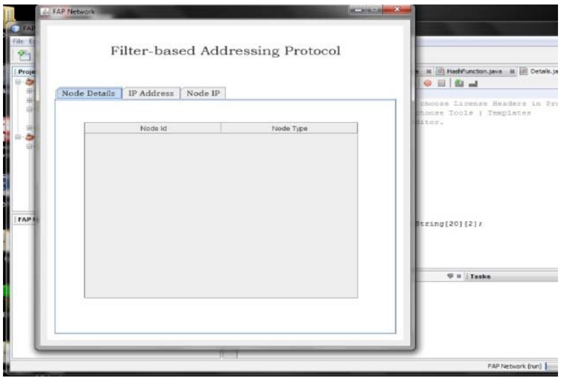
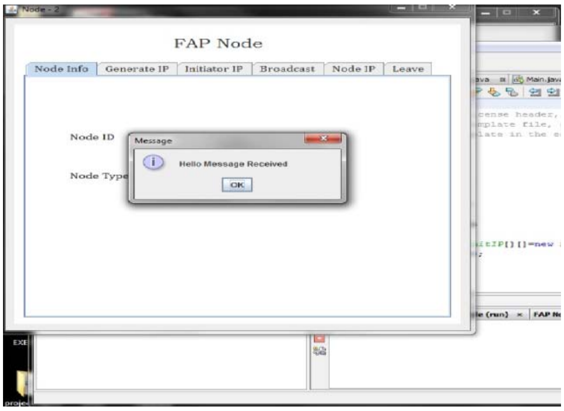


Fig.5. Nodes in the network ready to connect

In this step first you have to ready the network for connecting the node in FAP. Then send the hello message.

B] It shows the node information details.

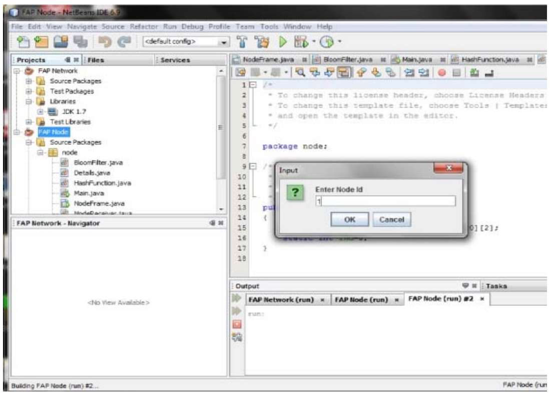


Hello Message Received

Fig. 6. Node information

In this step network will connected to the FAP node. Node gives information about message. In this step the hello message will be received.

C] It shows the node information details.



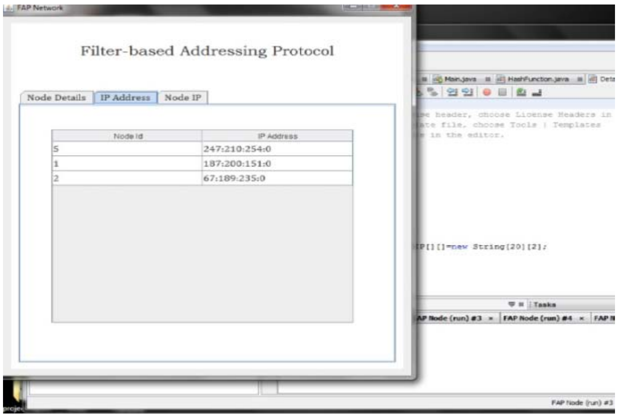
```
/*
 * To change this license header, choose License Headers
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package node;
```

Enter Node Id: 1

Fig. 7. Node information

In this step network will connected to the FAP node. Select FAP node. Entering node id one by one in network.

D] It shows the IP Address information details.

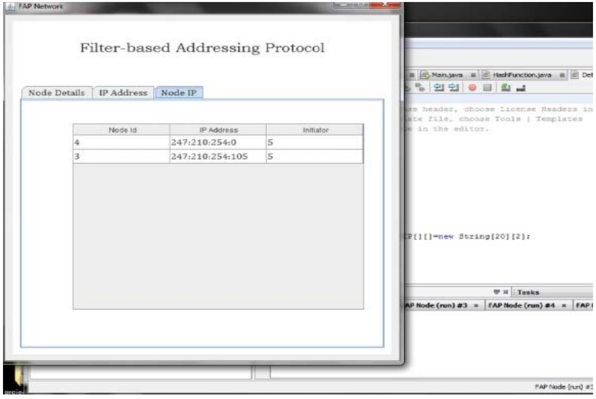


Node Id	IP Address
5	247:210:254:0
1	187:200:151:0
2	67:189:235:0

Fig.8. IP Address information

In this step network will connected to the FAP node. Select FAP node. Entering node id one by one in network. Then entering node id assigns the specific IP address. Each node id assigns separate IP address.

E] It shows the node IP Address information details.



Node Id	IP Address	Initiator
4	247:210:254:0	5
3	247:210:254:105	5

Fig.9. Nnode IP information

In this step network will connected to the FAP node. Select FAP node. Then entering one by one node id. Each node id assigns separate IP address.finally selects the node ip.

The impact of the network size, the network density, and the number of transmissions of flooding messages in abrupt network initialization is evaluated. Proposed protocol suffers a greater influence on the control load than the existing protocol. We first analyze the impact of one node joining the network. A rectangular space with nodes distributed in grid is considered. The control load after the last node joins the network and the required delay to obtain an address is measured. Simulation results reveal that proposed protocol resolves all the address collisions and also reduces the control traffic.

VI CONCLUSION

The proposed system uses the key idea is to use address filters to avoid address collisions, reduce the control load and decrease the address allocation delay. Proposed FAP avoid the collision of the address in partition merge event. It handles the join and leaves of the nodes properly. The proposed system reduces the control load. FAP provides the smaller delays in the partition merging events and node joining event. Compared to the existing work. This is achieved because FAP is able to detect all merging events and also because FAP is robust to message losses. FAP initialization Procedure is simple and efficient.

The FAP process may be improved in future by adding other techniques and other parameter to be considered to enhance the proposed approach to provide better results in delay and reduction of the control load.

REFERENCES

[1] M. Günes and J. Reibel, “An IP address configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-hoc Networks,” Proc. of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services, September 2002.

[2] N. C. Fernandes, M. D. Moreira, and O. C. M. B. Duarte, “An efficient filter-based addressing protocol for autoconfiguration of mobile ad hoc networks,” in *Proc. 28th IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 2464–2472

[3] B. Parno, A. Perrig, and V. Gligor, “Distributed detection of node replication attacks in sensor networks,” in Proc. IEEE Symp. Security Privacy, May 2005, pp. 49–63.

[4] C. E. Perkins, E. M. Royers, and S. R. Das, “IP address autoconfigura- tion for ad hoc networks,” Internet draft, 2000.

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

[6] 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.

[7] Broder and M. Mitzenmacher, “Network applications of Bloom filters: A survey,” *Internet Math.*, vol. 1, pp. 485–509, 2002