IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 8, Issue.12, December 2020 E-ISSN: 2347-2693

DADAR: Duplicate Address Detection using ARP in Mobile Ad Hoc Network (MANET)

K. Victor Rajan[1*], V. Rhymend Uthariaraj[2]

[1,2]Dept. of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai, India

[*]*Corresponding Author: victor@jts.co.in, Tel.: +91-9884034642*

DOI: https://doi.org/10.26438/ijcse/v8i12.1520 | Available online at: www.ijcseonline.org

Received: 15/Dec/2020, Accepted: 24/Dec/2020, Published: 31/Dec/2020

***Abstract*—** Mobile Ad Hoc Network (MANET) is infrastructure free, highly dynamic wireless network, where central administration or configuration by an administrator is very difficult. In contrast to infrastructure based network, MANET supports autonomous and spontaneous networking and thus should be capable of self-organization and configuration. This self-configuring network allows the nodes to automatically configure addresses and routes based on ongoing traffic. Due to the distributed and dynamic nature of MANETs, centralized servers like DHCP cannot be used to assign IP address to nodes. In this paper we present a novel approach DADAR, for the efficient duplicate address detection and auto configuration of nodes in a MANET. Special features of DADAR are the support for frequent network partitioning and merging and very low protocol overhead. Nodes choose initial address using random numbers. Duplicate Address Detection (DAD) algorithm resolves the address conflict which can occur due to partitioning or merging. We are devising a novel approach to detect duplicate address using Address Resolution Protocol. ARP messages are sent by nodes to translate IP address to Hardware address. ARP packets contain the IP address and Hardware address pair of the nodes. These packets are analyzed in the background to identify duplicate addresses and conflicts are resolved. This approach does not introduce any new protocol or packet format.

Keywords **--** DADAR, ARP, Duplicate Address Detection, MANET

I. INTRODUCTION

Automatic address allocation is more difficult in a MANET environment than in wired networks due to instability of links, mobility of the nodes, the open nature of the mobile ad hoc networks and lack of central administration in the general case. Automatic address assignment procedures like DHCP cannot be applied to MANET since there is no centralized server. Thus new protocols have to be developed. A significant challenge to auto configuration protocols is the potentially frequent merging and partitioning of independently configured networks. As a consequence, address conflicts may occur and influence the routing of data packets. Researchers in the area of mobile ad hoc networks focus on efficient configuration and routing protocols. The proposed protocols can be classified into two approaches namely, proactive or table-driven and reactive or on-demand approaches. While the former continuously maintains routes to all nodes in the net-work, the latter only discovers routes when needed. Auto configuration algorithms have been developed using information gathered from ongoing routing protocol traffic. These algorithms are tightly coupled to a particular routing algorithm and will have to be modified whenever a new routing algorithm is brought into use. We propose an auto configuration algorithm which is independent of routing protocols. It uses the information available in Address Resolution Protocol and hence can be used in any network. The algorithm need not be modified even if the network or higher layer protocols change over time. The DADAR module can run independently in the background without affecting other activities of Network Layer.

The rest of the paper is organized as follows: an overview of issues and current research efforts in the area of address auto configuration of MANETs is given in section II. The most important components of our approach are described in the sections III, IV and V. To analyze the performance of DADAR, simulation experiments were conducted and results are discussed in section VI. Finally, section VII discusses the enhancements to DADAR and directions for future research.

II. RELATED WORK

Many efficient algorithms have been proposed for automatic assignment of address for mobile nodes. But, duplicate addresses occur due to the mobility of nodes and merger of partitions. So, efficient duplicate address detection is inevitable for MANETs. There are few approaches to detect duplicate address suggested by re-searchers in the past.

One approach is Weak DAD [1] in which a node detects a duplicate address with information added to the routing

protocol packets. This approach proposes use of unique key to be generated and the key to be tagged to the IP address during link state routing. The drawback of this approach is routing protocol packet has to be modified to carry additional field.

Another state-full approach to assign addresses without duplications is discussed in [2]. In this approach each node acts like a DHCP server maintaining a pool of addresses and helps new nodes to configure from this pool. This approach requires maintaining states (allocation table) and reliable global state synchronization. State synchronization and returning pool of addresses requires reliable broadcast and generates additional network traffic.

Another approach is to detect the duplicate address using the cross-layer information derived from ongoing routing protocol traffic [3]. Though PACMAN doesn't alter the packet structure, it depends on the routing protocols which may or may not be used everywhere. It requires analysis of multiple routing protocols. More than nine different algorithms are needed to cover all the identified scenarios. Supporting multiple network layer protocols is an overhead and the duplicate address detection algorithm has to be modified whenever a new routing protocol is brought into use.

The requirements of a good duplication address detection algorithm include but not limited to

- It should not add any protocol overhead by introducing new packet formats.
- It should not flood the network by sending frequent request/response packets
- It should not change over the time or depend on network protocols which may not exist everywhere.

By taking these constraints into consideration, we propose a novel approach which detects duplicate address from the ARP messages which are sent by nodes to discover the Hardware (H/W) address. Address Resolution Protocol is ubiquitous and the ARP packet format doesn't change over the period of time. Since we device a mechanism to detect duplicate address based on ARP, the algorithm will work even if other layer protocols undergo changes over the time.

III. AUTO CONFIGURATION USING DADAR

This MANET is a self-configuring/self-healing network that can operate with or without any centralized management. If operators have to plan and configure MANET nodes prior to deploying them, the promise of MANET is not fully realized. We propose a novel approach for configuring MANET nodes with minimum support, so that one can simply power-on a group of MANET nodes without any pre-configuration and have them configure themselves into an operational network. The auto configuration is done in two phases. The first phase consists of choosing an MANET Local Address (MLA) without any conflict with the existing nodes in the partition. The second phase is to proactively discover duplicate addresses which can occur due to merging or partitioning. The conflicts are resolved by reconfiguring one of the nodes with new address. We propose a modular framework which contains the following components for successful configuration and survival of a MANET node.

3.1. MLA Selector: This module of the auto configuration framework is responsible for acquiring a MANET-Local Address (MLA). MLA is an IP address intended to be used inside a single MANET partition. An MLA for a node must be unique within the MANET partition it belongs to. For this purpose, we propose a 16-bit MLA prefix (192.168) which is combined with a 16-bit random number to form a 32-bit IP address. After choosing an address, the node performs a DAD by broadcasting an Address Request (AREQ) message, which contains the chosen address. A node having the same address defends it by replying an Address Reply (AREP) message, which is sent over the reverse path established by the AREQ message. If there is no other node in the network with this address, a timer at the originator node expires and the address is considered unique. When there is no conflict, the selected address is assigned to the wireless interface of the MANET node. This module also defends it address by sending Address Reply (AREP) message, if it receives an Address Request (AREQ) message for its own address. Figure 1 shows the MLA selection procedure.

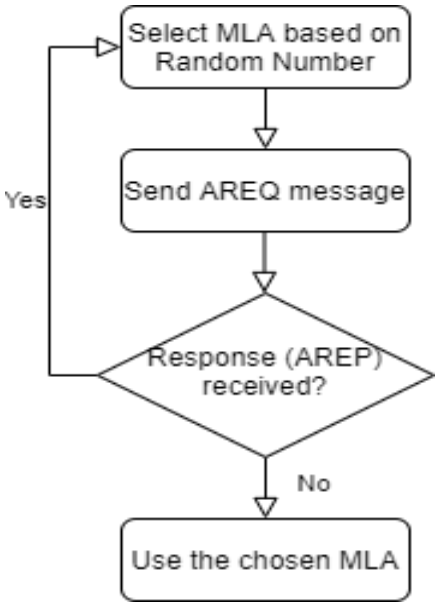

Figure 1 – MLA Selection Procedure

3.2. DADAR Manager: DADAR (Duplicate Address Detection using Address Resolution Protocol) Manager is the heart of this framework which is responsible for discovering duplicate addresses by analyzing the ARP messages in the background. It captures all the ARP request and response packets and applies the DADAR algorithms to detect the presence of duplicate address. The DADAR algorithms are discussed in section IV.

3.3. Conflict Resolver: This module resolves the conflict when duplicate address is detected. A node can either detect a conflict of its own address or another node's address. When a node detects conflict of its own address, no separate notification is required. It can choose a new address based on a conflict resolution procedure discussed in Section V. When a node detects conflict between two other nodes, one of the nodes should be notified so that it can change its address. Conflict Resolver performs these activities.

IV. DADAR

DADAR detects the presence of duplicate addresses in a MANET partition using Address Resolution Protocol. We explain the DADAR algorithms in this section. Ad-dress Resolution Protocol is used to translate the IP ad-dress to H/W address of nodes by sending broadcast messages. DADAR is a duplicate address detection approach which uses the ARP packets which are sent through the network. It uses the IP address and H/W address pair present in the ARP packets to detect conflicts. Each mobile node promiscuously captures all the ARP packets. These packets are sent to DADAR manager which performs the analysis in the background to detect conflict. A DADAR cache is created at each node. It is a table having IP address and H/W address pair. The DADAR cache is maintained as follows.

- When an ARP request arrives, the IP address, H/W ad-dress pair of the sender is extracted and DADAR cache is updated. Receiver H/W address is broadcast address and hence it is ignored. Since ARP requests are normally broadcasts, every node will get chance to collect this information evenly.
- When an ARP response arrives, the IP address, H/W address pairs of both sender and receiver are extracted and DADAR cache is updated. Since DADAR manager is capturing all ARP packets, DADAR cache is updated during response also.

We discuss two algorithms called DADAR algorithms below to detect address conflicts.

4.1 DADAR Algorithms

A node can detect the conflict of its own IP address or other nodes. We describe an algorithm DADAR-SELF which is used to detect conflict of self-address. The algorithm DADAR-ARBITRATOR is used to detect conflict of others addresses.

DADAR-SELF: This algorithm is used by every node to detect whether another node is using its IP address. It works as follows.

a)A node receives an ARP request. Target H/W ad-dress will be broadcast address. So, it ignores target address and extracts sender address. If the IP address matches with its own IP address and H/W ad-dress is different from its H/W address, then it is an indication that another node uses its IP address. Otherwise it adds the IP address, H/W address pair to DADAR cache.

b)A node receives an ARP response. It will have valid sender and target addresses. It examines the packet to see if its IP address is present in the packet. If the H/W address corresponding to its IP address is different from its H/W address, then it is an indication that another node uses its IP address. Otherwise it adds the IP address, H/W address pair to DADAR cache.

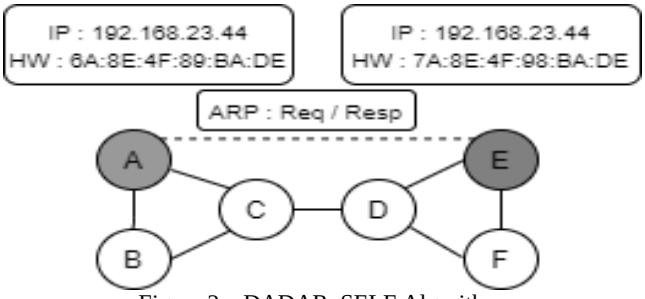

Figure 2 – DADAR SELF Algorithm

In the above shown example, two nodes A and E are having same IP address 192.168.23.44. When an ARP request or response from A reaches E, E finds out that its IP address is paired with a different H/W address. After detecting the conflict, one of the nodes will choose a new IP address.

DADAR-ARBITRATOR: A node can detect two other nodes using the same IP address. We call this technique as DADAR-ARBITRATOR algorithm. It performs the following check while updating the DADAR cache.

a) A node receives an ARP request. It extracts the IP address, H/W address pair of the sender. While adding this pair to DADAR cache, it finds out that an entry exists in DADAR cache with the same IP address but with different H/W address. Then it is an indication that two different nodes with same IP address are present in the MANET partition.

b) A node receives an ARP response. It extracts the IP address, H/W address pair of both sender and target. While adding the IP address, H/W address pairs to DADAR cache; it finds out that an entry exists in DADAR cache with the same IP address but with different H/W address. Then it is an indication that two different nodes with same IP address are present in the MANET.

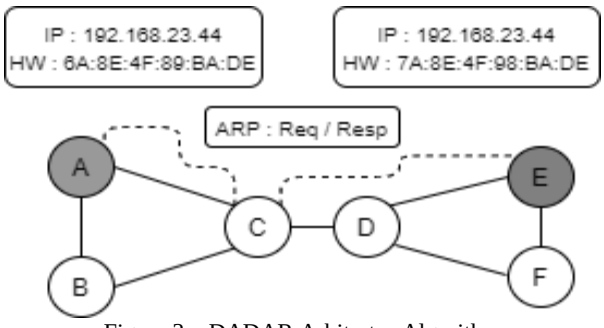

Figure 3 – DADAR Arbitrator Algorithm

In the above shown example, two nodes A and E are having same IP address 192.168.23.44 When an ARP request or response from A reaches C, it adds IP-H/W pair for 192.168.23.44 in its DADAR cache. Later when an ARP packet comes from E to C with the same IP address, the H/W address will be different. It then detects the conflict and notifies one of the nodes to resolve the conflict.

The above mentioned algorithms are implemented in a separate module in the Network layer. All ARP packets are captured by DADAR manager and it applies DADAR algorithms to detect conflict. Then we use the conflict resolution mechanism discussed in section V to resolve the conflict.

4.2 DADAR and Duplicate H/W Addresses

Two nodes may have same IP address as well as H/W address. Though it is possible to have H/W address conflict, it is highly unlikely to happen in real world. If two nodes have the same IP address and H/W address then such conflicts can be detected by DADAR as follows. When a node sends an ARP request/response, it starts a timer. If it receives the same ARP request/response within a given time period, it discards the packet. Otherwise it processes the packet, because it would have originated from the other node with the same IP address and H/W address. In this type of conflict the node should immediately change its IP address without applying the conflict resolution procedure discussed in section V.

A packet undergoes transmission delay in the wireless media and processing delay at each node. In our experiment, we set a timeout value of 10ms. If an ARP packet comes back within this time limit, then it would be discarded. Otherwise, it would be processed.

4.3 Proactive DADAR

DADAR algorithms are reactive. DADAR manager captures the ARP packets in background and performs analysis. In a highly active network, ARP packets will be flying around frequently and the probability of duplicate address detection will be high. If nodes are not communicating fast enough, the number of ARP packets sent out will be less. It may take longer time to detect a duplicate address. In such cases, we need to adopt a proactive approach to increase the probability of conflict detection. As per this proactive DADAR, if the number of ARP packets received in a given interval of time is less than a threshold, then it sends an ARP re-quest for its own IP address at regular intervals. This will increase the ARP traffic and enable faster detection of conflicts. Nodes can count the number of ARP packets flowing per minute in the network. If it is below a threshold value, then proactive DADAR can be turned on. If ARP packets are flowing above the threshold, proactive DADAR can be turned off.

4.4 DADAR Reliability

In a MANET partition with duplicate IP addresses, the conflict will be detected by DADAR algorithms, if the conflicting nodes exchange messages between them or others. We shall prove this using proof by induction.

i) When n = 1

Assume that there is only one node in a MANET partition. Then its IP address is unique within the partition and doesn't conflict with any other node. So, unique address is guaranteed.

ii) When n = 2

Assume that there are two nodes in a MANET partition and both of them are using the same IP address. When one of the nodes wants to communicate with other, it will send ARP request to get the H/W address of other node. When the second node receives the ARP request, it will immediately detect the conflict using DADAR-SELF (a) algorithm.

iii) Proof by Induction

Assume that the algorithm works for n nodes and we shall prove for (n+1) nodes. There are n nodes in a MANET partition each having unique address. A new node N(n+1) joins the partition whose IP conflicts with a node Nk ($1 \leq k \leq n$) . If the newly joined node doesn't communicate with any other node, then its conflict doesn't create any problem. But, when N(n+1) wants to communicate with any other node, it will send an ARP request to get the H/W address of its communicating partner. If this request is received by Nk, then using DADAR-SELF algorithm, it will detect the conflict. If the request doesn't reach Nk due to some reason, but reaches at least one node Ni in the partition, then Ni will detect the conflict using DADAR-ARBITRATOR algorithm and send conflict notification.

So, this algorithm works for (n+1) nodes when it is true for n. Since we have already proved for n = 1 and n = 2, by principle of induction, DADAR algorithm will detect the conflict without fail. The only requirement for the algorithm to be successful is the conflicting nodes should actively communicate within the partition.

V. CONFLICT RESOLUTION

A node can either detect a conflict of its own address or another node's address. In this section, we will see how conflicts can be resolved in these two scenarios.

A) Node detecting its own conflict: When a node detects conflict of its own address, no notification message needs to be sent to resolve the conflict. But, there must be a rule to resolve the conflict without any additional network traffic. We suggest a procedure based on H/W address. The conflict resolution procedure is, the node with smaller H/W address should change its address. This relieves us from the overhead of taking complicated decisions and sending notifications. The node which has detected the conflict knows its H/W address as well as the conflicting node's H/W address. It converts both the H/W addresses to 48-bit hexadecimal value (for e.g. 0x6A8E4FBADE). If its H/W

value is less than its counterpart's H/W value, then it chooses new IP address using the MLA selection procedure discussed in section III. Otherwise it will continue to use its IP address and other node will change its IP address. The following example illustrates this conflict resolution procedure.

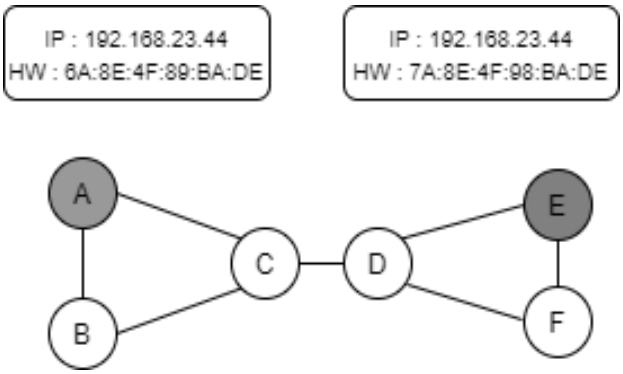

Figure 4 – Conflict Resolution based on H/W address

In the above example, since the H/W address of A is algebraically smaller than E, A will change it IP address.

B) Node detecting conflict of other nodes: When a node detects conflict of two other nodes, one of the nodes should be notified, so that it can change its address. The conflict resolution procedure is; the node with smaller H/W address should change its IP address. The arbitrating node which has detected the conflict informs this to the conflicting node. Introducing a new notification message will result in protocol overhead and new packet formats. So, we follow a simple approach by using ARP response message as follows. Arbitrator node sends a fake ARP response message to the target node as if it originated from the other conflicting node. Using the algorithm DADAR-SELF (b) that node will detect the conflict and change its address. The following example illustrates this procedure.

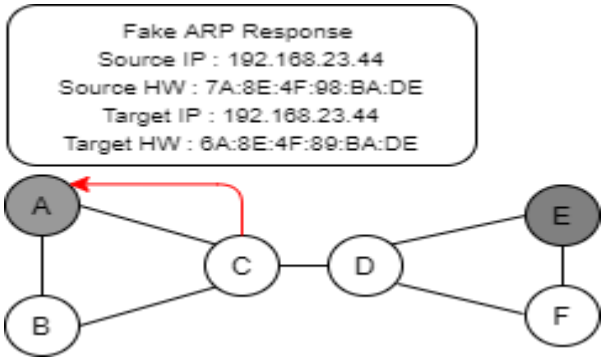

Figure 5 – Notification of Conflict using ARP Response

Assume that node A and E are using the same IP and node C detects the conflict. Node C should inform A about the conflict. Node C sends a fake ARP response to A with E's address as sender. Since A is promiscuously capturing all ARP packets, it will receive this packet. Using algorithm DADAR-SELF (b) node A will detect the conflict and change its address. Node E will continue to use its old IP address.

VI. EVALUATION AND TEST RESULTS

DADAR was implemented and evaluated in simulation environment using GlomoSim. GlomoSim does not send ARP messages as part of regular communication. ARP protocol was implemented and plugged into GlomoSim network layer. DADAR algorithms were implemented to capture the ARP messages and analyze them in the background. Two independent partitions were chosen and assigned unique IP addresses based on the MLA selection procedure discussed in section 3. IP addresses were unique within the partitions but not across the partitions. When the partitions merged, duplicate IP addresses resulted in conflict. DADAR manager captured the ARP packets exchanged between the nodes and resolved the conflict. When the number of nodes is less, the number of ARP packets exchanged per unit time interval will also be less. Thus it takes longer duration to detect a conflict. As the number of nodes increases more ARP packets are exchanged per unit time and conflicts can be detected faster. The following graph shows the relationship between number of nodes in a MANET partition and the time taken to detect a duplicate address.

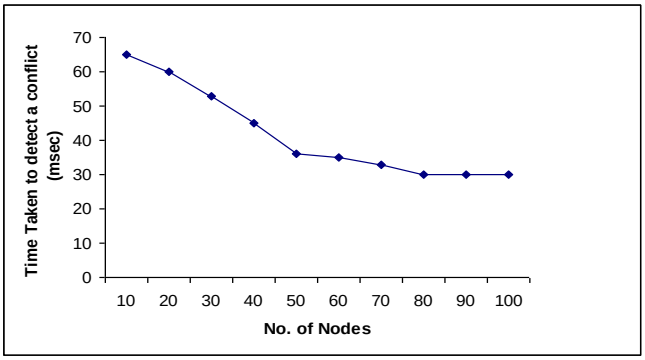

Figure 6 – Average time taken to detect conflict

From the graph we see that the time taken to detect a conflict reaches an optimum around 30 milliseconds if at least 50 nodes are actively communicating.

We shall discuss the various parameters affecting the conflict detection. Let 'p' be the probability that a node sends an ARP request/response at a given time interval, 't'. To detect a conflict the two conflicting nodes should send out at least one ARP packet each. The posterior probability for ARP packets to be transmitted from the two conflict nodes is given by (using Bayes' theorem)

$$P_c = (2 / n) * \binom{n}{2} p^2 (1-p)^{(n-2)} / p$$

$$P_c = (n-1) * p (1-p)^{(n-2)}$$

From this we observe that the conflict detection probability is directly proportional to the number of nodes n. When the number of actively communicating nodes increases, the conflict detection probability also increases and hence conflicts can be detected quickly. Thus the conflict detection time, 'tc' is inversely proportional to the average

number of ARP packets travelling in the partition. Other factors affecting the time taken are propagation delay and packet loss percentage. If μ is the propagation delay and λ, the percentage of packet loss, then the time taken to detect a conflict can be written as

$$tc = \mu + \lambda * (c / np)$$

where c is the constant of proportionality known as DADAR constant. Since μ and λ are characteristics of the network, the conflicting detection time can be reduced by increasing the average number of packets. n and p are the major factors influencing the average number of packets. As the value of np increases, the conflict can be detected quickly. But, it does not go below a certain value because of the propagation delay μ and the packet loss percentage λ. However conflicts can be detected quickly enough to prevent incorrect routing by having a higher value for the average number of ARP packets (np).

VII. CONCLUSION AND FUTURE WORK

The proposed algorithms will help to identify the presence of duplicate addresses in a MANET. These algorithms do not introduce any new packet format or a new protocol. Normal ARP packets are used to infer the presence of duplicate address. So, we believe that it is an efficient approach when compared the existing algorithms. Following are the problems which can be taken up for further analysis and research.

7.1 DADAR Enhancements

If a node doesn't have a H/W address assigned by the manufacturer, then it may not be able to use the DADAR algorithms. In such situations, the DADAR manager assigns a pseudo H/W address based on a random number. This address will be used in all ARP packets and DADAR algorithms can be applied to detect conflict.

DADAR algorithm relies on the ARP packets transmitted in a network. In a hostile environment, misbehaving nodes can send out wrong ARP packets leading to frequent re-configuration of IP addresses. Though it doesn't affect the communication between the nodes, it might degrade the performance. Improved measures can be taken to discard such misleading ARP packets.

REFERENCES

[1] N.H. Vaidya, "*Weak duplicate address detection in mobile ad hoc networks*" in the Proceedings of ACM MobiHoc 2002, Lausanne, Switzerland, June 2002

[2] M. Mohsin and R. Prakash, "*IP address assignment in a mobile ad hoc network*" in the Proceedings of MILCOM 2002, Anaheim, USA, Oct 2002

[3] K. Weniger, "*PACMAN: Passive Autoconfiguration for Mobile Ad Hoc Networks*", In the Proceedings of IEEE WCNC 2003, New Orleans – USA (March 2003).

[4] Boudjit S, Adjih C, Muehlethalar P, Laouiti A, "*Duplicate Address Detection and Autoconfiguration in OLSR*" in Journal of Universal Computer Science, vol. 13, no. 1(2007).

[5] S. Nesargi and R. Prakash, "*MANETConf: Configuration of Hosts in a Mobile Ad Hoc Network*", Infocom 2002.

[6] Seung Yi, Jeff Meegan and Jae H. Kim: "*Network auto Configuration for Mobile ad Hoc Networks*", In the Proceedings of IEEE 2007.

[7] David C. Plummer, "*Ethernet Address Resolution Protocol*". IETF RFC 826, November 1982**.**

[8] Thomas L., "*A Scheme to Eliminate Redundant Rebroadcast and Reduce Transmission Delay Using Binary Exponential Algorithm in Ad-Hoc Wireless Networks*", International Journal of Computer Sciences and Engineering, Vol.**3,** Issue.**8**, pp.**1-6**, **2017**.

[9] .GlomoSim: Global mobile information systems simulation library. [Online] http://pcl.cs.ucla.edu/projects/glomosim

[10] S. Tamilarasan, P.K. Sharma, "*A Survey on Dynamic Resource Allocation in MIMO Heterogeneous Cognitive Radio Networks based on Priority Scheduling*", International Journal of Computer Sciences and Engineering, Vol.**5,** No.**1**, pp.**53-59**, **2017**.

AUTHORS PROFILE

Mr. K. Victor Rajan completed Master of Engineering degree in Computer Science and Engineering from College of Engineering Guindy, Anna University. He has more than 20 years of experience in software development related to netowrk and mobile computing domains. Currently, he is working as Adjunct Faculty in Computer Centre, Indian Institute of Food Processing Technology, Thanjavur.. His main research work focuses on Mobile Computing, Artificial Intelligence and Machine Learning.

Dr.V. Rhymend Uthariaraj is Currently Professor and Director of Ramanujan Compting Centre, Anna Univerisity, Chennai. He completed his Ph.D in Computer Science and Engineering from College of Engineering Gundy, Anna Univesity. He has more than 150 research papers with 25 doctoral dissertations to credit. His main research work focuses on Computer Networks,Network Security,Pervasive Computing and Internet on Things. He has more than 35 years of Teaching and Research experience.

