INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Node Auto Configuration in MANETs Using Filter Based Addressing Protocol

Rakesh Patil M. S.
Department of Computer Science and Engineering, JNNCE, Shimoga, Karnataka, India
Dr. Prabhudeva S.
Professor, Department of Computer Science and Engineering
JNNCE, Shimoga, Karnataka, India

Abstract:
MANETs are the wireless media, and it's infrastructure is changing and dynamic in nature, where we cannot assure that every mobile node will be connected at a given time neither predict the topology or size of the network. On the other hand, as soon as a node joins a MANET, it becomes part of the routing mechanism to exchange messages and performs actively routing tasks, hence there is not a sense of subneting like in wired networks. For this reasons, IP address assignment is a challenge in MANETs, In order to solve this problem many protocols proposed but they increases the control load of the network. Another important issue is frequency partition in the network due to fading channels. Proposed system FAP gives the solutions to these problems. This achieves low control overhead and low latency, resolving all address collisions even in network partition and merging events when compared to other system and also it reduces the rate of address collision and control messages required to allocate unique address to the nodes in the network.

Keywords*: Ad hoc network, FAP, AREP, AF*

1. Introduction

Address assignment is a main challenge in ad hoc networks due to dynamic nature of nodes that makes not fixed infrastructure network. Autonomous or self addressing protocols require a distributed and self-managed mechanism to avoid address collisions in a dynamic network with fading channels, frequent partitions, and joining/leaving nodes. The proposed system proposes the FAP to node's address configuration. On the other hand, ad hoc network protocols have to deal with limited resources, such as bandwidth and energy, the shared medium, and high bit error rates due to the properties of the wireless channel [1].

This paper gives a robust and efficient approach called Filter-based Addressing Protocol (FAP). The proposed protocol allocates unique addresses according to the current set of addresses assigned to network nodes. This set of addresses is represented in a compacted fashion using filters, which assure the unique configuration of the joining nodes in the network and the detection of possible address collisions after the merging of several network partitions. Filter contains all the set of addresses which are allocated and non allocated and they are maintained in compact manner. With the filters, any node can check if an address is available before trying to allocate it. Also, a node can check if the hash of its filter is the same of the hash of the filter of its neighbors, to identify network partitions. 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.

2. Related Work

The absence of servers does not suites the use of centralized addressing schemes in ad hoc networks. In distributed addressing schemes, however, it is hard to avoid duplicated addresses to different nodes because a random choice of an address by each node would result in a high collision probability in address allocation, as demonstrated by the birthday paradox [4].

Stateless proposals to autoconfigure IP addresses in ad hoc networks are typically based on a distributed protocol called Duplicate Address Detection (DAD) [3]. 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 of address to the joining node by 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, however, does not take into account network partitions and is not suitable for ad hoc networks.

Other proposals uses routing information to work around the addressing problem. Weak DAD [5], for instance, routes packets correctly even if there is an address collision that means same address is allocated to two or different nodes. In this protocol, every node is identified by its own address and a key of that node. 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, however, the collision is not detected and packet is routed to that node also. Moreover, Weak DAD depends on modifying the routing protocols.

Prophet [6] allocates addresses based on a pseudo-random function with high entropy. The first or initial node in the network, called prophet, chooses a seed for a random sequence and take responsibility of assigning addresses to any joining node that contacts it. Then that joined 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, so it generates a low control load. The protocol, however, needs an address range where it is 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 take off the advantage of a low control overhead.

Our proposal aims to reduce the address collision probability, delay for address allocation and the control overhead 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, 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, filter signatures improves the ability to correctly detect and merge partitions.

3. Problem Statement

A crucial and usually unaddressed issue of ad hoc networks is the frequent network partitions. Network partitions, caused by node mobility, fading channels, and nodes joining and leaving the network, can disrupt the distributed network control. Network initialization is another challenging issue because of the lack of servers in the network.

As other wireless networks, ad hoc nodes also need a unique network address to enable multi hop routing and full connectivity. Address assignment in ad hoc networks, however, is even more challenging due to the self-organized nature of these environments. Centralized mechanisms, such as the Dynamic Host Configuration Protocol (DHCP) or the Network Address Translation (NAT), conflict with the distributed nature of ad hoc networks and do not address network partitioning and merging

4. Proposed Solution

The proposed solution is Filter-based Addressing Protocol (FAP). The proposed protocol maintains a distributed database stored in filters containing the currently allocated addresses in a compact fashion. There are two types of filters, 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 after merging partitions. The hash of this filter as a partition identifier, providing an important feature for an easy detection of network partitions. Hence, here introduce the filters to store the allocated addresses without incurring in high storage overhead.

For easy detection of the collision of address in the network it uses the filter signature. The use of two different filters, depending on the scenario: the Bloom filter, which is based on hash functions, and the Sequence filter, which compresses data based on the address sequence.

4.1. Bloom Filters

Bloom filters are a data structure with high compression capacity, used on many applications. While Bloom filter is suitable for an extensive address space and a small number of allocated addresses The common way to represent a set of addresses 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 represents the set. This approach yields very small error probabilities. For example, using $2\log_2 n$ bits per set element, the probability that two distinct elements obtain the same hash value is $1/n^2$. Hence the probability of any element that may 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 known as a natural generalization of hashing that makes more interesting tradeoffs between the number of bits used per set element and the probability of false positives. Bloom filters gives 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 received little attention in the theoretical community. In contrast, for practical applications the price of a constant false positive probability may well be enough to reduce the necessary address space.

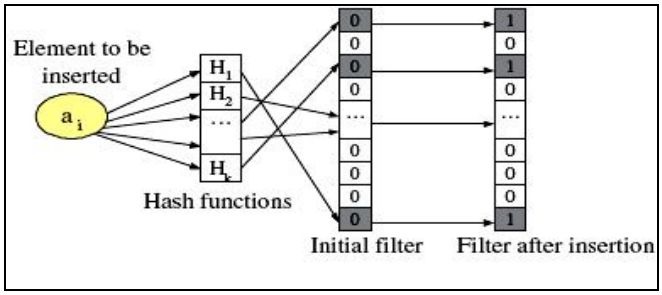

Figure 1: Insertion of elements in the filter

Fig.1. shows the insertion of elements into the filter through hash function. Through the hash functions addresses of all the nodes in the network are inserted in the filter where filter is present in each node of the network.

4.2. Sequence Filters

The another filter which has the structure to store and compact addresses based on the sequence of the addresses called Sequence filter. In this filter, each address suffix is represented address suffix, as shown on Fig.2. Therefore, there is no false-positive or false-negative in the Sequence filter. The procedure to insert an element on the filter is depicted on Fig.3.

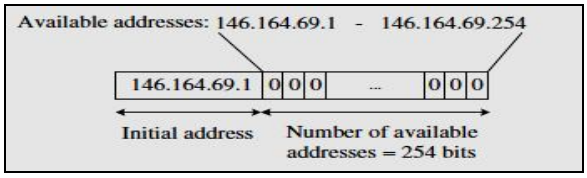

Figure 2: The Sequence filter

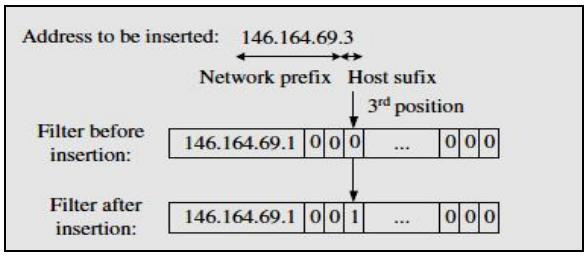

Figure 3: Insertion of elements in the Sequence filter

While Bloom filter is suitable for an extensive address space and a small number of allocated addresses but the Sequence filter is adequate to a narrow address space but supports large number of address spaces.

5. System Architecture

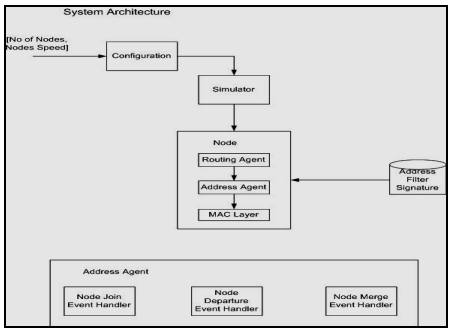

Figure 4: System Architecture

Fig.4 shows the system architecture of FAP protocol. Following are the modules by node. The system architecture diagram explains the various modules.

- Routing Agent.
- Address Agent
- MAC Layer.

- Filter Signature

The proposed model auto configures the network address. The filter signature is present at every node modules. Filter Signature is the hash of the address filter, as a partition identifier. The filter signature is an important feature for easily detecting network merging events, in which address conflicts may occur.
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 paper, which compresses data based on the address sequence.
MAC Layer, which addresses an hardware-based scheme.
Address agent helps to node joining, node departing and node merging event handler mechanisms where these mechanisms are carried out in following manner. They are as follows;

5.1. Network Initialization

There are two kinds of network initializations can happen in the networks. They are, abrupt and gradual. In abrupt initialization, where joining of the nodes at the same time in large manner takes place. But in gradual initialization, joining of the node at different intervals of time happens.
Initially a node has to wait to join or try to join in the network for that it listen 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 will start the network alone or with some other initiator nodes. Otherwise it acts as the joining node with the network which is already exists. Fig.5 shows the how operation sequentially happens in network initialization period. An initiator node randomly chooses an address, own partition number and it also creates an empty filter and starts the network initialization phase where Address agent helps to node joining, node departing and node merging event handler mechanisms.

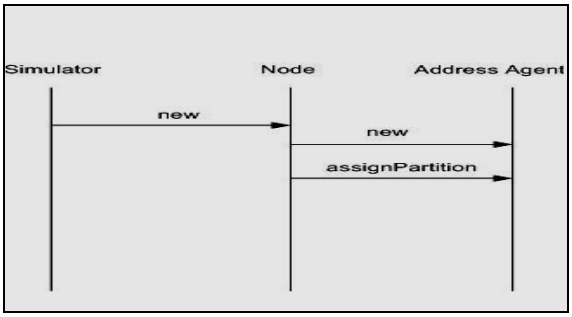

Figure 5: Network initialization of FAP

Address Filter that the node floods the AREQ messages Nf times in the network. If there is other initiator 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 into its filter, finally address received.
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 detection of partition. If the initiator node receives the same address with different identifier, then 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.

5.2. Node Ingress (or) Joining and Partition Merge Events

During the node joining the network Host node checks the messages whether for the joining procedure or for partition procedure i.e. shown in Fig.6. After the initialization, if any node tries to join the network, then Address Agent takes responsibility by removing filter signature from the nodes which are already present in the network. Then checks the address of the joining node or partition by exchanging AF messages. Then those newly coming nodes addresses are inserted into filter, which are present in every nodes. Once addresses are inserted into the filter, hash of the filter is calculated for partition detections.

Figure 6: Node Ingress (or) Joining of Node of FAP

Nodes in different partitions choose their own 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. 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 belongs different partitions. 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 message 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. Finally, all the nodes merges the other partition filter with its own filter, insert the addresses received in the AREQs into the new filter, then new filter is updated filter and hash of the update filter yields updated filter signature

5.3. Nodes Departure

When a node leaves the network or it shutdown in the network, its address should become available for the joining nodes. The departure of the node is shown in Fig.7. So each time every node verifies that its filter fraction bit to check or to know the departure of node.

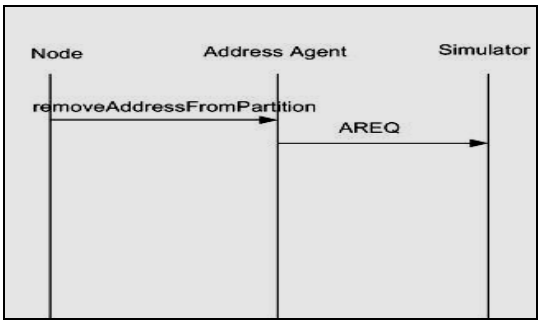

Figure 7: Nodes Departure of FAP

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.

6. Simulation Results

Performance of the protocol is evaluated using tool as Network Simulator (ns-2). Proposed protocol is compared with existing protocol like DAD by considering three main parameters which are delay for unique address allocation, control overhead required and collision rate while assigning addresses to the nodes.

6.1. Delay

Time taken by each node for the joining node procedure and on network partition merging events shown in Fig.8.From the figure FAP face less delay compared to DAD protocol.

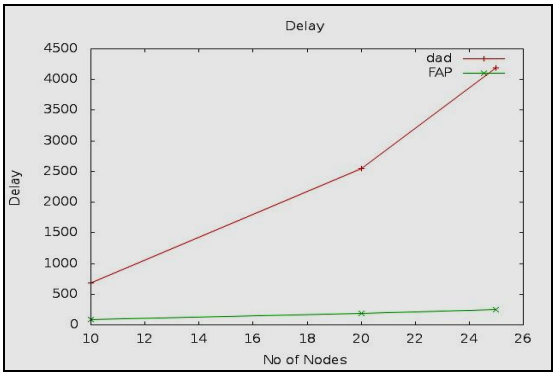

Figure 8: Comparison between numbers of nodes increases the delay of the DAD and the FAP

6.2. Control Overhead

It denotes the number of messages involved in address initialization and comparison and allocation process in Fig.9. Here FAP needs less control overhead compared to DAD protocol.

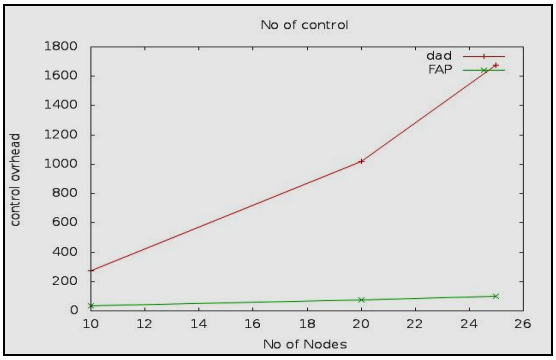

Figure 9: Comparison between numbers of nodes increases the overhead of the DAD and the FAP

6.3. Collision Rate

It denotes the number of collision possible in address initialization and allocation process in Fig.10.Here less chance of collision of address is possible compared to DAD protocol.

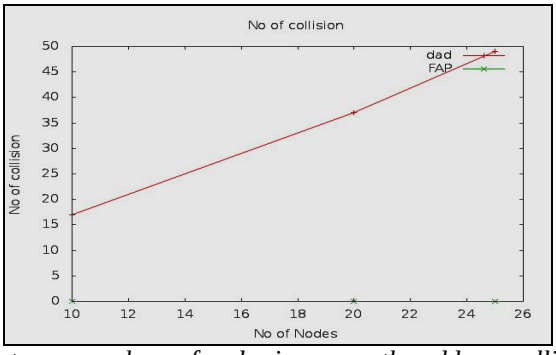

Figure 10: Comparison between numbers of nodes increases the address collision in DAD and the FAP

Proposed protocol FAP gives greater advantages compared to the existing protocol in terms of control overhead, collision probability and delay for address allocation. In the first analyze, the impact of delay for address allocation while node joining the network. A rectangular space with nodes distributed in grid is considered. The delay after the last node joins the network and the control overhead to assign unique address is measured and also collision rate is also measured. Simulation results reveal that proposed protocol resolves all the address collisions and also reduces the control traffic and delay when compared to existing protocol.

7. Conclusion

Address assignment in ad hoc networks should be automatic, fast and without collisions. The Filter based Addressing protocol (FAP), which uses address filters to reduce the control load and the delay to allocate addresses. Besides, filters allow an accurate partition merging detection and increase the protocol robustness.

Simulation results shows that proposed system resolves all the address collisions even during partition mergings, and 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.

8. References

1. D. O. Cunha, O. C. M. B. Duarte, and G. Pujolle, "A cooperation aware routing scheme for fast varying fading wireless channels," IEEE Commun. Lett., vol. 12, no. 10, pp. 794–796, Oct. 2008
2. T. Narten and R. Draves, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, RFC 3041, Jan. 2001.
3. S. Thomson and T. Narten, *IPv6 stateless address autoconfiguration*, RFC 2462, Dec. 1998.
4. 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.
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. A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," *Internet Math.*, vol. 1, pp. 485–509, 2002. 1304 1311