Results Analysis of IP Address Auto-Configuration in Wireless Manets

S. Zahoor Ul Huq[1], S. Shabana Begum[2], Dr. K.E. Sreenivasa Murthy[3], Prof. B. Satyanaryana[4]

[1]Associate Professor, Dept. Of CSE, G. Pulla Reddy Engineering College(Autonomous)
Kurnool, Andhra Pradesh, India

s_zahoor_2000@yahoo.com

[2]Assistant Professor, Dept. Of CSE, Sri. Kottam Tulasi Reddy Memorial College of Engineering
Kondair, Andhra Pradesh, India

zahoor2saba@yahoo.com

[3]Princiapl, Sri. Kottam Tulasi Reddy Memorial College of Engineering
Kondair, Andhra Pradesh, India

kesmurthy@rediffmail.com

[4]Dept. Of MCA, Sri Krishnadevaraya University
Anantapur, Andhra Pradesh, India

bachalasatya@yahoo.com

***Abstract*—The main task of an address allocation protocol is to manage the address allocation to the nodes in the ad hoc MANETs. All routing protocols assume nodes to be configured a priori with a unique IP address. Allocating addresses to mobile nodes is a fundamental and difficult problem. A mobile device cannot participate in unicast communications until it is assigned a conflict-free IP address. So addressing in MANETs is of significant importance, and the address configuration process should be fast, as the algorithm must be able to select, allocate and assign a unique network address to the unconfigured node before with a unique IP address. Here we are providing two solutions for unique address assignment. One is by the using the broadcasting method (BrM), in which unique addresses are assigned, unique addresses are assigned with the cost of network load. This method works fine whenever a new ad hoc network has to be initiated and at a same time a group of nodes have to be configured with a unique IP addresses. But this method loads the network with much network traffic, when new nodes are to be joined. In order to overcome this we are using another approach which uses Modular Arithmetic (MoA). Modular Arithmetic with some modifications is used to generate the unique IP Addresses without loading the network. The proposed scheme is capable of assigning a unique IP address with low communication overhead, even addresses distribution and low latency when applied to large scale MANETs and even supports network merging and partitioning..**

Keyword-Address Allocation, Address Character, IP Auto-Configuration, Permanent Address

I. INTRODUCTION

A Mobile Ad hoc Network (MANET)[3] is an independent self organizing network in which each node functions as both an end host and a router. This form of wireless network is created by mobile nodes without any existing or fixed infrastructure. An Ad hoc network is a form of community network because it relies on the willingness of mobile hosts to forward and relay packets toward the destination. The formed network can be changed dynamically without the need of any system administrator. Ad hoc networks generally consist of hand held devices and laptop computers. These devices usually have limited transmission range, bandwidth and battery power.

The topology of a mobile Ad hoc network is typically highly dynamic because its nodes are free to move independently and randomly. The size of a MANET is constantly changing as nodes come in and out of the network range. A node is not part of a MANET until it is within the transmission range of an already configured node in the MANET. During the time a node is present in the MANET; it may or may not participate in communication or packet forwarding.

Nodes in the MANET need some form of identity before participating in any form of communication. Each end host in the MANET need to be uniquely addressed so that the packets can be relayed hop-by-hop and delivered ultimately to the destination. Routing protocols in MANET assume a priori that mobile nodes are configured with a valid (conflict free) IP address. Each node has a 48 bit MAC address at the link layer level. However, each end host needs some form of network address to successfully establish connection between two

end hosts. This network address will uniquely identify each node present in the network. Using traditional IP-based address assignment, such as DHCP [2], is not possible because nodes in the MANET are highly mobile and a central authority is not always reachable. Mobile IP [1] is also not a solution because MANET nodes do not stay connected to a wired network all the time. Addressing thus becomes significant in Ad hoc wireless networks due to the absence of any centralized coordinator. An address allocation protocol is required to enable dynamic address assignment to all nodes in a MANET.

Ad hoc networks have been used in military operations, shopping malls, disaster relief operations, conference rooms and peer to peer networks. Extensive research has been done on MANETs but still many problems and challenges remain unsolved. In MANETs, one must consider scalability limitations, communication overhead, bandwidth constraints, routing protocols, address assignment, power consumption, security concerns, and quality of service (QoS) mechanisms.

We have used two methods for IP Auto-Configuration in Mobile Ad Hoc Networks. One method is **Broadcasting Method** (BrM) and the other method is **Modular Arithmetic** (MoA). In this paper we are Comparing the BrM, MoA with the **Filter-Based Addressing Protocol** (FAP) [4].

II. BROADCASTING

A. IP Address Assignment

Communication in between the nodes is necessary to exchange and share the information. For the nodes to communicate with one another they should be in the network. In order to uniquely identify the nodes in the network they must be assigned with unique IP. If the IP addresses are not unique misrouting takes place and the information cannot be delivered or sent to the right receiver. So the nodes must be configured with the non duplicate IP address. This can be done by broadcasting method. This is one of the method to implement the IP Auto-Configuration to the MANETs. So far we have seen what is meant by MANETs. In the MANETs there is no fixed infrastructure or the DHCP present to configure the nodes, the nodes have to configure on their own. The broadcasting method is used when we want to assign the group of nodes together at the same time. This method makes use of broadcasting. In this method the MAC_IP table is maintained, which contains the information regarding the MAC addresses of the nodes, the random number generated for that node, and the new IP address that has been generated.

B. Ad Hoc IP Address Auto-Configuration

Whenever a new node enters into a network, Hello messages are broadcasted to the neighbors to know who the neighbors are. As shown in the Fig.1 node 4 and node 8 are the new nodes and they are sending the broadcasts telling their presence and also to know its surrounding neighbors. In the similar manner the neighbor also does the same thing, exchange the hello messages with the neighbors to know who the neighbors are and also to tell their existence. In this manner within no time each other nodes know their existence by broadcasting.

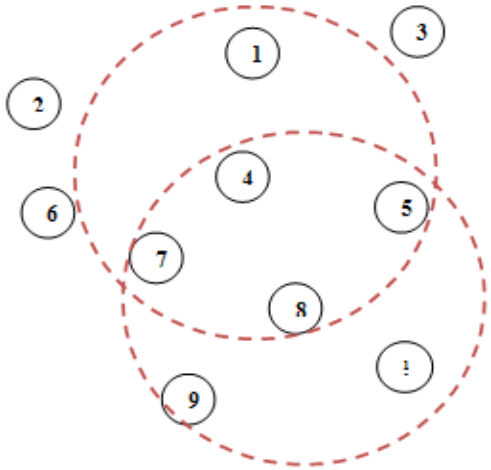

Fig. 1: Broadcasting of the hello packets by the nodes

The nodes have the MAC address assigned to them. These MAC address are 48 bits and are of the format e2-fe-2e-92-3c-39 which has six segments separated by 5 hyphens. The MAC address is associated with almost all

the nodes. In our case the nodes can also generate the Nonce. These Nonce are the random numbers generated by the nodes and are exchanged with the hello packets.

In our case as soon as some of the nodes become active that is when the network is initiated, the nodes present in the network send the hello messages to the neighbors to find out who the neighbors are and also to tell their existence. The nodes generally have the Mac addresses. Embed in the hello packets the nodes send their Mac Addresses and the nonce. Nonces are the random numbers generated by the nodes only once.

This information about the neighbors is stored in the MAC_IP table which is of the format shown in Table 1 and further forwarded to the next neighbor through the hello packets. In this way the information regarding the Mac addresses and Nonce of all the nodes spread across the network are maintained in all the nodes. Thus the MAC_IP table contains the information regarding the MAC-Address and Nonce of all the mobile nodes.

Table 1: Format of the MAC IP table

Mac - Address	Nonce	IP-Address

The MAC Addresses are sorted based on the first 8 bits and then on the next 8 bits and then so on and then finally on the Nonce. Though in some of the mobile devices MAC address does not exists then also they are sorted based on the Nonce. These sorted entries are given the IP addresses in an incremental order. Thus each and every node will be maintaining the MAC_IP table which consists of the MAC Address, Nonce and IP address of all the nodes.

If any of the nodes or set of the nodes are not within the range of the broadcast from any of the nodes which are broadcasting the hello messages, then for such node or set of nodes a separate network is formed. Later due to the mobility of the nodes if they come closer to the already existing network, then both the networks are merged explained later.

As already mentioned before, devices now a days have higher storage capacity, more computational power, and greater wireless communication capabilities, storage of 1 row in the MAC_IP table takes 12 bytes storage space, storage of 150 rows may take a maximum of 2 KB.

Once the MAC_IP table is build, the new nodes which want to enter into the network can just know from its neighbors, the max IP present in the MAC_IP table, maintained by the periodic broadcasts. The new node assigns itself with the next IP, makes an entry in the MAC_IP table and stores the MAC_IP with itself and inform its neighbors about the change. If two nodes enter into the network at the same time then such a situation needs to be handled differently.

III. MODULAR ARITHMETIC

We are using here the Address Characters to denote the IP Addresses that are to be assigned to the nodes. The procedure find the Address Characters (AC) is given below, we take two primes, p and q and compute N. In Order to find the Address Characters we can use the Euler Criterion which has the computational complexity of $O(\log p)^3$. As well we can use the Jacobi Symbols which can compute the Address Characters with the computational complexity of $O(1ogp)^2$ which is less than the Euler criterion of $O(\log p)^3$. The Jacobi symbol of N is,

$$Jacobi\ (x, N) = Jacobi(x, p)\ X\ Jacobi(x, q) \quad \ldots\ldots\ldots (7.1)$$

If the Jacobi symbol of $(x, N) = 1$, then x can be either a Address Character or a pseudo square (PS). If it is a Address Character then the Jacobi symbol of $(x, p) = 1$ and $(x, q) = 1$. If it is pseudo square then Jacobi symbol of $(x, p) = -1$ and $(x, q) = -1$. We only need to check the Jacobi symbol for N and p. We implemented our algorithm in NS-2.

In our algorithm Fig. 2 if x is a AC and is not in the list LIST, then we take it as a start and compute the sequence of ACs until we hit the last element before the start repeats again in the cycle. At that time the cycle is finished. List LIST_AC keeps track of AC in the current cycle along with its period (length of cycle) and list LIST contains all the ACs modulo N.

```
ACCycles(n::integer, p::integer, q::integer)
    LIST:={ }; x;
    for i from 1 to n-1 do
          LIST_ACC := { }: x:= I;
          if (Jacobi( x, n) = 1 and Jacobi(x, p) = 1) then
               While not member (x, LIST) do
                    LIST_AC := LIST_AC union x;
                    LIST := LIST union LIST_AC;
                    x := x^2 mod n;
                od;
                Period := nops (C);
                If (nops(LIST) >= 1) then print(Period); fi;
          fi;
     od;
end;
```

Fig: 2 Address Character Algorithm

In our first experiment, we ran a test for our clustering approach by choosing two safe primes p and q of 12 bits each, and then computing $\varphi(N)$ ($\varphi(N)$ = (p-1) * (q-1)) and N before computing the AC cycles to generate addresses. In the first experiment we chose

P = 2207
Q = 3467
N = p x q = 76,51,669
$\varphi(N)$ = 76,45,996

Table 2: Safe Prime Experiment Result

Cycle Length	Number of Cycles	Number of AC
1	1	1
29	38	1,102
1,732	1	1,732
50,228	38	19,08,664

We get 38 long cycles of length 50,228 each. This means we can have 38 clusters and each cluster can configure 50,228 nodes. Using these long cycles we can configure close to two million nodes. Note that 99.85 percent of the ACs are in the long cycles. As the number of addresses we will get from these long cycles are 19,08,664. A cycle length of 50,228 is long enough to ensure that when the number repeats again, the node which was assigned this address previously would have left the MANET.

If some clusters get partitioned from the original MANET, they can still assign unique addresses to newly joining nodes and if these partitions merge back together there would be no duplication. As N is 24 bits long, we can fix the 8 – bit prefix for the network address. When a node hears a message from a node with a different prefix, it assumes a network merger has occurred and runs a network merging algorithm. Safe primes here were 12 bits each but we could choose bigger safe primes of 16 bits each and get an N of 32 bits. In that case, we would be able to configure billions of nodes and we can piggyback the NID in the hello messages generated by the first node in the MANET.

In another experiment, we chose two doubly safe primes instead of just safe primes to see how big a cycle (address block) we can get. We noticed that the length of the cycles we get is huge. This results in fewer distinct cycles (address blocks) because the length of a cycle is extremely big. The two doubly safe primes are 13 bits each:

P = 4799
Q = 4919

Table 3: Double Safe Prime Experiment Result

Cycle Length	Number of Cycles	Number of AC
1	1	1
1,199	2	2,398
2,458	1	2,458
29,47,142	2	58,94,284

We get two long cycles of length 29,47,142 each. This ensures that the interval before a number (address) repeats again is quite large. When an address in a cycle repeats again the node which was assigned this address previously would have left the MANET. Hence, we avoid duplication and reclaim address automatically.

IV. COMPARING WITH THE BENCHMARK ALGORITHM

As the results of the above algorithm are conducted with various terrain ranges using NS-2 simulator, the comparison is made with the Benchmark algorithm "An Efficient Filter Based Addressing Protocol for Auto-configuration of Mobile Ad hoc Networks" [4] was implemented taking the network topology as an area of 670x670(m^2). Each node has a transmission range of 250(m) and the propagation model is *Two Ray Ground.* Propagation delay is set to 200(ms). The performances of these schemes were evaluated in terms of address allocation time (AAT) and the control overhead (CO) incurred. The AAT is defined as the time taken from address request initiation till the time when a new address is successfully acquired. We also define the address allocation ratio as the number of IP addresses in use versus the total number of usable/available addresses.

In the first scenario the incoming node arrives every 2 seconds. Therefore only one node arrives in the network in each interval of 2 seconds. In a interval, a node arrival is uniformly distributed. Also, the "class" IP address range is used when computing the address allocation ratio. An IP address pool has a size of 65,535 and this simulation was finished when the address allocation ratio became close to one.

In the second scenario the address request arrives at every 0.5 seconds i.e. the node request arrival rate is 0.5 requests per second. The number of requests increases within each time short time period. The short period is set to (10 X propagation delay). With an IP address pool of 20,000 the probability that identical IP address is selected at the nearly same time is very negligible in the MoA.

A. *Simulation Results*

1) Scenario One: Fig. 3(a) shows the address allocation time according to the address allocation ratio when a new node arrives in the network. For FAP, assuming sufficient IP addresses are available, it takes on average of 1.6s for a new node to acquire its IP address. In the case of high address usage case, AAT increases due to address conflicts. For MoA it takes an average of 1.3s for a new node to acquire its IP address.

As can be seen from Fig. 3(a), the variation of AAT in MoA is larger than in FAP. This is because of the node arrival scenario. In each interval, one node arrives with a uniform distribution. In FAP, a new node floods a message of its IP address to the entire network. In MoA, however, a new node waits-for a hello message for some time period to determine the IP to be assigned and does not flood the message. This time for a node to flood the message is saved, so there is some time variation in the AAT of MoA.

Fig. 3(b) shows the control message overhead required for a new node to acquire its IP address under varying address allocation ratio in terms of the number of control messages. FAP needs many control messages when most IP addresses are already in used. This is because the number of retries increases due to address conflict because of the false positives. For MoA, there is no address conflict and, consequently, no retries are needed. Hence the required message overhead in MoA is constant. MoA generates less message overhead than FAP. The overhead generated by the MoA is the start Sequence generated and distributing it to the allocators. This also has implications for power saving during the MoA address acquisition phase.

Fig. 3(a) reveals a weak point in FAP, compared to MoA. For the scenario in which one node arrives in one interval, the variation in address allocation ratio does not generate any address conflict. So MoA is not affected by the address usage ratio

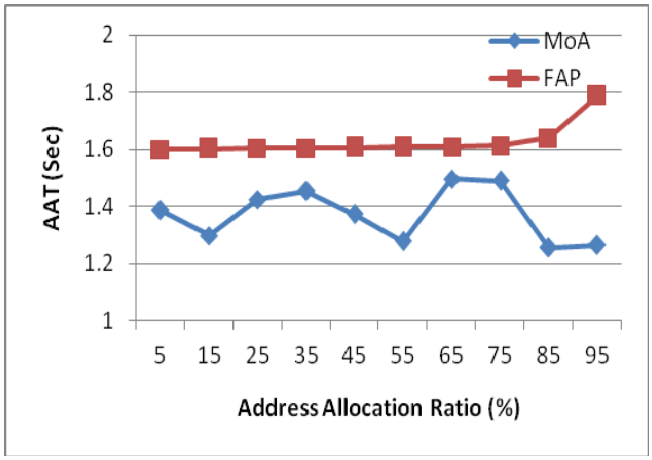

Fig. 3(a): **Scenario One -** Address allocation time wrt. Address

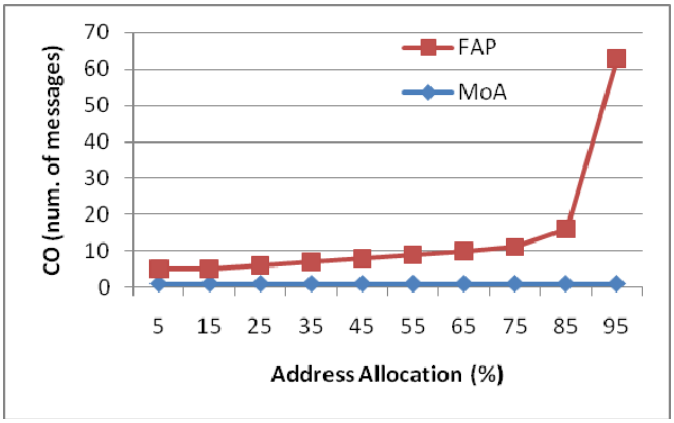

Fig. 3(b): **Scenario One -** Address Allocation Ratio (%)

2) *Scenario Two:* Fig. 4(a) shows the ATT according to the number of address requests generated over a short time period. In FAP, suppose there are enough IP addresses. In this case, although several nodes request IP addresses within an interval, there are no address conflicts. In MoA, IP addresses are allocated using the Start Sequence within an interval. Hence, if several nodes request IP address within an interval, address collision will not occur and the AAT will not increase. But whereas in the BrM if several nodes request IP addresses within an interval, address collision will occur and the AAT will increase. Note that there is huge difference between the AATs of MoA and BrM, as can be seen in Fig 4(a).

Fig. 4(b) shows the control message overhead according to the number of address requests generated over a short time period. With the same reasoning, FAP has constant message overhead. However, BrM has more overhead than MoA. For BrM, if N address requests occur in a interval, only one node can acquire its IP address and N – 1 other nodes will contend with one another during the next interval. However, in MoA, new N nodes will collect control messages in the first interval and the IP addresses are given to the nodes based on their arrival. Hence, MoA reduces the number of contending nodes in each interval to one, as compared to the N – 1 nodes in BrM.

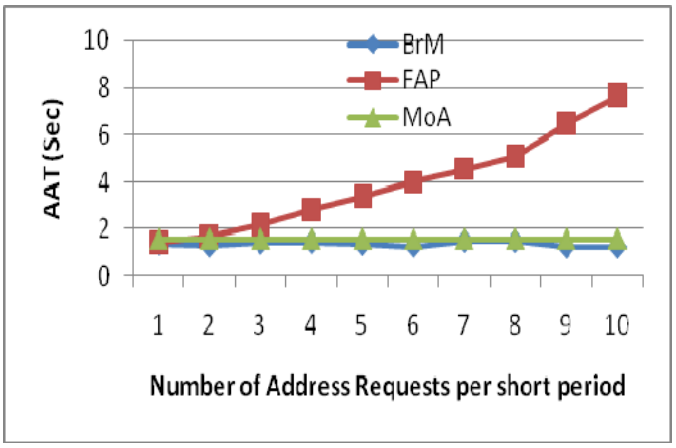

Fig. 4(a): **Scenario Two** - Address allocation time wrt. Number of address requests

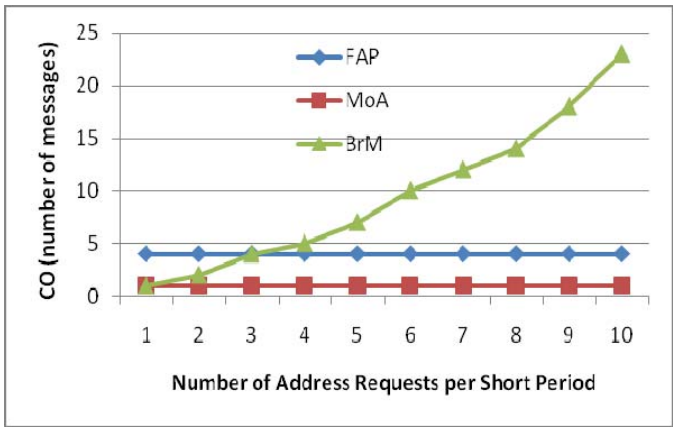

Fig. 4(b): **Scenario Two –** Control overhead wrt. Number of address requests

Fig. 4 reveals the weak points of BrM, compared to MoA and FAP. In the scenario in which several nodes arrive within one interval, the increased number of address requests does not impact MoA, because the probability that address conflict occurs is very low, if enough unused IP addresses are available to choose from, For BrM, although has more overhead than FAP, but BrM can be used to assign more number of IP addresses at the same time than that of the FAP.

In the next sequence of the experiments we have considered a rectangular space with 49 nodes. All nodes join the network at the same time with a different seed for each node. The initial address set was 8 collisions and, at the end of the simulation, all protocols resolved all the collisions. The total load in bytes of the address auto-configuration protocols are on Fig. MoA has shortest load with zero transmission of flooding messages, because MoA need not notify in the network regarding the address configuration. FAP on the other hand has more control load compared to that and MoA as the information of the IP address configuration is been flooded to the neighbors. BrM has more control load as it has to broadcast the MAC_IP table present in the hello packets every often to its neighbors to exchange the information regarding the neighbors.

The next analysis is related to the impact of the number of nodes in the network. We used again a rectangular space, simulating a community network, composed of static nodes.

3) *Scenario Three:* Mobility effect is analyzed in a scenario of 50 x 50m and 49 nodes, to evaluate the impact of mobility in the protocols without the influence of the partitions merging. Each node in our simulation moves according to the random-way point model with a pause time of 1sec. We suppose two transmissions of each flooding message, to reduce the effects of message losses. The nodes randomly choose the time in the network among 1 and 20s, to simulate the network. None of the protocols suffered with mobility. The difference between FAP, BrM and MoA control load is due to the use of Hellos and to the joining node procedures. BrM has more load campared to that of FAP and MoA as the MAC_IP table has to be exchanged in every broadcast. Therefore the use of Modular Arithmetic method has great impact on reducing the control load.

The last analysis is the performance on partition merging of the three protocols. We vary the number of partitions merging in a static scenario composed of 50 nodes by activating/deactivating some nodes responsible for connecting isolated partitions. The nodes on each isolated partition are activated simultaneously and the nodes that interconnect the partitions are active in fixed times. Fig. 5(a) shows the number of address collisions after the end of the simulations for one partition collision after the end of the simulation for one partition merging. BrM, as expected, does not resolve any collisions, because this protocol was not designed to control partitions merging, and, consequently, cannot control collisions created by the partition merging. FAP has no address collision after all runs.

Fig. 5(b) depicts the control load of the protocols during partition merging procedures with each flooding message being transmitted. The control load of the network initialization is not considered in this analysis. In BrM the smaller NID is chosen to be the new NID of the merged method. All conflicted addresses are checked and if two nodes hold the same IP address, then one of these nodes has to give up its address and acquire a new IP address. This process is repeated for all duplicated addresses until there are no remaining duplicates. We can see here that the number of partitions will impact the control load of FAP. In FAP the number of messages increases with the number of partitions, but these messages are sent in unicast and do not impact the control load.

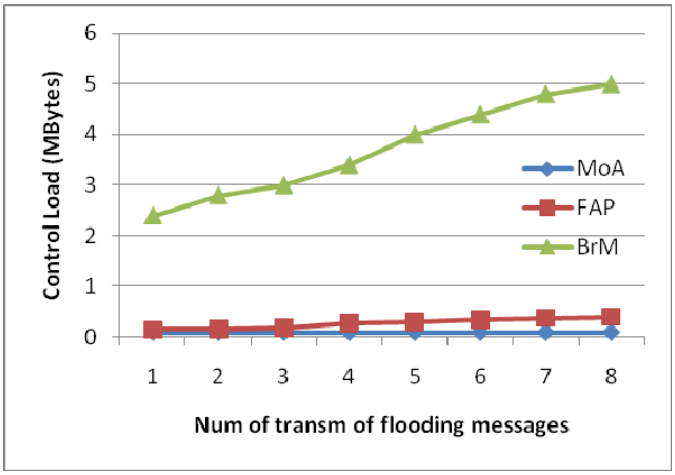

Fig. 5(a): Initialization control load with 49 nodes and simultaneous access

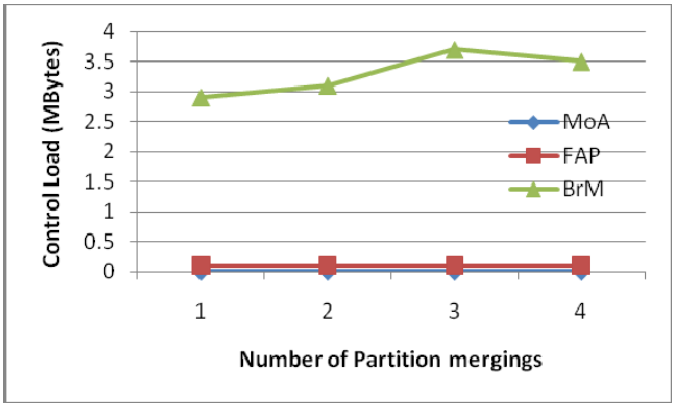

Fig. 5(b): Control load with 50 nodes on partition merging and two transmissions of flooding messages

V. Conclusion

In this thesis, we presented two IP Address auto-configuration schemes for use in a mobile ad hoc network; **Broadcasting Method** (BrM) and **Modular Arithmetic** (MoA). Basically, both schemes try to acquire a IP address, BrM is based on linear fashion to assign the IP addresses and MoA uses the Sequence Seed to assign the IP addresses.

If the size of the IP address pool is sufficient, MoA can assign IP addresses to nodes without any address conflicts arising, but not all of the IP addresses in the pool are available to the nodes. The Address Characters

(ACs) can be used to assign the IP Addresses and the Non-Address Character (NAC) cannot be used to assign the IP addresses. On the other hand, BrM can assign IP addresses to nodes without any waste of IP addresses from the IP address pool. However, in the case of the arrival of burst nodes, BrM, is associated with a longer address allocation time and more control overhead. MoA uses the reduced control overhead, without increasing the address allocation time compared to BrM.

We evaluated the performance of FAP, BrM and MoA for Various address allocation ratios and node arrival scenarios via simulation performed on NS-2. The simulation results clearly support our claims regarding the performance of these different schemes.

REFERENCES

[1] Perkins, IP mobility support. In Internet Engineering Task Force (IETF), RFC 2002, Oct 1996.
[2] R. Droms, Dynamic host configuration protocol, Internet Engineering Task Force (IETF), RFC 2131, March 1997.
[3] Mobile Ad-hoc Networks (MANET), www.ietf.org/html.charters/manet-charter.hml.
[4] Natalia Castro Fernandes, Marcelo Duffles Donato Moreira, and Otto Carlos Muniz Bandeira Duarte, "An Efficient Filter-Based Addressing Protocol For Autoconfiguration of Mobile Ad Hoc Networks," Proc. IEEE INFOCOM 2009

S.Zahoor ul Huq obtained his M.E in Computer Science & Engineering from Anna University in 2004. He is currently pursuing PhD degree from Sri Krishna Devaraya University, Anantapur, India,. He presented more than 12 research papers in various international journals. He is presently working as Associate Professor in CSE Department, G. Pulla Reddy Engineering College, Kurnool, A.P, India. His research interests include Computer Networks , Databases and data mining.

S. Shabana Begum obtained her M.Tech. in Computer Science & Engineering from Anna University in 2010. She is currently pursuing PhD degree from Rayalaseema University, Kurnool, Andhra Pradesh, India,. She is presently working as Assistant Professor in CSE Department, Sri Kottam Tulasi Reddy Memorial Engineering College, Kondair, A.P, India. Her research interests include Object Oriented Programming, Computer Networks, Databases..

Dr. K.E. Sreenivasa Murthy obtained B.Tech and M.Tech degrees in Electronics and Communication Engineering from Sri Venkateswara University, Tirupati, India in 1989 and 1992 respectively and PhD degree from Sri Krishna Devaraya University, Anantapur, India, in 1997. He presented more than 10 research papers in various national and international conferences and journals. He is at present working as principal at Sri Kottam Tulasi Reddy Memorial Engineering College, Kondair, India. His research interests include FPGA and DSP applications.

Dr. B. Sathyanarayana *graduated* from Madras Christian College, Madras University, Tamilnadu India in1985 and post graduated from Madurai Kamaraju University, Tamilnadu in 1988 and obtained PhD from Sri Krishna Devaraya University Ananthapur India in 2000. He worked as Head of the Departement in Department of Computer Science & Technology**.** Sri Krishna Devaraya University Ananthapur ,Andhra Pradesh India. Presently he is the Chairman of Board of Studies. He Published 11Papers for National and International Journals. He attended for 3-National Conferences His area of interest is on Computer Networks, Data Mining, Artificial Intelligence and Image Processing.