A Novel Lightweight Protocol for Address Assignment in Ad Hoc Networks Based on Filters

M. Anusuya Shyamala and R. Velayutham

Abstract Ad hoc networks play a vital role in latest technologies, but assigning address to the nodes in ad hoc network is a major challenge due to its lack of infrastructure. A lightweight protocol is proposed in this work which helps to configure the mobile ad hoc nodes which are based on a distributed address database that is stored in filters which reduces the control load and also makes the proposal strong in partition of networks and to loss of packets. The performance of the protocol is evaluated by considering the joining nodes, partition merging events, and initialization of networks. The result of control message reduction and address collisions for the proposed protocol is shown in simulation results.

Keywords Ad hoc networks · FAP · DAD · MANETconf

1 Introduction

Mobile ad hoc networks are a special category of networks exclusively classified on the basis of attributes like dynamic topology and infrastructurelessness [1]. Due to the dynamic nature and inherent infrastructureless architecture, solutions developed for configuration and deployment of infrastructure-oriented networks cannot be directly applied in MANETs [2]. Assigning address in ad hoc networks is a challenging one due to its self-organized environment. Network address translation (NAT) or dynamic host configuration protocol (DHCP) [3] feels difficult with the distributed feature of ad hoc networks and also does not address with the network partition and node merging.

M. Anusuya Shyamala (✉) · R. Velayutham
CSE Department, Einstein College of Engineering, Tirunelveli, Tamil Nadu, India
e-mail: mshyashya@gmail.com

R. Velayutham
e-mail: rsvel_kumar@yahoo.co.uk

L.P. Suresh et al. (eds.), *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, Advances in Intelligent Systems and Computing 324,
DOI 10.1007/978-81-322-2126-5_81

An efficacious approach called as filter-based addressing protocol (FAP) is proposed in this work [4]. A distributed database allocated address is maintained by FAP, thus database is stored in filters. These addresses are stored in a tamped manner. Both the bloom filter and the sequence filter are considered to design a filter-based protocol that ensures both the univocal address configuration of the nodes joining the network and detection of address collisions because each node can easily identify whether the address has been already allotted or not. Hash of this filter is used here as the partition identifier in order to find the network partitions. The hash of the filters is exchanged with the neighbors so that address collisions could be found by detecting the small control overhead where neighbors using different filters and then the filters are maintained and distributed with the nodes in the network .

This paper is structured as follows. Section-I contains the introduction. Overview of the related work is given in Sect. 2. Section 3 contains the proposed system. System design is given in Sect. 4. Simulation results are given in Sect. 5. Finally, conclusion of the paper is given in Sect. 6.

2 Existing System

2.1 Duplicate Address Detection (DAD)

Duplicate address detection (DAD) protocol is used as an addressing protocol where all the joining nodes selects an address in a random manner and informs the network with an Address REQuest message (AREQ). If the randomly chosen address is already allocated to another node, the already existing node announces the duplication to the joining node by replying with an Address REPly message (AREP). When the joining node receives an AREP, the joining node again selects another address and repeats the flooding process or else it allocates the chosen address. This protocol does not mind about network partitions and is not appropriate for ad hoc networks. The WDAD proposed aims at extending the DAD mechanism [5]. The idea behind WDAD is that duplicate addresses may be tolerated as long as packets reach the destination node intended by the sender .

The main drawback of WDAD is its dependency on the routing protocol. It requires some changes to the routing layer to support the introduction of the key identity. Each node will be identified at the routing layer by a kind of virtual address consisting in the combination of the IP address and the key value. In addition, WDAD detects address duplication based on local routing information; thus, it is totally adapted to proactive routing where each node maintains a complete routing table. For reactive routing, it is not the case where the nodes cache partial routing information for only ongoing and relayed connections, which reduces the possibility of detecting in moderate delays address duplication.

2.2 MANETconf

MANETconf is based on a "common distributed address table" where each node is able to assign IP addresses and maintains an allocation table that contains already allocated addresses and pending allocations [6]. Thus, the synchronization of these distributed tables constitutes the most critical and complex task of this protocol.

The advantages of this protocol are that it guarantees address uniqueness and it is totally distributed in terms that each node has the possibility to assign new addresses. In addition, it generates no unnecessary address changes when networks merge because only nodes involved in duplication release their IP addresses.

The problems of this protocol are its high complexity in terms of communication, table maintenance, and synchronization. The mechanism for assigning new addresses is bandwidth consuming; it consists of a network flood and a large number of unicast. All nodes should give their permission to the initiator to assign a new by another node. That's why each node generates large delays. Finally, this protocol is very sensible to network losses because of its dependency on unicast communication.

3 Proposed System

The proposed system is mainly used to auto-configure the network addresses dynamically, solving collisions with a low control load and in joining or merging events. This work uses FAP to achieve the addressing problems. FAP uses a distributed compact filter to represent the current set of allocated addresses [4]. Frequent node joining events can be simplified by the filters that are present at every node and also reduce the control overhead that is required to solve address collisions implicit in random address assignments. The filter signature is the hash of the address filter. It is considered as a partition identifier. Detection of network merging events can be easily found by filter signature. Two different types of filters are used in this work. They are bloom filter and sequence filter. The bloom filter is purely based on hash functions, and the sequence filter is used for compressed data based on the address sequence.

3.1 Bloom Filters

The bloom filter is a compact data structure which is most often used in distributed applications. It consists of an m-bit vector that represents a set A = {a_1, a_2…, a_n} through a set of independent hash functions h_1, h_2… $h_{k,}$ and the elements are inserted into the filter. Initially, all the bits in the filter are set to zero, and after that, all the elements are hashed where the output is set to 1 [7]. To check whether an

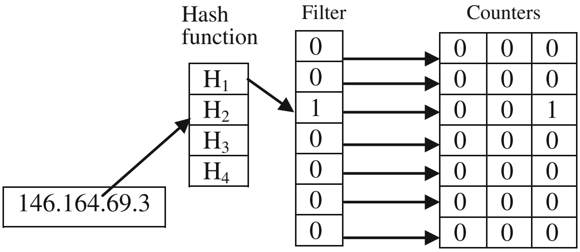


Fig. 1 Bloom filter

element is present in A, check whether the bits corresponding to $h_1(a_j)$, $\underline{h}_2(a_j)$, …, $h_k(a_j)$ are all set to 1, and if any one bit is 0, then a_j is not present in A. This shows the false-positive probability that an element $\mathrm{a_j}$ € A be recognized as being in A. Such cases may occur when the bits at the positions $h_1(a_\mathrm{j})$, $h_2(a_\mathrm{j})$, …, $h_k(a_j)$ are all set by previously inserted elements (Fig. 1).

3.2 Sequence Filters

Sequence filter stores and compacts addresses based on the sequence of addresses [7]. By concatenating the initial element that is the first element of the address sequence with an n-bit vector address range, the filter is created. Here, each address suffix is represented by one bit, indexed by Δ, which gives the distance between the initial element $\mathrm{a_{suffix}}$ and the current element a_{suffix}. The address with the given suffix is considered as inserted into the filter if the bit is in 1, or else if the bit is in 0, it says that the address is not in the filter. Since the available address is represented by its respective bit, neither false positives nor false negatives are present in sequence filter.

3.3 Selection of Filter

Network characters like number of nodes in the network and number of available addresses make to select the best filter for FAP. False-positive and false-negative rates of the filter are also considered. False negatives are not presented by the bloom filters, which show a membership test for an element that was placed in the filter is always positive. Though filters present a false-positive probability, a membership test of an element that was not placed into the bloom filter may be positive, whereas the sequence filter size is constant for the number of elements and the address range size increases. As a result, the bloom filter is more suitable for a large address range, whereas the sequence filter is more enough to a large number of elements. Thus, the filter can be selected based on our need.

4 System Design

4.1 Network Initialization

Auto-configuration of initial set of nodes is handled in network initialization. Two different events can occur here: gradual initialization, where the node arrives at enough time gaps between them, and abrupt initialization, where all nodes arrive at the same time. The first node chooses a partition identifier, and joining nodes are handled through the joining node procedure by the first node [3]. High control load can occur when partition merging events are triggered; such case may happen when all the nodes join the network at the same time and choose different partition identifier. It may also lead to address collisions by causing inconsistencies in the address allocation. FAP is suitable for both gradual and abrupt initialization using Hello message, and AREQ message is used to advertise that the previously available address is now allocated. AREQ messages have a unique identifier number which can be used to distinct the AREQ messages of the nodes.

4.2 Node Ingress and Network Merging Events

After the initialization phase, every node broadcasts periodic Hello message containing its address filter signature. After receiving a Hello message, the neighbor node detects merging events by evaluating the signature and the signature in the message by comparing. The nodes which have already joined the network alone can send Hello messages, can receive the request from a node, and can detect merging events. A node turns ON and listens to a medium for a time period, and if it receives a Hello message, the node assumes that it is a joining node and not an initiator node [3]. Again the joining node asks the host for the source node to send the address filter network using Address Filter (AF) message. The host node checks the bit after receiving the AF if the message is used in a node joining procedure or in a partition merging procedure (Fig. 2).

4.3 Node Departure

The address of the node leaving from the network should be available for other nodes, and when a node departing it floods the network with a notification to remove the address, the available address may scarce with time which can be identified in the address filter by the fraction of bits set to 1 in the bloom filter and in the sequence filter and also by the fraction of counters greater than one in the

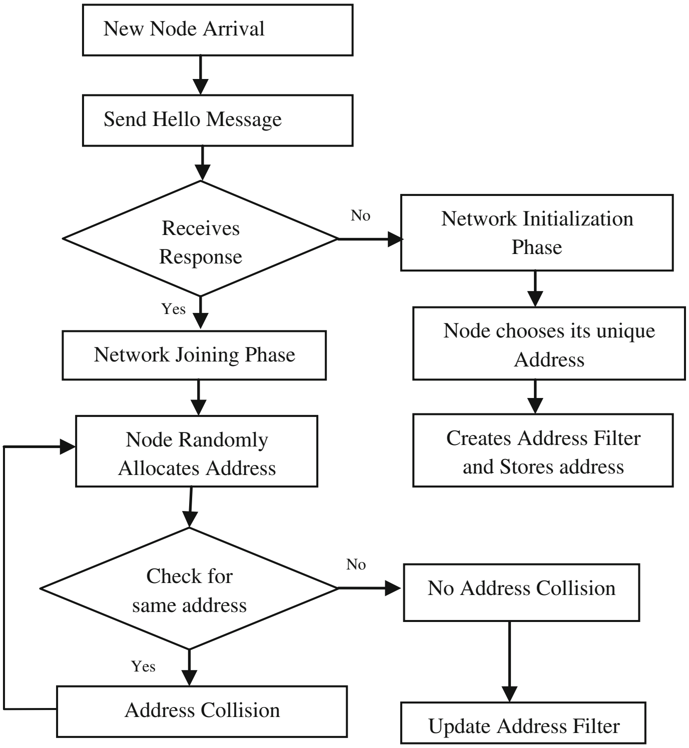


Fig. 2 Workflow architecture

counter bloom filter. Hence, whenever the filter is updated, the nodes verify this fraction in the address filter every time. If the fraction value reaches the threshold, address filter is reseted (i.e.) filter returns to initial as it is full.

5 Result Analysis

5.1 Probability of Collisions in FAP

A collision normally occurs in two different cases. First case is when any two joining nodes produce AREQs of same address and also with the partition identifier, and here, joining nodes do not note that their addresses are same because the message from other node looks like a retransmission of its message to the first node. Second case is when two disjoint partitions own exactly the same filters, and here, signatures of the Hello messages are same for both the partitions and as a result, the network would have an address collision; thus, partition merging procedure is not started in this case.

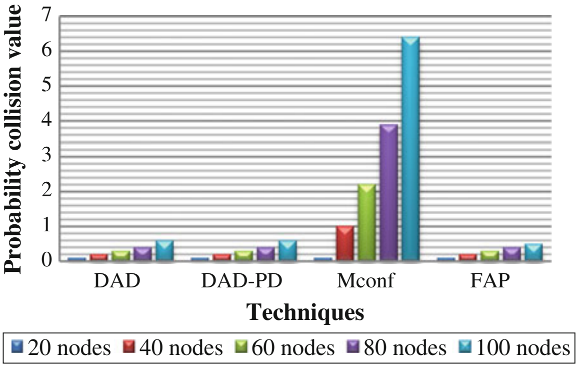


Fig. 3 Collision graph

The graph (Fig. 3) shows that MANETconf is vulnerable to address collisions on increasing the number of nodes. In this protocol, the abrupt initialization occurs through parallel partition merging procedures, which are not robust and cause collisions in the addresses.

5.2 Control Overhead Estimation

The procedures in addressing protocol like network initialization, node joining/ leaving, and merging reduce the available bandwidth by generating the overhead. FAP performs effective when compared to DAD-PD when one or more address collisions occur because this increases the number of floods only in DAD-PD and floods are the most costly operation. The ratio of the number of nodes in the network and the number of available addresses determines the value. MANETconf has a larger overhead compared to FAP because MANETconf is based on the assumption that all nodes must agree before allocating an address which demands many control messages. The initiator floods the network asking whether all nodes agree with the availability of chosen address (Fig. 4).

If all the other nodes agree with the chosen address, then the initiator floods the network again to allocate the chosen address. Besides the overhead caused by all the flood events, depending on the routing protocol, each unicasted message flow can imply in a flood to search for a route between the source and the destination nodes. Hence, an intensive use of uncast to different destinations, such as in MANETconf, can generate a high control overhead compared to FAP. Thus, the proposed protocol produces low control overhead.

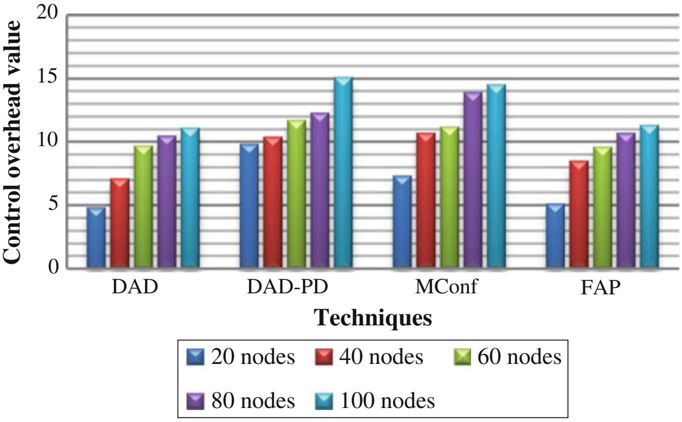


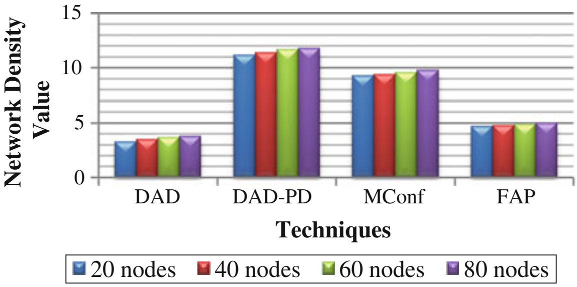
Fig. 4 Control overhead graph

5.3 *Impact of Network Density*

The impact of the network density is evaluated and the number of transmissions of flooding messages in abrupt network initialization. DAD-PD suffers a greater influence on the control load compared to FAP, MANETconf, and DAD because it floods the network many times due to the false positives in the partition merging detection. Compared to MANETconf and DAD-PD, FAP has a lower control load (Fig. 5).

The delay of FAP is lesser than the delay of DAD and MANETconf for a high number of nodes. DAD-PD has the greatest overlap delay and becomes unstable in networks with more number of nodes.

Fig. 5 Network density graph



6 Conclusion

Address assignment in ad hoc networks should be automatic, fast, and without collisions. A filter-based addressing protocol (FAP) is proposed, which uses address filters to reduce the control load and the delay to allocate addresses. Filters allow an accurate partition merging detection and increase the protocol robustness. Simulation results show that FAP resolves all the address collisions during partition merging. In the initialization of the network, the control load of FAP is more or less same as the control load of DAD which is a simple protocol that does not handle partition. Therefore, FAP is an efficient proposal to configure addresses automatically in the network. FAP withstands to message losses, which is considered as an important issue for ad hoc networks, which has fading channels and high bit error rates. Thus, the proposed protocol efficiently solves all the address collisions. FAP initialization process is also considered as the simplest one compared to other proposals.

References

1. S. Khalid, A. Mahboob, Design and implementation of id based MANET auto-configuration protocol. Int. J. Commun. Netw. Inf. Secur. (IJCNIS) **5**(3), 141–151 (2013)
2. S. Rafiul Hussain, S. Saha, A. Rahman, SAAMAN: scalable address auto configuration in mobile ad hoc networks. J. Netw. Syst. Manage. **19**, 394–426 (2010)
3. Z. Fan, S. Subramani, An address auto configuration protocol for IPv6 hosts in a mobile ad hoc network. Comput. Commun. **28**(4), 339–350 (2005)
4. N.C. Fernandes, M.D.D. Moreira, O.C.M.B. Duarte, An efficient and robust addressing protocol for node auto configuration in ad hoc networks. IEEE Trans. Networking **21**(3), 845–856 (2013)
5. N.H. Vaidya, Weak duplicate address detection in mobile ad hoc networks, in *Proceedings of the 3rd ACM International Symposium On Mobile Ad Hoc Networking & Computing*, pp 206–216 (2002)
6. S. Nesargi, R. Prakash, MANETconf: configuration of hosts in a mobile Ad Hoc network, in *Proceedings of 21st Annual IEEE INFOCOM*, vol 2, pp 1059–1068 (2002)
7. N.C. Fernandes, M.D. Moreira, O.C.M.B. Duarte, An efficient filter-based addressing protocol for autoconfiguration of mobile ad hoc networks, in *Proceedings of 28th IEEE INFOCOM*, pp 2464–2472 (2009)