Computing (2015) 97:309–331
DOI 10.1007/s00607-014-0417-6

Filter-based address autoconfiguration protocol (FAACP) for duplicate address detection and recovery in MANETs

T. R. Reshmi · K. Murugan

Received: 29 June 2013 / Accepted: 18 June 2014 / Published online: 20 July 2014

Abstract Assigning unique addresses to the nodes in mobile ad-hoc networks is a challenging issue due to dynamic topology, resource constraint, network merging and partitioning. The existing address autoconfiguration protocols designed to provide unique addresses to the nodes, address one or two of the challenges like efficiency of duplicate detection, address space management, scalability etc. In this paper, a scheme addressing most of the issues challenging the autoconfiguration is presented. A filter-based address autoconfiguration protocol (FAACP) for duplicate address detection and recovery scheme has been proposed which use sequence filtering technique for address space management. The scheme present a grid structured network topology which manages the network merging and partitioning in effective manner. The specialized nodes called "Unique IP Address Verification Agents" are dynamically selected to improve the efficiency and reliability of distributed duplicate detection. The scheme uses significantly less number of control packets, and hence incurs less address acquisition delay and communication overhead. The FAACP scheme is simulated in Network Simulator-2 and has proven to be scalable without significant change in the performance. The scheme performs well inspite of the packet losses that occur due to high node mobility.

Keywords Autoconfiguration · Addressing · MANETs · Duplicate address detection · Stateless protocol

T. R. Reshmi · K. Murugan (✉)
Ramanujan Computing Center, Anna Univerity, Chennai 600025, India
e-mail: murugan@annauniv.edu

T. R. Reshmi
e-mail: reshmi.engg@gmail.com

Mathematical Subject Classification 68M10 Network design and communication · 68M12 Network protocols

1 Introduction

The wireless network is broadly divided into two categories as infrastructure networks and infrastructure-less networks. A self-organized network without any infrastructure is defined as the mobile ad-hoc network (MANETs). The MANETs consist of set of mobile nodes, which are capable of self-configuring and routing packets in between them. These networks are immensely used in disaster recovery, defense environment and applications used at home and offices. The MANETs are characterized as autonomous and self-management networks that shares common communication channel with constrained bandwidth. The connectivity of the nodes depends greatly on the IP address. The network will be benefited if the network addresses are assigned automatically. Thus, there is a need for autonomous address assignment to overcome IP address conflict and to hold up spontaneous network. When two or more MANETs are merged together there arises the problem of routing errors due to address conflicts. Hence an efficient autoconfiguration method is required to solve the issues. The nodes in the network can automatically create link-local address on their own in autoconfiguration schemes, and can obtain additional network prefixes later from routing table updates. The characteristics of the address autoconfiguration schemes are described as follows:

Uniqueness For every network, each node must obtain a unique IP address. If the auto configuration scheme fails to ensure uniqueness, then it may leads to severe challenges such as misrouting or failure of service in regard with duplicate addressing in routing.

Scalability Communication overhead and allocation latency are the two factors that affect the scalability of the scheme. Communication overhead can be described as the number of packets that has been exchanged to obtain an IP address, and the allocation latency is the waiting time of a node to obtain an address.

Independency of routing protocols Typically, the routing protocols are broadly classified into two: proactive routing protocols and reactive routing protocols. The address autoconfiguration scheme ought to operate in a MANET regardless of the type of routing algorithm.

Reusability (*Garbage collection and IP leaks*) The address of a node which has left the MANETs must be traced and added as the free address list in the address pool. This is a challenging task in distributed schemes. The scheme that lacks a technique for handling address reuse will suffer the address leakage problem.

Availability The address autoconfiguration schemes must always be available regardless of network status. The availability is highly essential during the network situations such as the network partition and merging, node mobility and packet losses.

The autoconfiguration protocols are classified as stateful or stateless approaches. In stateful protocols, the address allocation tables keep the state information of the nodes and the addresses that have already been configured in the network. In these schemes few nodes will act as Dynamic Host Configuration Protocol (DHCP) servers to distribute the addresses to the requesting nodes. In stateless approach, each node self-generates its address and communicates with every other node to check the duplication of an address. If the duplication is detected, the node opt another address and does the duplication detection. An autoconfiguration scheme for MANETs must ensure simplicity, hastiness and capability of handling MANET characteristics such as mobility, power as well as memory limitations, radio coverage and packet loss. It must also ensure minimum exchange of packets and computational overhead during the functioning of the scheme.

The paper is organized as follows. Section 2 summarizes the related works about the autoconfiguration protocols, Sect. 3 presents the system model and the description of working of the Filter-based Address Autoconfiguration Protocol (FAACP) scheme, and Sect. 4 provides the performance analysis and simulation results of the scheme. Finally Sect. 5 states the conclusion and future works of the scheme.

2 Related work

An address pool based autoconfiguration protocol was proposed by Perkins et al. [1]. The protocol used two sets of address pools called temporary and permanent address pool. The newly joining node initially selects an address from the temporary address pool and checks for the duplication of IP address selected from permanent address pool. If any of the nodes own the same address, the node chooses another address until it finds a non-duplicated address. This scheme uses a flooding scheme to check the duplication using a duplicate address detection (DAD) scheme. The scheme performance deteriorates if two or more nodes choose the same temporary address. The scheme does not scale and does not support network merging and partitioning.

The Weak DAD proposed by Vaidya et al. [2] tried to overcome the issues caused by flooding of packets during the process of DAD. The scheme used a method of integrating unique keys to each addresses and thereby ensuring uniqueness to allocated IP addresses. The keys selected are incorporated into the routing packets and checked for the uniqueness of IP address during routing. If any of the nodes detects the same IP address with different keys, it intimate the concerned nodes about the duplication of IP address. The scheme reduced the probability of duplication to great extent. But as the key length is not constant, it induces extra overhead to the packets exchanged between the nodes.

Gutman et al. [3] proposed a DAD algorithm to assign unique link local address for every node in a network. The address range selected by the nodes are within the range 169.254.1.0–169.254.254.255. This approach focused on wired networks and was not practical in MANETs because of the multi-hop behaviors. Park et al. [4] extended the work to MANETs using IPv6 site-local addresses. The site-local addresses using different subnet-ID uses the neighbor discovery protocol (NDP) messages to check the duplication of addresses. It used a flooding mechanism which causes overhead and it does not discuss the ways to overcome message losses, network congestion and so on.

A stateless autoconfiguration protocol (SLAAC) scheme was proposed by Thomson et al. [5]. The scheme was specifically for IPv6 based MANETs and was different from earlier IPv4-based protocols. The IPv6 node has the capability to assign several addresses to the same interface and each of the IP address is divided in to a suffix and prefix. The node receives the prefix advertisement from the router, based on the network prefix. The suffix is generated by the same node using the IEEE MAC identifier. The derived IP address is checked for duplication using a flooding technique. The scheme assumes that the IEEE MAC identifiers are unique and hence the derived IP addresses are also unique. But there are many existing techniques to change the MAC identifiers, so these schemes are not feasible to allocate unique addresses. The flooding technique used for duplicate detection makes it non-scalable.

A Hierarchical SLAAC scheme was proposed by Weniger et al. [6] to overcome the scalability issue in the working of SLAAC. The scheme divides the network into many subnets and each of the subnet has a leader node which acts as edge router and issues *Router Advertisement* (RA) message to their scope. The RA message contains the network prefix, which is used as the network prefix of the IP address. The suffix is generated using the same algorithm used in SLAAC [5]. The node checks the duplication of the derived IP address with the help of the leader node. The leader node selection algorithm and maintenance of hierarchical address structure incurs cost and resource consumption issues in MANET deployment.

Weniger et al. [7] proposed passive duplicate address detection scheme for MANETs called PACMAN. The scheme is a hybrid autoconfiguration protocol which uses the feature of self-generating the IP address and checking its duplication using the stateful information stored in the centralized nodes. The routing protocols are used to ensure the uniqueness of the IP address. The scheme is dependent on the routing protocols and induces complexity during the implementation of MANETs. Scalability of the scheme is low, as it induces complexity of integrating routing and addressing together.

Sonia Mettali Gammar et al. [8] proposed a distributed IP address configuration approach for called "DAACP" for MANETs. Allocating unique IP address for every node in the network is the main objective of the scheme. The scheme is a fault tolerant address allocation approach, since it assures address allocation under various situations such as node failures, network partitioning and merging. They have also addressed the address management approach, which recovers the lost addresses and spaces. The scheme selects an IP address offer from the configuring node with the large address buffer size. It does not consider other resource constraints of the configuring node during the implementation.

Luis Javier García Villalba it et al. [9,10] introduced two schemes: Distributed Dynamic Host Configuration Protocol (D2HCP) and Enhanced Distributed Dynamic Host Configuration Protocol (ED2HCP). The autoconfiguration protocol uses a stateful scheme and exploits the Optimized Link State Routing Protocol (OLSR) for synchronization. It ensures the uniqueness of the IP addresses under various network conditions such as link failure, message losses and network partition. In these schemes, every node is responsible for handling a range of addresses. Any node in the network gives half of its address range, when a new node joins the network. The schemes are dependent on the link state protocols and hence are not applicable for other routing protocols.

Syed Rafiul Hussain et al. [11] proposed an efficient and scalable address autoconfiguration protocol called SAAMAN. The protocol automatically configures a network by allocating unique IP addresses to the nodes with low overhead and minimal processing cost. The scheme allocates Duplicate-IP address Detection Servers (DDS) in the network and checks the uniqueness of IP address. The scheme uses a grid based quad tree hierarchy to distribute the DDSs, and does not require any leader election. The DDS have to maintain IP address state information and check the duplication, so these servers consume a big memory space for storing the stateful information. The synchronization of the stateful information also incurs additional overhead in the network.

Natalia Castro Fernandes et al. [12] presented a Filter-based addressing protocol (FAP) for autoconfiguration in MANETs. The protocol employs a lightweight and reliable scheme which uses filters functionalities and probabilistic analysis for address autoconfiguration. The scheme performs well by handling merging and partitioning in static and mobile scenarios. The scheme uses a flooding technique for duplication detection and therefore causes network performance degradation.

Grajzer et al. [13] proposed an enhanced neighbor discovery protocol called ND++ for IPv6 based MANETs. The protocol is an extension of neighbor discovery protocol (NDP) with features of optimized link state routing protocol (OLSR). The protocol proves to handle uncertainties due to high mobility, network merging and partitioning. The protocol employs a routing based distributed method for duplication detection and hence incurs very less overhead. The scheme is routing dependent and therefore its usage is limited to networks with link state routing protocols.

3 Proposed scheme

The Filter-based Address Autoconfiguration Protocol (FAACP) scheme assumes the nodes in the network are deployed in grid structure. The network merging and partitioning can be well handled if the network topology is divided in to grid structures. Every node knows its geographic position and uses geographic forwarding scheme to route the packets. The technique exploits sequence filters to represent IP addresses in a compact manner. Initially temporary addresses are allocated to the nodes, with which the permanent address chosen are checked for duplication. By following three rules, the technique specializes some nodes as UAV agents and these nodes are responsible for allocating and maintaining conflict-free permanent IP addresses in the network. Every node chooses a random IP address and forwards it to the UAV agents. The UAV agents verify the UAV-Table (Example shown in Table 6) and then allocate the conflict-free IP address to the node. The proposed address retrieval scheme recovers both lost IP addresses and memory spaces simultaneously.

3.1 Motivation

The various autoconfiguration protocol schemes discussed in the related works satisfy only disjoint subsets of the characteristic requirements of the addressing schemes. Likewise the two appreciable schemes called “Scalable address autoconfiguration protocol for MANETs” (SAAMAN) [11] and “Filter based addressing protocol” (FAP)

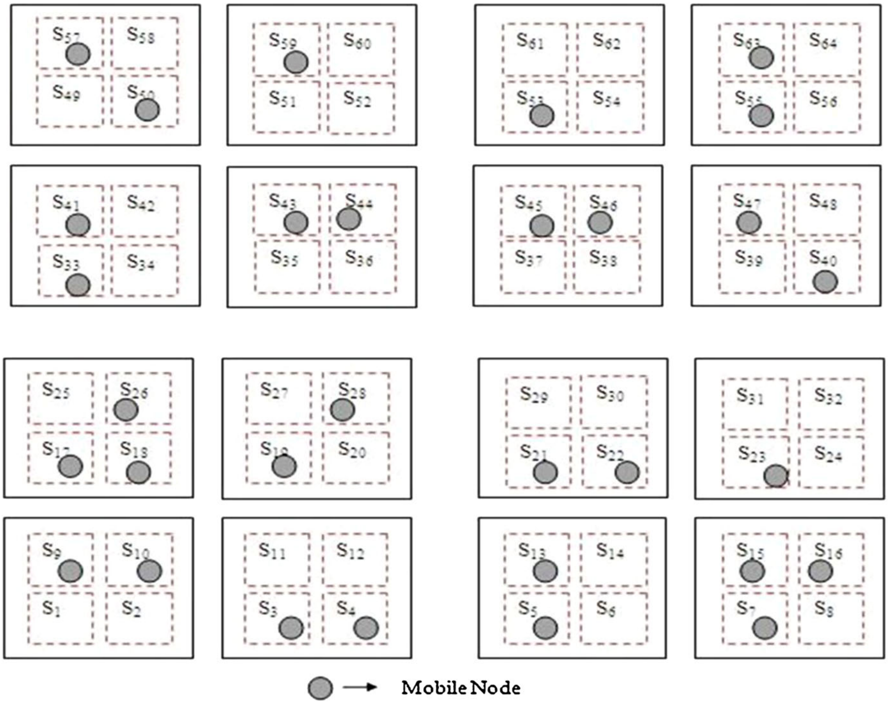


Fig. 1 The system design

[12] could not fulfill the complete requirements of the autoconfiguration protocols. FAP scheme has not utilized any techniques to verify the uniqueness of an address that is assigned to a node. Moreover the scheme faces the address leakage issue as it does not have any address recovery mechanism. Further, when the address space range reaches high, the performance of the scheme is decreased dramatically. The other scheme, SAAMAN uses the DDS servers to check the duplication of the allocated addresses. This scheme consumes high memory space for storing stateful information and has complex structures for the functioning of the protocol. Addressing the problems described in the above schemes, the FAACP scheme is presented, which combines the techniques employed in both schemes. The FAACP scheme performs well and guarantees the characteristic requirements of the autoconfiguration protocol.

3.2 System design

The FAACP scheme makes use of geographic forwarding scheme to employ routing packets among nodes. It uses Global Positioning System (GPS) to obtain a node's position such as altitude, latitude and longitude. According to the scheme, the network topology is divided into many hierarchical grid structures. The grid structure is organized as increasing sizes of squares as shown in Fig. 1. The grid structures are represented in terms of sectors. Any combination of lower sector squares cannot con-

Table 1 Format of PIT

Node's IP address	Node position	Velocity of node movement	Log time

Table 2 Sector-1 square components

Sector-1 square	Components
S_1	Nil
S_2	Nil
S_3	Nil
⋮	:
S_{64}	Nil

struct any higher sector square. Sector-1 square is the smallest unit of the grid. Four Sector-1 squares form Sector-2 square and again four Sector-2 squares form Sector-3 square and so on. The directive for constructing a grid is given in [11] and is stated as follows.

Every Sector-x square (where x $\geq$ 1) comprises of four sector of (x$-$1) squares, and each Sector-x square is a basic part of incredible sector of (x+i) squares, (where, i = 1, 2, 3…n). When the grid is constructed according to the rule stated above, and then a Sector-x square that its lower left coordinates will take the form of, $p.2^{x-1}, q.2^{x-1}$ (Where p and q are integers).

The system design given in Fig. 1 is a Sector-4 grid structure. The $S_1, S_2, S_3 \ldots S_{64}$ represents the Sector-1 squares. S_1, S_2, S_9 and S_{10} jointly form Sector-2 squares and go on. The hierarchical sectors and its components are represented in Tables 2, 3, 4, 5. Every node maintains a Position Information Table (PIT) that contains IP address of neighbor, position, velocity and time of recently received *HELLO* message. Periodically, every node forwards *HELLO* messages to their neighbors to intimate their existence with essential information like IP address and position details. On receiving *HELLO* messages from neighbors, each node enters corresponding node information in its PIT. The format of PIT is shown in Table 1.

3.3 Filter-based address space management

The addresses spaces of nodes are represented in a compact manner in the form of sequence filters. Before allocating an address to a node, the availability of addresses can be obtained by means of the filters. Further, partition and merging can be detected accurately by filters and it incurs low control overhead. By using filters, a node can become aware of partitions by verifying the hash value of its filter with other neighboring nodes filters. The Fig. 2 shows the compressed addresses of a node according to the address sequence. The sequence filter is deterministic and it does not construct any false-positives or false-negatives. The size of the filter depends on the size of address space and address size. The sequence filter employed for IPv4 and IPv6 vary in size because of the variation in address size.

Table 3 Sector-2 square components

Sector-2 square	Components	Sector-2 square	Components
A	S_1, S_2, S_9, S_{10}	I	$S_{33}, S_{34}, S_{41}, S_{42}$
B	S_3, S_4, S_{11}, S_{12}	J	$S_{35}, S_{36}, S_{43}, S_{44}$
C	S_5, S_6, S_{13}, S_{14}	K	$S_{37}, S_{38}, S_{45}, S_{46}$
D	S_7, S_8, S_{15}, S_{16}	L	$S_{39}, S_{40}, S_{47}, S_{48}$
E	$S_{17}, S_{18}, S_{25}, S_{26}$	M	$S_{49}, S_{50}, S_{57}, S_{58}$
F	$S_{19}, S_{20}, S_{27}, S_{28}$	N	$S_{51}, S_{52}, S_{59}, S_{60}$
G	$S_{21}, S_{22}, S_{29}, S_{30}$	O	$S_{53}, S_{54}, S_{61}, S_{62}$
H	$S_{23}, S_{24}, S_{31}, S_{32}$	P	$S_{55}, S_{56}, S_{63}, S_{64}$

Table 4 Sector-3 square components

Sector-3 square	Components
η	A, B, E, F
μ	C, D, G, H
σ	I, J, M, N
ω	K, L, O, P

Table 5 Sector-4 square components

Sector-4 square	Components
ψ	$\eta, \mu, \sigma, \omega$

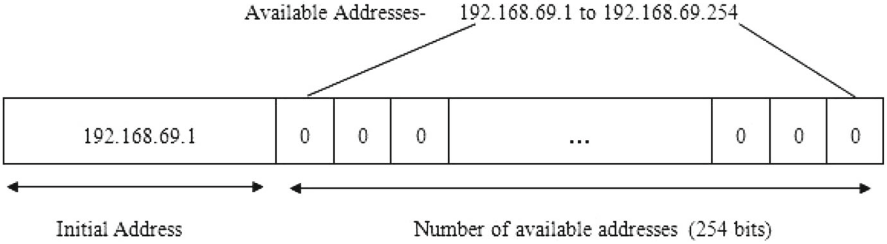

Available Addresses- 192.168.69.1 to 192.168.69.254
192.168.69.1
0
0
0
...
0
0
0
Initial Address
Number of available addresses (254 bits)

Fig. 2 The sequence filter

In sequence filter, the suffix of every address is represented with a single bit. It is determined by the position of the bit in the filter. If the bit value is set as 0, it represents that the address is unassigned and if the bit is set as 1, it represents the address is assigned. So if the bit value is set to 1, the request for the particular IP address allocation will be negatively acknowledged. The Fig. 3 shows an example of inserting the assigned address 192.168.69.2 in the sequence filter by setting the value of the host's concerned bit as 1.

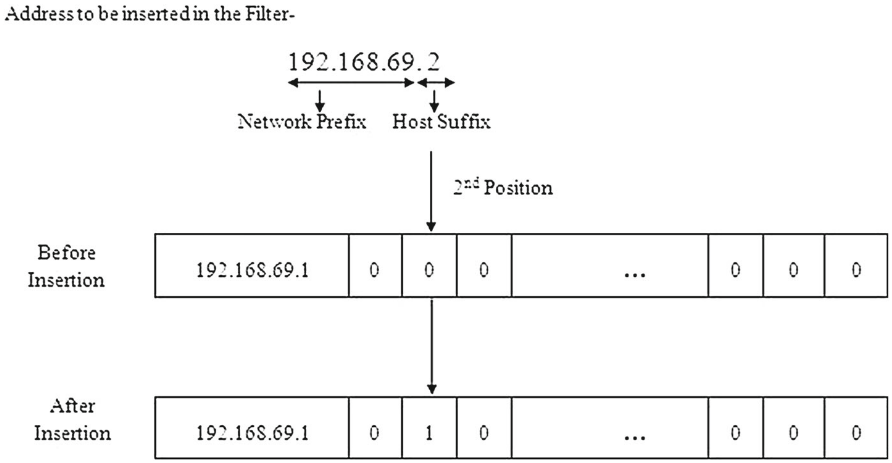


Fig. 3 The IP address insertion in the Sequence filter

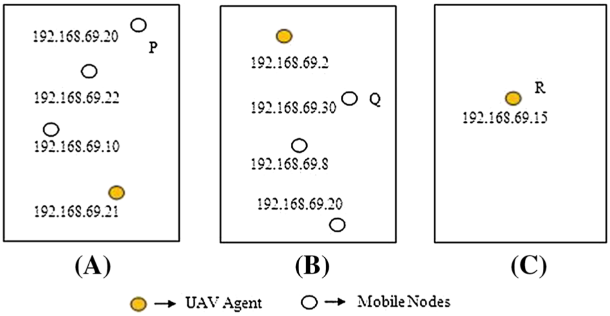


Fig. 4 UAV agent Selection Scheme

3.4 Duplication detection using unique IP address verification agents (UAVs)

The UAV agents are responsible for detecting the duplicate IP addresses in the network. UAV agents are selected based on two parameters namely selected IP address and predetermined grid hierarchy. To select UAV agents, the network exploits three rules as follows.[1]

Rule-1: For a sector square, node n_i can be elected as a UAV agent whose IP address is smallest, but higher than the IP address of node n_{i+1} for which a UAV agent is going to be elected. If the sector does not contain such a node then go to *Rule-2*.
Rule-2: For a sector square, node n_i can be elected as a UAV agent whose IP address is absolutely smallest in that sector. If the sector does not contain such a node, then go to *Rule-3*.
Rule-3: For a sector square, node n_i can be elected as its own UAV agent.

[1] Consider n_i and n_{i+1} represent two nodes in the network, where i = 1, 2,…, N and N is the total number of nodes in the network.

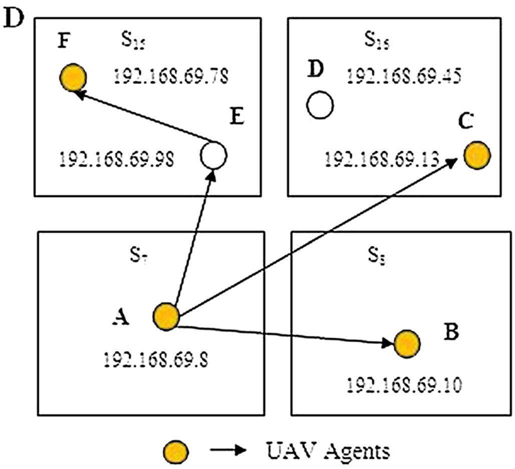


Fig. 5 Selection of UAV agents

In Fig. 4a, UAV agent has to be selected for the duplication detection of node P's IP address. It has three neighbors namely 192.168.69.22, 192.168.69.10 and 192.168.69.21.Initially, *Rule-1* is applied in this scenario and node with IP address 192.168.69.21 is elected as UAV agent as the node has smaller IP address (192.168.69.21) compared to 192.168.69.22, and higher compared to node P's IP address (192.168.69.20). Though IP address 192.168.69.10 is small, it cannot be elected as UAV agent, since it is less than node P's IP address, which violates *Rule-1*. In Fig. 4b, UAV agent is selected for duplication detection of node Q's IP address. At first, *Rule-1* is applied in this scenario but it will not be satisfied for the UAV agent selection. Therefore, *Rule-2* is applied and the IP address 192.168.69.2 is elected as UAV agent, which is the absolute smallest IP address. In Fig. 4c, the sector contains only one node hence *Rule-3* is applied and node R itself is selected as UAV agent.

The FAACP scheme is a distributed solution as it does not use any centralized database to store IP addresses of the nodes. Assigning a node as UAV agent does not require any pre-agreement. To offer a distributed IP address management service, every node maintains a UAV agent table (UAV-Table). This table contains the IP address, and sector position of nodes for which it acts as a UAV agent. In this scheme, *HELLO* messages and Notification messages (*Notify*) are used for selecting UAV agents. As the technique uses geographic forwarding scheme, every node in the sector forwards its information to other neighboring nodes. Thus, while triggering Sector-x UAV algorithm, the network has the capability to set up Sector-(x+1) UAVs. As Sector-1 is the smallest Sector square in the Grid structure and nodes within it have mutual transmission range, and hence each node knows the other nodes with periodic *HELLO* messages.

The Fig. 5 shows the Sector-2 square-D which constitutes, S_7, S_8, S_{15} and S_{16} Sector-1 squares. The S_7 has only node A present in it, therefore it will not receive any *HELLO* messages. So after the time interval T_n, node A is elected as UAV agent for itself. Similarly based on the rules explained in the above section, node B, C and F are selected as UAV agents in S_8, S_{16}, and S_{15} respectively. The *HELLO* messages from

Table 6 UAV-table of node F

Node-ID	Node's IP address	UAV- IP addresses	Position
F	192.168.69.78	192.168.69.78	S_{15}
		192.168.69.98	S_{15}
		192.168.69.45	S_{16}
		192.168.69.13	S_{16}
		192.168.69.8	S_7
		192.168.69.10	S_8

its neighbors are used for the selection of UAVs and subsequently the IP addresses of the nodes are updated in corresponding UAV-Tables.

The format of UAV-Table for node F is shown in Table 6. The first field represents the Node ID, the second field represents the IP address of the node. The third field records the IP addresses of the UAV agents in the network and the fourth field indicate the position of the UAVs in the sector squares.

After time interval T_{n+1}, every node in Sector-2 square forwards *Notify* messages to all its neighbors. It includes the IP address, position and time of last received *Notify* message. Based on received *Notify* messages, each UAV agent modifies its UAV-Table. For example consider *Notify* messages of node A. When node A forwards*Notify* message to node E, it compares its IP address with its neighbor node F and finds that the suitable node to become UAV agent is node F. Then, it forwards *Notify* message to node F. On receiving *Notify* message, it updates its UAV-Table with the changes. Now, node A transmits *Notify* message to node C. While receiving the *Notify* message, the node checks the IP addresses of its neighbors and finds itself to be suitable to act as UAV agent. Therefore, it drops the *Notify* message and updates its UAV-table. Since, there is no node other than node B in S_8, node B remains as the UAV agent even after receiving *Notify* message. The similar operation is performed in each and every sector, and UAV agents are selected and updated.

3.5 IP Address Allocation

There are two types of addresses allocated during the scheme. Initially a temporary address is allocated which is used to configure the permanent address of the node. Both of the allocations are explained in the section given below.

(A) Temporary IP address allocation

The temporary IP addresses are used as the source address by the nodes for transmitting *Probe* messages to the network. These messages are useful to allocate successful conflict free addresses in the node. Generally the temporary IP addresses are in the range between 1 and 2048, then temporary addresses of Sector-1 square is defined to have addresses from 1 to 32 and Sector-2 square from 33 to 64 and so on. Thus, this temporary address allocation strategy lessens the IP address conflicts to high extent. Though, different Sectors have different range of IP address, and sometimes two or more new nodes may select the same

IP address that results in IP address conflicts. Every node runs a DAD algorithms in order to avoid these address conflicts.

The temporary IP address allocation is done as follows.

1. When a node enters the network, it first recognizes its position through GPS, and then measures the Sector-1 square in which it resides.
2. Secondly, every node collects the information about the neighboring nodes within its Sector-1 square using *HELLO* messages
3. The node chooses a random temporary IP address from the temporary IP address range reserved for its Sector-1 square and monitors the NAT table for any conflicts.
4. For assuring the conflict-free temporary IP address, the node runs DAD algorithm. If it gets a *Positive Acknowledgement* (*PACK*) message, it takes the corresponding IP address as its temporary IP address and starts the permanent IP address allocation strategy.
5. If the node receives *Negative Acknowledgement* (*NACK*), it gives up the selected IP address and chooses another random IP address from temporary IP address group and repeat DAD algorithm. This process is repeated until the node obtains conflict-free temporary IP address.

(B) Permanent IP address allocation

In this scheme, the permanent IP address allocation strategy is well designed for achieving scalability. Even though we assign disjoint IP address ranges for every sector, after a long time there can be an issue as the number of newly joining node may exceed predefined IP addresses in Sector squares. In that case the newly joining nodes may have to use the IP addresses that are already in use. To combat this situation, the permanent IP address allocation strategy is designed. According to this strategy, once a node obtains temporary IP address, it immediately selects permanent IP address. Then, it transmits *Probe* message to all its UAV agents. The *Probe* message comprises of selected permanent IP address, temporary IP address and position in Grid structure. After transmitting *Probe* message, the node starts timer T_{ACK}. By receiving the *Probe* message, UAV agents observes the address space for address conflicts. If any UAV agent discovers IP address conflict, it immediately transmits *NACK* message to the corresponding node. On receiving *NACK* message, the node selects another permanent IP address and repeats the same process. If a node does not receive any response from the UAV agents even after the expiration of T_{ACK}, then it is considered as a positive response and chooses the same IP address as permanent IP address. Initially, the node transmits *Probe* message only to UAV agents in neighboring Sector-1 squares and then, it is transmitted to other higher sector squares. When an address conflict is identified at Sector-1 squares, then there is no need for retransmitting the *Probe* messages to other Sector (x+1) squares.

3.6 Address retrieval scheme

Every UAV agent is assigned with a sufficient addressing space based on the type of application and network. The allocated addressing spaces are disjoint to one another. Though the address blocks are separated, it becomes more space restricted after many configurations. Consider $\mathrm{UAV_i}$ and $\mathrm{UAV_{i+1}}$ as the two UAV agents. Each UAV agent upholds four parameters namely, Initial lower limit of the addressing space ($\mathrm{L_I}$), present lower limit of the addressing space ($\mathrm{L_P}$), initial upper limit of the addressing space ($\mathrm{U_I}$), and present upper limit of the addressing space ($\mathrm{U_P}$).

When the $\mathrm{UAV_i}$ reaches ($\mathrm{L_P} = \mathrm{U_P}$), UAV agent will not have enough addressing space to offer and maintain new addresses. Therefore, $\mathrm{UAV_i}$ immediately disseminates recovery request (*Re-REQ*) message in the network. *Re-REQ* comprises the parameters $\mathrm{L_I}$, $\mathrm{L_P}$, $\mathrm{U_I}$, and $\mathrm{U_P}$. UAV agents that receive *Re-REQ* messages will trigger their recovery process. Thus, the recovery processes of UAV agents are accomplished at the same time. At the end of dissemination of *Re-REQ* messages, every node receives many *Re-REQ* messages. However, a node do not have to reply to all the recovery messages it has received, rather, it has to reply only to the *Re-REQ* message that follows the following rule.

> *Rule-4:* If the node's own address belongs to the range $[\mathrm{L_I},\ \mathrm{L_P}]$ of the parameters in *Re-REQ* message, then it should reply to that corresponding *Re-REQ* message.

If a node satisfies the condition in *Rule-4*, then it sends a reply (*Re-REP*) back to the $\mathrm{UAV_i}$ agent indicating that the node is still alive. Based on the received *Re-REP* messages, $\mathrm{UAV_i}$ agent constructs a list of non-existent nodes. In some cases, the UAV agents may leave the network and the address spaces utilized by them is no longer used by any other agents. These lost address spaces can be recovered as along with recovery process. According to this, every node examines each *Re-REQ* message that it has received for identifying the ultimate recovery address spaces. Whenever $\mathrm{UAV_i}$ agent receives *Re-REQ* message from $\mathrm{UAV_{i+1}}$ agent, it evaluates the area of intersection between $\mathrm{UAV_i}$'s ultimate recovery memory spaces [min, $\mathrm{L_I}\,(\mathrm{UAV_i}) - 1$] and [$\mathrm{U_P}\,(\mathrm{UAV_i}) + 1$, max], and the memory space maintained by $\mathrm{UAV_{i+1}}$ [$\mathrm{L_I}\,(\mathrm{UAV_{i+1}}),\ \mathrm{U_P}\,(\mathrm{UAV_{i+1}})$]. After the evaluation of area of memory space intersection between two UAV agents ($\mathrm{UAV_i}$ and $\mathrm{UAV_{i+1}}$), the technique allows $\mathrm{UAV_i}$ to resume the recovered memory spaces.

The sequence of processes of the FAACP scheme is explained with a flow chart given in Fig. 6 for better understanding of the scheme.

4 Performance evaluation

The performance of FAACP scheme is evaluated using mathematical formulations and simulations. FAACP is an extension of SAAMAN (discussed in Sect. 3.1) for enhanced duplicate detection and address recovery. FAACP is a hybrid autoconfiguration scheme implemented in a hierarchical network topology, which is similar to that used in SAAMAN [11]. Similarly DAACP [8] is the only recent scheme which handles with address recovery and losses. Hence the performance of the FAACP scheme is compared with SAAMAN and DAACP schemes.

4.1 Cost analysis

The cost of the schemes is analyzed based on the memory used for storing the stateful autoconfiguration information such as allocated addresses and free addresses. The

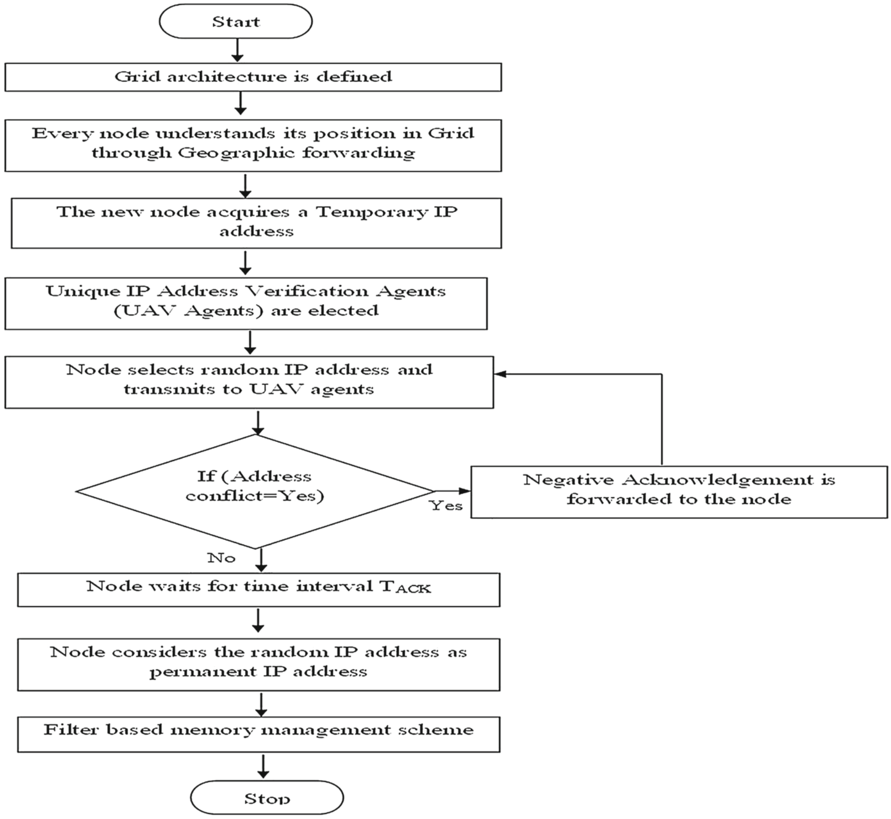

Start
Grid architecture is defined
Every node understands its position in Grid through Geographic forwarding
The new node acquires a Temporary IP address
Unique IP Address Verification Agents (UAV Agents) are elected
Node selects random IP address and transmits to UAV agents
If (Address conflict=Yes)
Yes
Negative Acknowledgement is forwarded to the node
No
Node waits for time interval T_ACK
Node considers the random IP address as permanent IP address
Filter based memory management scheme
Stop

Fig. 6 Flowchart of the FAACP scheme

memory usage of the schemes (C_{mem}) for storing information about 'n' addresses is discussed in [14] and is given by

$$C_{mem} = n \times number\ of\ bits\ needed\ to\ store\ address\ information \tag{1}$$

The Eq. 1 clearly states that the memory usage of the scheme is dependent on the representation of stateful information. The stateful information stored in DAACP [8] and SAAMAN [11] schemes uses a 32 bit representation of IP addresses[2] whereas the FAACP scheme uses 1 bit representation of the IP addresses using sequence filter (Explained using Fig. 2). The memory usage of the schemes are compared and tabulated in Table 7. The memory consumption of the schemes at different network prefixes is shown in Table 8. The value in Table 8 concludes that FAACP consumes less memory.

[2] The scheme here uses the IPv4 addresses of 32 bit length. The scheme can also use IPv6 addresses of 128 bit length.

Table 7 Memory usage of the autoconfiguration schemes

Memory usage (n addresses)	DAACP	SAAMAN	FAACP
	32n bits	32n bits	32 + n bits

Table 8 Memory consumption of autoconfiguration schemes at different network prefix

Network prefix	Memory usage in Kbytes		
	FAACP	DAACP	SAAMAN
/8	2048	65536	65536
/12	128	4096	4096
/16	8	256	256
/20	0.5	16	16
/24	0.04	1	1
/28	0.0059	0.0625	0.0625

Table 9 Simulation parameters

Parameter type	Value
Number of nodes	50, 100, 150, 200, 250
Mobility model	Random waypoint
Node mobility	10, 20, 30, 40, 50 m/s
Simulation area	1250 × 1250 m^2
Simulation duration	50 s
MAC protocol	IEEE 802.11b
Transmission range	250 m
Routing protocol	AODV

4.2 Simulation environment

The FAACP scheme and two of the existing schemes such as DAACP [8] and SAAMAN [11] were implemented in Network Simulator (ns-2) [15] for comparison and analysis. The AODV protocol is used as the underlying routing protocol. The message format used in the scheme supports the "Generalized MANET Packet/Message Format" [16] and the additional information (location, neighbor list, etc.) are added into the packet header through its type-length-value (TLV) block. Detailed simulation parameters are described in Table 9.

The nodes are deployed in a coverage area of 1250 × 1250 m^2, and the size of an Order-1 square is assumed to be 156.25 × 156.25 m^2. The simulation results of the schemes are plotted with an average of 20 runs. The message exchanges in the scheme include *HELLO, Notify, Probe,* Positive Acknowledgement (*PACK*), Negative Acknowledgment (*NACK*), Recovery Request and Reply (*Re-REQ, Re-REP*) messages. These messages are analyzed in the simulation. The simulation focuses on the following five metrics to analyze the performance of the schemes.

- *Protocol overhead*: It is defined as the total number of control and maintenance message packets flooded in the network.
- *Address acquisition delay*: The time taken by the node for a successful address configuration and allocation is calculated as the address acquisition delay.
- *Successful allocation ratio*: It is defined as the ratio of the number of successful addresses allocated after the duplicate detection, to the total number of address requested.
- *Packet loss*: The total number of control and maintenance packets lost during the working of the scheme is defined as packet loss.
- *Address reclamations*: The total number of addresses reclaimed during address space management is analyzed to evaluate the schemes.

The simulations use the same arrival and departure patterns. The time interval between the successive node arrivals is set to 0.2 s. The time interval between the node departures is set to 0.25 s. The arrivals and departures are simulated as independent procedures. The time interval between the *HELLO* messages is set to 3 s. The acknowledgement timer T_{ACK} is set to 0.15 s. The threshold number of DAD trials is set to three times.

4.3 Simulation results and analysis

The performance of the FAACP scheme is evaluated by using five of the performance metrics: (1) protocol overhead; (2) address acquisition delay; (3) successful allocation ratio; (4) packet loss and (5) address reclamations. The evaluations are done with respect to two factors:

I. The number of total nodes in the network (from 50 to 250)
II. The node mobility (from 10 to 50 m/s)

4.3.1 Impact of node population

The performance metrics are compared to evaluate the performance of the scheme by varying the node population. The nodes are assumed to move with a constant speed of 10 m/s.

1. *Protocol overhead*: The control messages exchanged for duplication detection and periodic maintenance are compared and the results are plotted in Fig. 7. The results show that the number of packets exchanged by FAACP is 70.16 % low when compared to the DAACP. This is because of the hierarchical flooding method used for DAD. Moreover since FAACP uses a hastier searching technique using filters, it avoids the retransmission of packets due to T_{ACK} timeouts. In SAAMAN the retransmitted messages due to T_{ACK} timeouts incurs 11.08 % high overhead packets when the node population is between 50 and 150. The overhead in FAACP scheme is similar to SAAMAN after 150 nodes. This is because of the address retrieval message packets used in FAACP scheme. From the results obtained, it can be concluded that the number of overhead packets exchanged by FAACP is considerably low when compared to the other schemes.

Fig. 7 Number of nodes versus number of overhead packets of the schemes

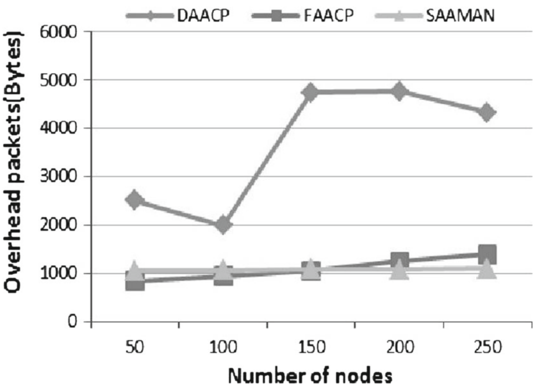
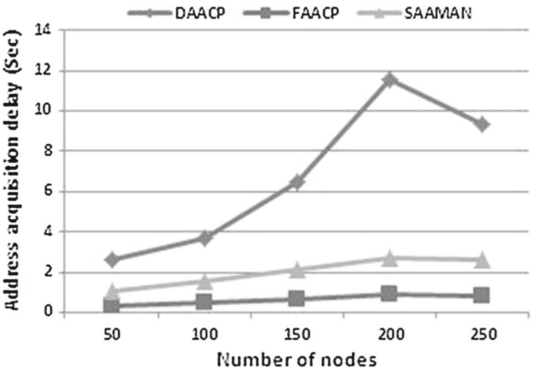


Fig. 8 Number of nodes versus address acquisition delay of the scheme



2. *Address acquisition delay*: The Fig. 8 shows that the impacts of the node population over address acquisition delay of the schemes. The address acquisition delay is highly dependent on the number of messages exchanged, the hop limit for message exchanges, number of address entries stored in information and the efficiency of duplicate detection. The results show that the DAACP scheme incurs 90.47 % more address acquisition delay, as it uses a conditional allocator selection (even in higher hop counts) and numerous control messages. The DAACP scheme chooses nodes with threshold address buffer level, as address allocating agents. If there are no neighbor nodes with threshold address buffer level, the address reclamation mechanism is revoked before allocation. In DAACP, the number of nodes undergoing address reclamations in 200 node scenario is more when compared to 250 node scenario. Hence the address acquisition delay is high for the same. The SAAMAN scheme shows a 67.63 % increase in address allocation delay when compared to FAACP scheme. This is because of the complex 32 bit linear search technique employed for duplicate detection. The address acquisition delay is very minimal in FAACP scheme, as it uses filters for enhanced duplication search and uses only few control messages for the working of the scheme.
3. *Successful allocation ratio*: The average duplicate address detection and successful allocations, increases with the increase in node population. The Fig. 9 shows

the comparison of the successful allocation ratio of FAACP with SAAMAN and DAACP.The filter used in the FAACP performs efficient updating of the UAVs thereby improving the efficiency of duplicate addresses detection. The hierarchical duplicate detection method used in FAACP also improves the reliability of the searching technique thereby assuring more successful allocations. Overall FAACP scheme shows 60.77 and 45.85 % high successful allocation ratio when compared to DAACP and SAAMAN respectively.

4. *Packet loss:* The packet losses of the schemes are analyzed and plotted in Fig. 10. The results show that the DAACP scheme suffers high packet loss when compared to the SAAMAN and FAACP. DAACP scheme proves to incur more address acquisition delay (Fig. 8) because of extra packets exchanges during address reclamations followed by address allocation. During the scenario with 200 nodes, the packet exchanges in DAACP scheme is high and hence causes more packet drops when compared to scenario with 250 nodes. The filters used in FAACP scheme increase the speed of duplication detection, therefore avoid the packet losses and retransmissions due to T_{ACK} timeouts. Moreover FAACP uses less number of control and maintenance messages, and hence the packet loss in FAACP scheme is 64.62 % less when compared to SAAMAN. To conclude the increase in node population does not have significant impact on packet losses in FAACP.

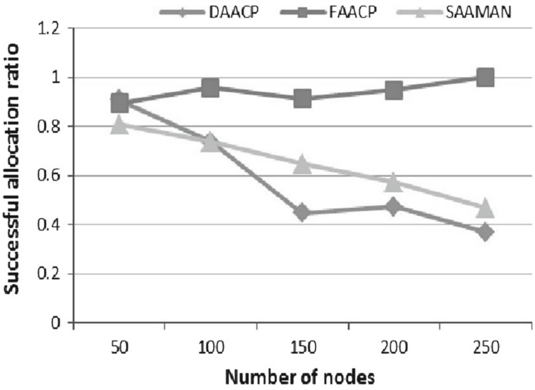


Fig. 9 Number of nodes versus successful allocation ratio of the schemes

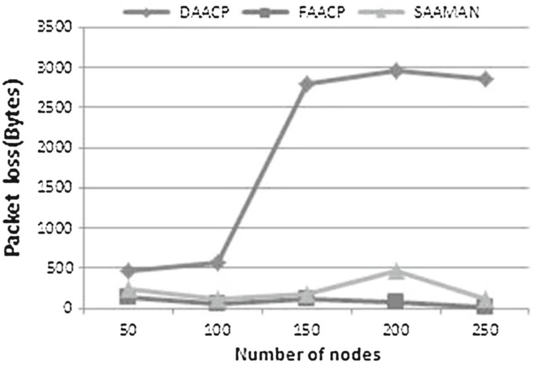


Fig. 10 Number of nodes versus packet loss of the schemes

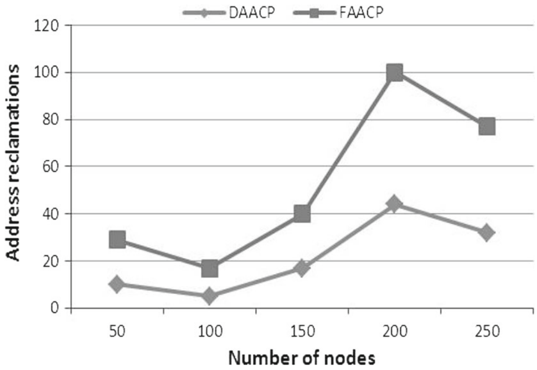


Fig. 11 Number of nodes versus address reclamations in the scheme

5. *Address reclamations*: The Fig. 11 shows the number of addresses reclaimed by the schemes with the increase in node population. The filter based scheme used in FAACP uses an efficient and simple address space management technique for retrieving lost address and address space memory. The DAACP uses an address reclaim process using specialized nodes called "agent" or "configuration nodes". In both the schemes address recovery mechanism is invoked when the address space in the nodes reaches a threshold level. The simulation at 250 nodes show less number of addresses reclaimed, as the scenario has only few nodes reaching the threshold level of address space. The address reclamations process in DAACP scheme causes more protocol overhead, packet losses and address acquisition delay. The results show that performance of the DAACP scheme is 56.49 % low when compared to the FAACP. The address reclamation process in FAACP scheme does not affect any other characteristic features of the protocol. The SAAMAN does not support reclaiming of addresses.

4.3.2 Impact of node mobility

The simulations are conducted by changing node mobility speed from 10 to 50 m/s. The results are analyzed to study the performance of the schemes and assure its applicability in Vehicular Networks (VANETs), in which the nodes mobility speed varies from 10 to 38.89 m/s [17]. The simulations are done with a node population of 50 nodes.

1. *Protocol overhead*: The overhead of the schemes increases as node mobility increases. The overhead is due to the message losses caused by unsuccessful packet deliveries and retransmissions. As FAACP, exchange less control packets, the overhead of the scheme is low compared to DAACP and SAAMAN schemes. The results of the impact of node mobility over the overhead packets of the schemes are plotted in Fig. 12. The FAACP scheme induces 55.07 and 26.88 % less overhead packets when compared to DAACP and SAAMAN schemes respectively. From the results obtained, it can be concluded that the increase in the node mobility causes slight increase in the number of overhead packets in FAACP scheme.
2. *Address acquisition delay:* The average address allocation delays of the schemes are plotted with varying node mobility in Fig. 13. The results show that the delay

increase with increase in node speed. The increase in delay is due to the packet losses caused by multi-hop transmissions, dynamic topological changes and congestion of routing control packets. Even at high node mobility, the efficient filters used in FAACP scheme reduces the response time taken for duplication detection and hence the address acquisition delay is 80.04 and 69.82 % less when compared to DAACP and SAAMAN schemes respectively.

3. *Successful allocation ratio:* The Fig. 14 shows the comparison of the average successful allocation ratio of the schemes at varying node mobility speed. The duplicate address detection and successful allocations, decreases with the increase in the node mobility speed. The transmission range issues, network merging and partitioning etc. imposed by node mobility results in link instability and unsuccessful packet deliveries. The hierarchical flooding and the filter used in FAACP scheme, handle uncertainties by improving the efficiency of search. So the successful allocations are 64.04 and 21.43 % higher in FAACP when compared to the DAACP and SAAMAN schemes.

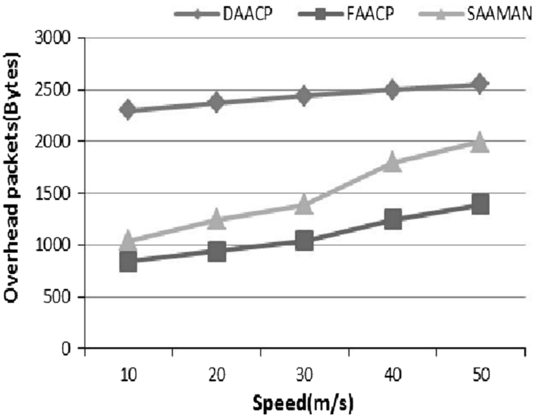


Fig. 12 Node mobility speed versus number of overhead packets of the schemes

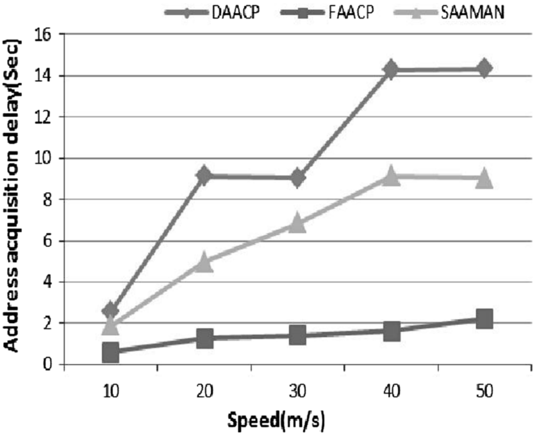


Fig. 13 Node mobility speed versus address acquisition delay of the schemes

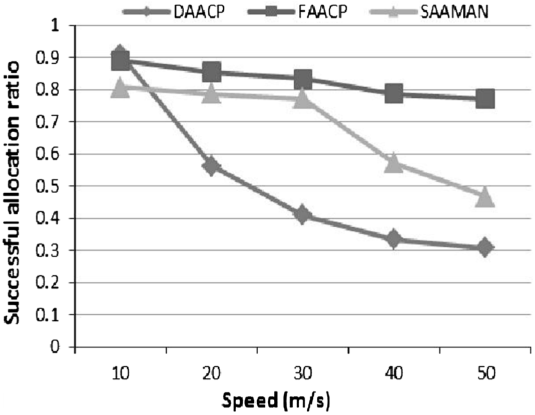


Fig. 14 Node mobility speed versus successful allocation ratio of the schemes

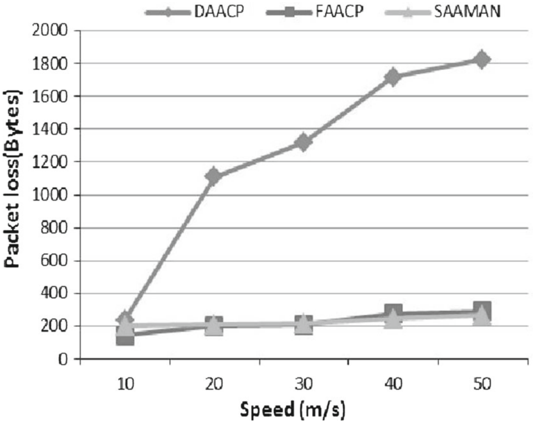


Fig. 15 Node mobility speed versus packet loss of the schemes

4. *Packet loss:* The packet loss of the schemes at varying node mobility are analyzed and plotted in Fig. 15. The hierarchical message flooding method with the help of UAVs, eases handling unsuccessful packet deliveries in FAACP. The filters used in FAACP reduce the search and updating time during mobility. Hence the total packet exchanges in FAACP are comparatively less, which resulted in minimal packet losses. The packet losses in FAACP scheme is 81.89 and 1.49 % less when compared to DAACP and SAAMAN schemes respectively. The results also conclude that the packet loss in DAACP scheme is very high when compared to FAACP and SAAMAN. This is because of the extra control and maintenance messages exchanges in the scheme for initializing new agent or configuring node for address allocation.

5 Conclusion and future works

In this paper, a filter based autoconfiguration scheme using a protocol called “FAACP” for Duplicate Address Detection and Recovery in MANETs is proposed. Initially, the network topology is defined as a grid structure and every node use a geographic forwarding scheme to collect information about its position in the grid structure. By exploiting three rules, the scheme specialize few nodes as Unique IP Address Verification Agents (UAV agents). These nodes are responsible for allocating and maintaining conflict-free IP addresses in the network. The technique exploits sequence filter to represent IP addresses in a compacted manner which saves the memory required for address space allocation and improves the efficiency of duplication detection. Further, the address retrieval mechanism used in the scheme recovers both lost IP addresses and memory spaces simultaneously.

Two existing schemes DAACP [8] and SAAMAN [11] are compared to study the performance of the proposed scheme. The mathematical formulations and values derived for memory usage conclude that FAACP scheme consumes less memory when compared to other schemes. The simulation results prove that the scheme has comparatively better successful allocation ratio and less address acquisition delay compared to other schemes. The overhead of control packets is significantly low in the scheme. The simulation results also show that FAACP scheme performs well in high node mobility; hence the scheme can also be used in VANET applications. The address retrieval mechanism for acquiring lost address space management reduces the address leakage issues and ensures scalability. The security of the autoconfiguration scheme is a highlighted issue which is not addressed in the FAACP. The scheme proposed in [18,19] can be extended as future works to mitigate the selfish and malicious node attacks.

Acknowledgments The authors are grateful to the Anna Centenary Research Fellowship (ACRF) Grant, for the support to carry out this research project. The authors also thank the reviewers and editors for their valuable suggestions and recommendations.

References

1. Perkins CE, Malinen JT, Wakikawa R, Belding-Royer EM, Sun Y (2001) IP address autoconfiguration for Ad Hoc Networks. Internet Draft draft-perkins-manet-autoconf-01, Internet Engineering Task Force. http://www.tools.ietf.org/html/draft-perkins-manet-autoconf-01
2. Vaidya NH (2002) Weak duplicate address detection in mobile Ad Hoc networks. In proceedings of the 3rd international symposium on mobile Ad Hoc networking & computing, Lausanne, Switzerland, pp 206–216. doi:10.1145/513800.513826
3. Cheshire S, Aboba B, Guttman E (2005) Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927, Internet Engineering Task Force. http://www.ietf.org/rfc/rfc3927
4. Park I, Kang N, Song HY (2007) Address Autoconfiguration for Hybrid Mobile Ad Hoc Networks. Internet-Draft, MANET Autoconfiguration (AUTOCONF). http://www.tools.ietf.org/html/draft-ikpark-autoconf-haa-03
5. Thomson S, Narten T, Jinmei T (2007) IPv6 Stateless Address Autoconfiguration. RFC 4862, Internet Engineering Task Force. http://www.ietf.org/rfc/rfc4862
6. Weniger K, Zitterbart M (2002) IPv6 autoconfiguration in large scale Mobile Ad-Hoc Networks. Proceedings of the European Wireless Conference. Florence, Italy, pp 142–148
7. Weniger K (2005) PACMAN: passive auto configuration for mobile ad hoc networks. IEEE Sel Areas Commun 23(3):507–519
8. Gammar SM, Amine E, Kamoun F (2009) Distributed address auto configuration protocol for Manet networks. Telecommun Syst 44(1–2):39–48
9. Villalba LJG, Matesanz JG, Orozco ALS, Díaz JDM (2011) Distributed dynamic host configuration protocol (D2HCP). Sensors 11:4438–4461

10. Villalba LJG, Matesanz JG, Orozco ALS, Díaz JDM (2013) E-D2HCP: enhanced distributed dynamic host configuration protocol. Computing. doi:10.1007/s00607-013-0307-3
11. Hussain SR, Saha S, Rahman A (2010) SAAMAN: scalable address autoconfiguration in mobile ad hoc networks. J Netw Syst Manag 19(3(2011)):394–426. doi:10.1007/s10922-010-9187-4
12. Fernandes NC, Moreira MDD, Duarte OCMB (2013) An efficient and robust addressing protocol for node autoconfiguration in ad hoc networks. IEEE/ACM Trans Netw 21:845–856
13. Grajzer M, Żernicki T, Głąbowski M (2013) ND++-an extended IPv6 Neighbor Discovery protocol for enhanced stateless address autoconfiguration in MANETs. Int J Commun Syst. doi:10.1002/dac.2472
14. Nazeeruddin M, Parr G, Scotney B (2006) DHAPM: a new host auto-configuration protocol for highly dynamic MANETs. J Netw Syst Manag 14(3):441–475
15. Network Simulator: http://www.isi.edu/nsnam/ns. (Accessed 30 June 2013)
16. Clausen T, Dearlove C, Dean J, Adjih C (2009) Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format. RFC 5444. Internet Engineering Task Force. http://www.tools.ietf.org/html/rfc5444
17. IEEE Trial-Use standards (2006) Wireless Access in Vehicular Environment—Security services for application and management messages, IEEE 1609.2-2006
18. Reshmi TR, Murugan K (2013) Application of fuzzy sets for Isolating selfish nodes by trust evaluation during auto-configuration and service establishment in MANETs. J Netw Innov Comput 1:65–73
19. Cavalli A, Orset J-M (2005) Secure hosts auto-configuration in mobile ad hoc networks. Ad Hoc Netw 3:656–667