Quadratic Residue Based Address Allocation for Mobile Ad Hoc Networks

Xiaowen Chu[1], Yi Sun[2], Ke Xu[3], Zeeshan Sakander[2], Jiangchuan Liu[2]

[1] Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

[2] School of Computing Science, Simon Fraser University, BC, Canada

[3] Department of Computer Science and Technology, Tsinghua University, Beijing, China

***Abstract* – Address allocation in Mobile Ad Hoc Network (MANET) receives significant importance recently, as a mobile device cannot participate in unicast communications until it is assigned with a conflict free IP address. All routing protocols assume nodes to be configured a priori with a unique IP address. Unlike infrastructure based networks, MANET supports autonomous and spontaneous networking and therefore, should be capable of self organization and configuration. We present a new address allocation protocol in MANET based on the concept of quadratic residue. Each node in the network is capable of assigning a unique IP address with low latency. Addresses are reclaimed automatically, as the quadratic residues lie in cycles. This saves lot of extra communication overhead and bandwidth. Our approach also has support for network merging and partitioning. The proposed scheme can be applied to large scale MANETs with low communication overhead, even distribution, and low latency.**

I. Introduction

Mobile Ad Hoc Network (MANET) is an independent self organizing network in which each node functions as an end host and a router. This form of wireless network is created by mobile nodes without any existing or fixed infrastructure. The formed network can be deformed on the fly without the need of any system administrator. An ad hoc network mainly consists of hand held devices and laptop computers. These devices usually have limited transmission range, bandwidth and battery power.

Nodes in the MANET need some form of identity before participating in any form of communication. Each end host in the MANET needs 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 MAC address at the link layer level. However, each end host needs some form of a network address to communicate with other end hosts that are several hops away. This network layer address will uniquely identify each node present in the network. Traditional IP based address assignment like DHCP [3] is not suitable because nodes in the MANET are highly mobile and a central authority is not always reachable. Mobile IP [2] is also not a solution because MANET nodes do not stay connected with a wired network all the time. Address allocation in MANET poses special characteristics as opposed to wired networks. An address allocation protocol is required to enable dynamic address assignment to all nodes in a MANET.

The vital concerns of MANETs due to mobility are network partitions and mergers. MANETs can get separated into several different small partitions thus result into no or impossible communication across these partitions. These network partitions may or may not merge back later. In this scenario, a node is unaware of network partitioning. Two or more independent MANETs can get merged into one big network. MANETs have low bandwidth, limited transmission range and power. Therefore it is essential to keep broadcast messages and extra communications overhead as minimum as possible. Nodes in the MANET may either switch off from ad hoc mode rapidly or move away permanently from the MANET without providing any information. Our research is motivated to provide a scalable and delay free network layer address assignment protocol that addresses all these issues with almost zero extra overhead.

The rest of this paper is organized as follows. Section II presents the related work. Section III introduces the concept of quadratic residue. Section IV presents the address allocations algorithm, and its performance is evaluated in Section V. Finally, Section VI summarizes the paper.

II. Related Work

In past decade, lot of research has been done on ad hoc routing protocols. All the routing protocols require each node to have its own unique IP address to transmit packets. Address allocation protocols for wired networks [2, 3] are not directly applicable to wireless networks. According to [10], address allocation protocols for MANET can be classified into three categories: 1) stateless protocol; 2) statefull protocol; 3) hybrid protocol.

In stateless approaches, conflict detection allocation scheme is used. In [1], a node randomly chooses an address from prescribed range of address block and then verifies its uniqueness by performing duplicate address detection (DAD). No address allocation table needs to be maintained. Another stateless approach named weak DAD [6] allows two nodes to have the same

IP address in the MANET but misrouting is prevented by associating a unique key with each node.

Stateful approaches normally assume the existence of one or more central authorities, and central or distributed allocation tables are maintained. If distributed tables are used then the states of all the tables have to be synchronized periodically. In leader based approach [4] the work load on a leader is too heavy and a node is not always reachable in MANET. In MANETconf [8] address assignment is based on a distributed mutual exclusion algorithm that treats IP addresses as a shared resource. Complete synchronization is required between all nodes to avoid duplicate addresses. In Buddy protocol [7], a new node gets half of the initiators address space. If a node leaves the network abruptly, part of the address space is not available for future allocation. To solve this problem, a synchronization procedure requires all nodes to periodically flood their allocation table in the network. Prophet address allocation uses a distributed function to allocate unique addresses [5]. There is a low probability that a number will repeat in a certain sequence; and as a result, it may lead to unknown address duplication. In [11] a ring based approach is used. Numbers are repeated in a ring but the problem is the assumption that when a node leaves the network, it informs its neighbor before switching off from ad hoc mode. When two networks merge, we will have duplicate addresses amounting to the size of smaller network.

In hybrid approaches, the techniques from stateful and stateless schemes are combined. In [12], nodes passively collect information about assigned addresses through piggybacking routing packets to save network bandwidth.

III. PRELIMINARIES OF NUMBER THEORY

In this section, we briefly review several concepts in number theory that are fundamental to our address allocation scheme.

Definition 1: Suppose n is an integer. Integer a is defined to be a **quadratic residue** (QR) modulo n if $a \not\equiv 0 \pmod{n}$ and the congruence $y^2 \equiv a \pmod{n}$ has a solution $y \in \mathbb{Z}_n$. a is defined to be a **quadratic non-residue** (QNR) modulo n if $a \not\equiv 0$ (mod n) and a is not a quadratic residue modulo n.

Definition 2: Suppose p is an odd prime and a is an integer. Define the **Legendre symbol** $\left(\frac{a}{p}\right)$ as follows:

$$\left(\frac{a}{p}\right)=\begin{cases} 0 & \text{if } a \equiv 0 \pmod{p} \\ 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ -1 & \text{if } a \text{ is a quadratic non-residue modulo } p \end{cases}.$$

Theorem 1: For prime number p, a is a quadratic residue modulo p if and only if $a^{(p-1)/2} \equiv 1 \pmod{p}$.

Definition 3: Suppose n is an odd positive integer, and the prime power factorization of n is $n=\prod_{i=1}^{k} p_i^{e_i}$. Let a be an integer. The **Jacobi symbol** $\left(\frac{a}{n}\right)$ is defined to be $\left(\frac{a}{n}\right)=\prod_{i=1}^{k}\left(\frac{a}{p_i}\right)^{e_i}$, where the symbols on the right side are the Legendre symbols.

IV. ADDRESS ALLOCATION ALGORITHM

In this section, we first present a concept named residue cycle, and then present our address allocation algorithm, and finally analyze the network partition and merger.

A. *Quadratic Residue Cycle*

We use an example to illustrate the idea of quadratic residue cycle. From Table 1, it is easy to see that 1, 3, 4, 5, 9 are quadratic residues modulo 11. If we calculate quadratic residue from 1, it gives back 1 repeatedly. If we start at any other point, for example at 4 or 9, we get the next quadratic residue as shown in Fig. 1. The sequence of quadratic residues is referred to as a quadratic residue cycle.

Table 1: Quadratic residue modulo 11

a	**1**	2	**3**	**4**	**5**	6	7	8	**9**	10
a^2 mod 11	**1**	4	**9**	**5**	**3**	3	5	9	**4**	1

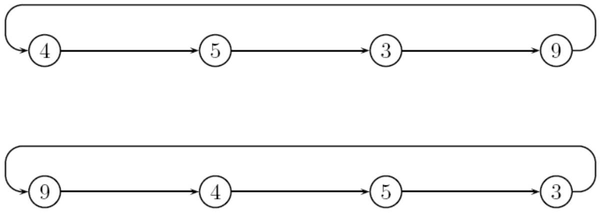

Figure 1: Quadratic residue cycle

B. *Address Space and Quadratic Residue Algorithm*

Assume we have two odd prime numbers p and q, then calculate $N = p \times q$. Denote ϕ as Euler's phi-function, i.e., $\phi(n)$ is the number of integers a, $1 \le a \le n$ and gcd(a, n)=1. Then the number of quadratic residues modulo N can be shown to be $\phi(n)/4 = (p-1)(q-1)/4$.

We want the quadratic residue modulo N to occur in distinct cycles. A quadratic residue lying in one cycle would not be present in any other cycle modulo N. Initial quadratic residue of a cycle acts as a seed S_0 to generate a sequence of numbers until the seed repeats again. Period or interval is the gap between the first and second occurrence of a same number in a sequence, i.e., the length of the cycle. We are interested in long intervals. In Fig. 2, S_0 is a seed, such that it is the first quadratic residue which generates a sequence of quadratic residues. These sequence repeats in cycles and the last quadratic residue before an interval ends is denoted by S_π.

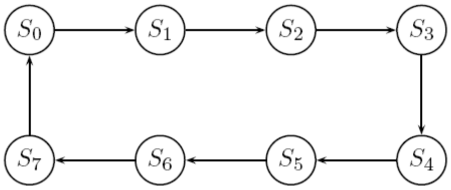

Figure 2: Quadratic residue sequence

For explanation purpose, we will give a small example. Let p = 23 and q = 7. Then N = 161 and $\phi(n)$ = $(p-1)\times(q-1)$ = 132. The total number of quadratic residues modulo N is 33 and there are five cycles in total as shown in Table 2: one cycle of length one, one cycle of length two, and three cycles of length ten. Note that if we use large primes we could get cycles of long lengths. Quadratic residues lying in these long cycles are used to allocate addresses in our protocol. In this way the address space is reclaimed automatically and each node can be assigned a unique address. Note that each of these cycles belongs to a unique seed. The intersection of any two different cycles is always a null set.

Table 2: Seed Generated Sequence

Seed S_0	Sequence S_i
1	1
93	93, 116
2	2, 4, 16, 95, 9, 81, 121, 151, 100, 18
8	8, 64, 71, 50, 85, 141, 78, 127, 29, 36
25	25, 142, 39, 72, 32, 58, 144, 128, 123, 156

Now we present an algorithm to find the quadratic residue cycles for $N = p \times q$ where p and q are two prime numbers. We used Jacobi symbols to find quadratic residues because its computational complexity $O(\log p)^2$ is less than the Euler criterion of $O(\log p)^3$. To test whether an integer a is a quadratic residue modulo n, we can calculate the Jacobi symbol $\left(\frac{a}{n}\right)$ and Legendre symbol $\left(\frac{a}{p}\right)$. If $\left(\frac{a}{n}\right)$ = -1, then a is a quadratic non-residue modulo n. If $\left(\frac{a}{n}\right)$ = 1, since $\left(\frac{a}{n}\right)=\left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$, there are two cases: (1) $\left(\frac{a}{p}\right)=\left(\frac{a}{q}\right)$ = -1; (2) $\left(\frac{a}{p}\right)=\left(\frac{a}{q}\right)$ = 1. In case (1), a is not a quadratic residue modulo p, hence it cannot be a quadratic residue modulo n. In case (2), a can be shown, from Chinese remainder theorem, to be a quadratic residue modulo n. Based on this observation, our algorithm to find quadratic residue cycles is shown in Fig. 3.

```
Algorithm: QRCycles($p$, $q$)
$T = \phi$;
$n = p \times q$;
$phi = (p-1)\times(q-1)$;
for ($i \leftarrow 1$ to $phi/4$)
{
  if ($i \in T$) break;
  $C = \phi$;
  $x = i$;
  if ($\left(\frac{x}{n}\right)$ == 1 and $\left(\frac{x}{p}\right)$ == 1)
  {
    while ($x \notin C$) do
      $C = C \cup \{x\}$;
      $x = x^2 \bmod n$;
    $T = T \cup C$;
    output($C$);
  } // end of if
} // end of for
```

Figure 3: Quadratic Residue Algorithm

C. Address Allocation Protocol

When a mobile node switches on to ad hoc mode it starts the timer, sends a DISCOVER message, and waits for a reply. If it receives no reply, it repeats the process for a suitable number of times. If all attempts fail, it assumes that it is the first node in the MANET and chooses two prime numbers p and q that are congruent to 3 mod 4. Prime numbers of this form are chosen because their square roots are easy to calculate. Then it computes $N = p \times q$ and $\phi(N)$ = $(p-1) \times (q-1)$. This initiator node calculates the number of distinct cycles and length of each long cycle (address block). If the length of the long cycle is small, the initiator can repeat the process until it finds long cycles. The initiator (first node) then configures itself with an IP address, keeps the seed S_0 value of each distinct long cycle and its corresponding range.

When a new node joins the network, it sends a DISCOVER message. An already configured node provides it an IP address. Along with an IP address a new node also receives a set of seeds S_0, and the corresponding range of each seed. Therefore, a new node gets an IP address, state value (seed) and range. Now the new node along with participating in communication is also capable of assigning a unique conflict free IP address without taking permission from any other node in the MANET. Each state represents the sequence of addresses in that cycle. A newly joined node after configuration will calculate the next address by squaring the current quadratic residue modulo N. Each cycle of quadratic residue is disjoint, therefore a node using a cycle x knows that a cycle y would not have any address in common. Therefore, the probability of duplicate address assignment is zero. This greatly reduces the delay associated with address assignment. In other techniques, a node needs to run DAD or has to require permission from remaining nodes in the MANET for address assignment. Our algorithm not only decreases the latency and delay but also reduces the communication overhead and saves bandwidth.

After maximum address range has been reached by a node, it has two options: either to start re-assigning the address as they repeat; or increment its state value by choosing an unused seed. If the re-assignment of first address will lead to duplication, the node will increment its state value to another seed. To confirm that the seed is not in use by another node in the MANET, it floods the network with a NEWSEED message and waits for the reply from all other nodes. If it receives one negative acknowledgement (NACK), it chooses another seed and repeats the process. If it does not receive any NACK, it assumes that the seed is free and will use it to assign addresses to new nodes.

D. Network Partitioning and Network Merger

If a network gets partitioned, node in each partition can still continue assigning unique address. As each QR cycle is disjoint, no address duplication will occur. When partitions merge back later address uniqueness is guaranteed. This way we handle network partitioning without incurring any extra communication overhead. Consider two independent MANET as in Fig. 4. All of the nodes in each MANET have unique IP addresses. Now we will see, what happens when two independent networks merge. Network ID (NID) is piggy backed in HELLO messages. A network merger is detected when a node hears a HELLO message with a different NID. Node D is communicating without any problem with node A having IP address ‘a’. Note that in the second MANET there is node K that also has an IP address ‘a’. As these nodes belong to different networks, there is no address duplication or communication problem. Let us consider what happens when these two independent networks come close to each other and get connected to form one big MANET. Misrouting can occur because of duplicate addresses as shown in Fig. 5. Nodes A and K have the same IP address and therefore we have duplicate addresses in the new MANET. As a result, packets that were meant to be routed to node A could be misrouted to node K. We have to solve this duplicate address problem and/or prevent misrouting after network merger.

We have two solutions of handling a network merger. One makes use of DAD and the other does not depend on DAD. In the first solution there is no need to invoke an explicit procedure upon merging of the two independent MANETs. We can associate $\phi(N)$ of each MANET to be the key of the respective MANET. The idea is similar to Weak DAD, but we associate a single network key to all nodes of one network instead of using a separate key for each node. The logic is to distinguish between the nodes having the same IP addresses and belonging to two separate networks during network merging. We will show that this does not increase communication overhead and we can still prevent misrouting. When we choose two large primes, $\phi(N)$ of each network would be different with a high probability, as each network selects p and q independently. Even two nodes have the same address, misrouting can still be prevented by associating $\phi(N)$ with IP address to identify each node. In this way, we can easily distinguish between two nodes holding the same IP address but having different key $\phi(N)$. Only if the two nodes have the same address and the two MANETs have the same value of $\phi(N)$, the conflict cannot be detected. This probability can be controlled to be very small by selecting large primes of p and q.

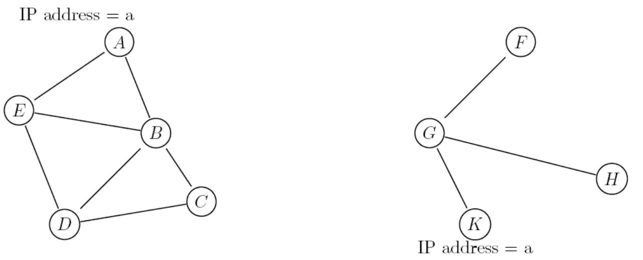

Figure 4: Two independent MANETs

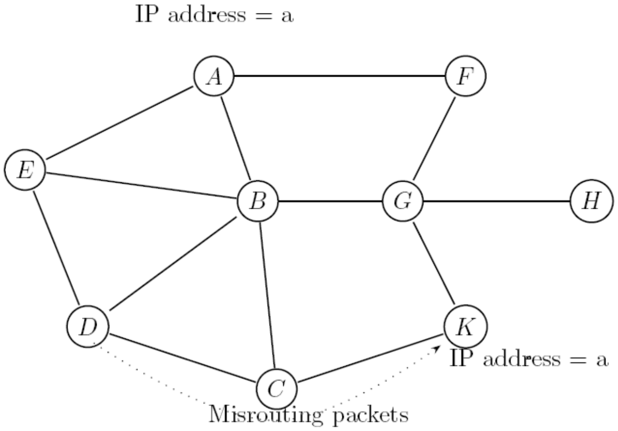

Figure 5: Network merger

In the second approach, we run DAD to remove duplicate addresses. Similarly, network merger is detected when a node hears a HELLO message with a different NID. The smaller NID is chosen to be the new NID of the merged network. In this approach two nodes from the intersection of MANET exchange N and the set of seeds S_0 of their individual MANET. Now these nodes will only check for the conflicted QR values by generating sequence of S_i of each cycle. If two values are found to be the same, duplicate addresses are detected. For example, if same QRs are found in both MANETs then they are in conflict with each other. All such 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 one. In our approach, we let the node with fewer TCP connections to change its IP address, in order to break minimum ongoing connections. This process is repeated for all duplicate addresses till there is no remaining duplicate address in the merged network.

V. Performance Evaluation

In our first experiment, we choose two safe primes of 12 bits: p = 2207 and q = 3467. We then can have N = 7,651,669 and $\phi(N)$=7,645,996. The total number of quadratic residues equals $\phi(N)/4$=1,911,499. By running our algorithm, we find 38 long cycles with length 50,228 each, as shown in Table 3, which means that we can have 38 clusters and each cluster can configure 50,228 nodes. If some clusters get partitioned from the original MANET, they can still assign unique addresses to newly joining nodes and when the partitions merge back there would be no duplication. As N is 24 bits long, we can fix the 8 bits prefix for network address. When a node hears a message from a node with different prefix, it assumes that it is a network merger and runs a network merging algorithm. Safe primes here were of 12 bit each but we can also choose bigger safe prime of 16 bit each and get N of 32 bits. In that case, we would be able to configure billions of nodes and we can piggyback the NID in hello messages, generated by the first node in the MANET.

Table 3: Results of safe prime experiment

Cycle Length	Number of Cycles	Number of QRs
1	1	1
29	38	1,102
1,732	1	1,732
50,228	38	1,908,664

In our second 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 less number of distinct cycles because the length of a cycle is extremely big. The two doubly safe primes we choose were 13 bits each: p = 4799 and q = 4919. The results are shown in Table 4.

Table 4: Results of doubly safe prime experiment

Cycle Length	Number of Cycles	Number of QRs
1	1	1
1,199	2	2,398
2,458	1	2,458
2,947,142	2	5,894,284

In our approach, the network part of the IP address can vary depending on the length of primes we have chosen. For example, consider Table 4, we chose two doubly safe primes of 13 bits each and get there product N to be 25 bits. As we are working in modulo, there will be no host part of the IP address generated by our program which will exceed 25 bits for this particular scenario. Therefore, we support Classless Interdomain routing(CIDR) [9]. With so called CIDRized network addresses, the network part of an IP address can be any number of bits long, rather than being constrained to 8, 16 or 24 bits.

VI. Conclusions

Our approach assigns unique addresses to new nodes with low latency without relying on periodic flooding which consumes a considerable amount of bandwidth. It distinguishes between concurrent address request and replies, and handles the mobility scenarios at the time of address assignment. After reaching the maximum address pool size our addressing function makes sure that addresses repeat automatically to ensure conflict free future address allocation. We guarantee disjoint address space.

Moreover, our protocol provides reliable state synchronization in presence of packet loss, delays and network merging. In our approach, network partitioning and subsequent merging does not induce duplicate addresses. Our solution ensures scalable and extremely low overhead solution for network mergers. In the best case scenario we do not have to run the DAD mechanism to check for duplicate address entries. This approach is capable of configuring at least millions of nodes with even distribution.

Acknowledgement

X. Chu's work is supported by Hong Kong RGC Grants HKBU 210605 and HKBU 210406. J. Liu's work is supported by a Canada NSERC Discovery Grant and an NSERC Strategic Project Grant.

References

[1] C.Perkins et al. IP address autoconfiguration for ad hoc networks. In Internet Engineering Task Force (IETF), Internet Draft, http://www.cs.ucsb.edu/~ebelding/txt/autoconf.txt, Nov. 2001.

[2] C.Perkins. IP mobility support. In Internet Engineering TaskForce (IETF), RFC 2002, Oct. 1996.

[3] R. Droms. Dynamic host configuration protocol. In Internet Engineering Task Force (IETF), RFC 2131, March 1997.

[4] M. Guines and J. Reibel. An IP address configuration algorithm for zero-conf mobile multihop ad hoc networks. In Int'l Wksp. Broadband Wireless Ad Hoc Networks and Services, Sophia Antipolis, France, August 2004.

[5] M.Mutka H.Zhou, L.Ni. Prophet address allocation for large scale manets. In Proc. of IEEE INFOCOM, San Francisco, CA, April 2003.

[6] N.Vaidya. Weak duplicate address detection in mobile ad hoc networks. In Proc. of ACM Mobihoc, June 2002.

[7] J.L. Peterson and T.A.Norman. Buddy systems. In Proceedins of Comm. ACM, June 1977.

[8] R.Prakash S.Nesargi. Manetconf: Configuration of hosts in a mobile ad hoc network. In Proc. of IEEE INFOCOM 2002, June 2002.

[9] J.Yu V.Fuller, T.Li and K. Varadhan. Classless inter-domain routing (cidr): an address assignment and aggregation strategy. In Internet Engineering Task Force (IETF), RFC 1519, September 1993.

[10] K. Weniger and M. Zitterbart. Address autoconfiguration in mobile ad hoc networks: Current approaches and future directions. In IEEE Network, pages 6-11, August 2004.

[11] Y.S.Chen and S.M.Lin. Raa: A ring based address autoconfiguration protocol in mobile ad hoc networks. In International Conference on Mobile Ad Hoc and Sensor Networks, December 2005.

[12] K. Weniger. PACMAN: Passive autoconfiguration for mobile ad hoc networks. In IEEE JSAC, Special Issue on Wireless Ad Hoc Networks, January 2005.