Partitioning and Merging of VASM protocol for IP Address Auto-Configuration in MANETs

Majid Taghiloo
Computer Engineering Department,Amirkabir University of Technology, Tehran, Iran
taghiloo@ce.aut.ac.ir

Mehdi Dehghan
Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
dehghan@ce.aut.ac.ir

Jamshid Taghiloo
Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran
j_taghiloo@comp.ac.ir

Raed Mousavi
Science and Research Branch, Islamic Azad University,Tehran, Iran
rmousavi@ece.ut.ac.ir

ABSTRACT
The main task of an address auto-configuration protocol is to manage the resource address space. It must be able to select, allocate, and assign a unique network address to an un-configured node. We present the VASM protocol [1], which allows nodes in an ad hoc network to configure their own IP address automatically. Due to the mobility of nodes, different networks can overlap, and nodes with the same address can accidentally get in contact. This paper proposes an innovative management of such merging of networks. The scheme uses virtual address space for addressing new nodes joining a network. The aim is to map one point from virtual address sheet to exactly one new node. The reason for using the term "virtual" is that the whole corresponding address space is a 2D flat sheet and each point of this sheet is virtually mapped to a node in MANET. The protocol uses coordinate values for generating addresses. This paper offers the high speed prediction of address space conflicts existence and it proposes a progressive method to avoid generating a huge amount of traffic on the wireless channels. Simulation experiments were performed using the behavioral analyzing to evaluate the performance of the protocol in terms of overhead and latency.

Categories and Subject Descriptors
[**Wireless Ad Hoc Network**]: Merging and Partitioning Management needs: Resolving possible Address Conflict, Reducing burst traffic overhead, Gr.

General Terms
Allocation, Mobility, Dynamic Topology, Mobile Ad Hoc Network, Unique, Security, Authentication Mechanism, Latency, Traffic overhead, Protocol Message, Progressive method can be useful.

Keywords
Ad Hoc Networks, Partitioning, Merging, IP Address, Auto-Configuration, MANET.

International Conference On Advanced Infocomm Technology'08,
Jul 29–31, 2008, Shenzhen, China.

1. INTRODUCTION
Mobile Ad Hoc Networks are collections of mobile hosts dynamically establishing short lived networks in the absence of fixed infrastructure. These mobile hosts are connected by wireless links and act as routers for all other hosts in the network. Hosts in a Mobile Ad Hoc Networks are free to move and organize themselves in an arbitrary manner.

With comparison to other wireless networks Mobile Ad Hoc networks are still in their infancy and there is scope for improvements in many respects, as in the following areas[]:

- Rapidly changing topology: Mobility of the hosts creates a dynamic network topology. Then topology management plays a key role in the performance of a routing protocol. Because the wrong topology information can considerably reduce the capacity, increase the end-to-end delay and routing control overhead, and increase the possibility of host failure [8][10].
- Unpredictable link properties: Wireless media is fairly unpredictable and packet collision is intrinsic to wireless network. Signal propagation face difficulties such as signal fading, interference and path cancellation [9].
- Power limitations: Mobile hosts are normally battery driven and this make the power budget tight for the entire power consuming component in the device. This will affect signal processing, transceiver input/output power, CPU processing etc.
- Security: Without adequate security, unauthorized access and usage may violate network performance and QoS because the physical medium of communication is inherently insecure [9][10].
- Routing: Since the topology of the network is constantly changing, the issue of routing packets between any pair of hosts becomes a challenging task. Multicast routing is another challenge because the multicast tree is no longer static due to the random movement of hosts within the network. Routes between hosts may

potentially contain multiple hops, which is more complex than the single hop communication.

- Address configuration: When new hosts joint a network, they need to be assigned a conflict free IP address as part of their initialization procedure. Nodes departing a network, as well as network merging and partitioning also need to be accounted for in order for MANETs to have adequate addressing.

Dynamic configuration in a wired network is accomplished using Dynamic Host Configuration Protocol (DHCP). However, it requires the presence of a centralized DHCP server which maintains the configuration information of all nodes in the network. Since a MANET is devoid of any fixed infrastructure or centralized administration, this approach cannot be used.

Since in a highly mobile, infrastructure-less scenario, pre-configuration of addresses is not possible, the node addresses need to be configured dynamically with minimum delay and packet loss. The main task of an address auto-configuration protocol is to manage the resource address space. It must be able to select, allocate, and assign a unique network address to each new node. When a node leaves the network, the corresponding address must eventually be de-allocated to prevent exhaustion of the address space. The proposed aims at mapping one point from virtual address sheet to exactly one new node using a virtual address space. The reason for using "virtual" word in this article is that the whole corresponding address space is 2D flat sheet and each point of this sheet is virtually mapped to one node in MANET. Every address auto-configuration protocol should fulfill the following requirements:

- Assign unique address for each node in network.
- Use the entire address space in address allocation process.
- Solve the problems associated with merging and partitioning of networks

VASM implements the progressive method of merging. This method allows the nodes to change slowly their configuration, so to obtain single system, if more networks tend to overlap.

2. RELATED WORKS

Previous research in the area of address auto-configuration primarily focuses on three problems: address generation, network merges and partitions, and duplicate address detection [2] [3] [4].

Paper [5] proposed an agent-based address auto-configuration protocol, MANETconf, which utilized the distributed agreement concept. A new node joining the network chooses its agent among the address configured nodes in the network, and its agent requests address allocation for all the nodes in the network. After all agreements, its local IP address is allocated to the node. However, address allocation time may become longer depending on the failures till getting all agreements and ACK implosion may occur.

[6] Suggested the address collision-free allocation scheme, Prophet Address Allocation, which utilized the function generating disjoint integer sequences. Although Prophet is able to reduce address allocation time, it is very difficult to devise such a function satisfying the mathematical constraints in the distributed environment. Besides, the probability of address conflict is not zero because the number of IP address is not infinity. Therefore, the assistance such as duplicate address detection is needed to avoid address conflict.

3. DEFINITIONS

In this protocol, nodes are classified into four categories:

- ***Allocator***: Maintain the address space. They allocate new addresses for joining nodes.
- ***Initiator***: An intermediate node between *Allocators* and *Requester* node that exchange all messages between them.
- ***Requester***: new node that needs to get IP address in order to join the network.
- ***Normal***: all other nodes are in this category.

Each *Allocator* in the network contains a disjoint address space. Therefore, address space overlap between Allocators is none [1].

4. VASM IDEA

According to VASN protocol [1], when a new node goes into ad-hoc mode, it sends a single-hop INITIATOR_SEARCH message, in order to find an *Initiator*. If there is no reply for this packet, the node assumes that it is the only node in that network and starts the network setup process. If the joining node gets more than one response, it selects the sender of the first arrived packet as *Initiator* and sends him an address request packet. The main task of *Initiator* is to obtain a new IP address from its *Allocator* and assign it to the requesting node (*Requester*).

If the received response was from an *Allocator*, select this node as its Allocator and asks for a new address immediately.

As mentioned before, in this protocol, each network has at least one *Allocator*. Each *Allocator* contains an address space used for assigning unique IP addresses for the newly joined nodes. Method of choosing nodes as *Allocator* and the way address space is assigned for that node is the chief task of this protocol. In addition to generating unique IP addresses, an *Allocator* can create another *Allocator* in the network for balancing the overhead of protocol traffic and minimizing the time of address assignment for new nodes. For efficient management of network merging and partitioning process, each Allocator holds a list of all *Allocator*s in the network.

When two nodes detect a partitioning, the Allocators of these nodes should exchange the list of its network *Allocators*. The number of *Allocators* in each network is limited. Thus the size of *Allocators'* list will be very small. By simple analyzing of *Allocators* address space situations, they can predect the possible address conflicts. First, the protocol should resolve these address conflicts and then address space integration should be done gradually.

The main difference between this protocol and other stateful protocols is in its bottom-up approach for address space distribution.

The protocol benefits from the following timers:

- **tJoin**: Requester node sets this timer after sending INITIATOR_SEARCH message and stops this timer after receiving INITIATOR_REPLY packet.
- **tAlloc**: The *Initiator* node activates this timer after sending ADDRESS_REQUEST packet to its own *Allocator* and stops that after receiving CONFIG_PARAM packet.
- **tInit**: By using this timer, the protocol can manages *Initiator* failure or departure.
- **tAllocatorSync**: This timer is used to synchronizing *Allocator's* table.

More detail of VASM protocol can be found in [1].

5. ADDRESS ALLOCATION MECHANISM

Address space of each *Allocator* maps to a square in the address sheet. Hence, every *Allocator* in the network has a mutually exclusive set of addresses. Address allocation process should be done in a way that minimizes the address conflict and also prevents wasting of address space.

The advantage of the proposed protocol, unlike [5], is that its *Allocators* hold some property of their corresponding square. The properties include coordination of bottom left corner, and last allocated point.

The simplest way to allocating a new address is selecting points from bottom-left corner of the square to top-right corner. But the probability of selecting the same square is higher than selecting the same point in the address sheet. The protocol has devised a method for solving this problem. For this purpose, STEP variable is used by setting its bits to one in a randomly fashion (Figure 1). The benefit of this method is that it decreases the probability of address conflict during network merging as the same square from two *Allocators* in separate networks may have been be assigned. Initially, all 8 bits in STEP variable are 0. A random number from 1 to 8 is selected and the corresponding bit in STEP variable is sets to 1.

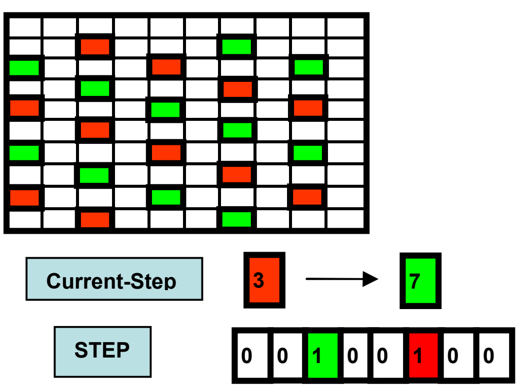

Figure 1. Point Selection Mechanism in Allocator's address space

The resulting value of STEP is assigned to CURRENT_STEP. At the next iteration, another bit from the 7 remaining bits is set to 1 in order to assign a new value to CURRENT_STEP. This process will continue until all 8 bits of STEP are set to 1. For address allocation, *Allocator* first starts *from* the CURRENT_STEPth point in its square and then, it adds eight to the last allocated point for allocating the next address. If the coordination of the new point is out of range, *Allocator* will continue with the next iteration to generate another STEP value. CURRENT_STEP is embedded in four most significant bits of ALLOC_STATE. If the FREE_ADDR_LIST is not empty, it is better to use IP address of depart nodes. It helps solving address exhaustion problem.

6. PARTITIONING

The nodes in the network can either depart abruptly or gracefully from the network.

6.1 Graceful Departure

The protocol assumes two modes for Allocator nodes: Normal Mode (NM) and Resource Saving Mode (RSM).

FD=Set of all depart node's address

S=All address space of Allocator

A=Set of all allocated addresses

MAX (A) =Last allocated address

Len (A) =Number of elements in set A

$$F_D = \{\forall x \mid x \in S \wedge x \notin A \wedge x \leq< MAX(A)\}$$

If $\dfrac{Len(A)}{2} < Len(F_D)$ *and mode* = NM *then*

Set mode = RSM

Else if $\dfrac{Len(A)}{2} \geq Len(F_D)$ *and mode* = RSM *then*

Set mode = NM

The above formulas states when an *Allocator* switches between the two modes.

Every *Allocator* node uses dynamic list (FREE_ADDR_LIST) for reclaiming IP address of depart nodes. In the worst case, the maximum size of the list is half the number of allocated addresses for each *Allocator.* This was the case in NM.

A node wishing a graceful departure sends a departure request to one of its neighbors before leaving the network. If this node is an *Allocator*, it sends DEPART_ALLOCATOR message giving coordination of bottom left corner of its square, last allocated point, ALLOC_BYTE, ALLOC_STATE, STEP, FREE_ADDR_LIST and *Allocator* tables, to one of its neighbors. There are several methods for selecting this neighbor. An example can be invoked the hello message of the routing protocol. Also, the selected neighbor being a normal node is the first priority. If the leaving node receives more than one message from its neighbors, it can select a node with greater IP address. After receiving departure message, the neighbor node changes its own address and state to the leaving *Allocator's* address and state. The new Allocator also sends DEPART_REGULAR message to its parent *Allocator*. The parent adds the IP address of that normal node to FREE_ADDR_LIST.

One exception can occur upon departure which is when that all neighboring nodes are *Allocator*s. In this situation, the departing *Allocator* will select one of its child normal nodes and then, sends its current state to that node. Furthermore, it may happen that all normal nodes of the *Allocator* departed from network. It is the time that *the departing Allocator* hands the task for finding a replacement for itself to one of its child *Allocators*. In the worst case, if none of the children could help their parent, the *Allocator* removing process will start.

When the depart node is normal, it sends the DEPART_REGULAR packet to its *Allocator*. If the receiving *Allocato*r is in NM mode, it adds the node's IP address to FREE_ADDR_LIST and increments FREE_ADDR_LIST_LEN by one. In case of RSM, the only task of *Allocator* is incrementing FREE_ADDR_LIST_LEN.

As mentioned before, when the neighbor of the departing *Allocator* is an *Allocator*, it suffices to relaying the DEPART_ALLOCATOR packet. So, if the parent *Allocator* of departs node is in the network, it can see this message. Then it starts tDepart timer and waits for receiving DEPART_REGULAR packet from depart node. After tDepart timeout, if there is no DEPART_REGULAR packet received, the parent *Allocator* supposes that there is no normal child node for the departing *Allocator.*

6.2 Abrupt Departure of Nodes

The assumption that every node leaves the network after broadcasting depart message is not correct. Some nodes may crash or switch off without prior notice and some other may go out of range. Consider a network in which some *Allocators* have crashed. The network doses not have any address remaining, except the crashed address pool. The address loss would be detected as soon as a new node joins the network.

If no free IP addresses are found during the address allocation, it dose not necessarily mean that there are no free IP addresses in the network. It is possible that some nodes may have left the network abruptly. Hence, reclamation of addresses needs to be done whenever the nodes have left the network abruptly. This feature solves the scalability problem of this protocol. Protocol simply searches all of its child nodes, and then it can assign absent nodes addresses for new nodes. It should be mentioned that, if the abruptly leaving node is an *Allocator*, the synchronization process which is performed earlier, can detect and replace the absent *Allocator*.

7. MERGING

Because of the mobility of nodes in MANET, merging may occur in the networks. The protocol should address the problems arising in this situation and resolves probable address conflicts. The merging networks can come from originally separated networks or they may belong to a single network that was partitioned before. The efficient protocol should benefit all of the available information to improve performance. VASM protocol uses evolutionary merging mechanism and avoids removing of *Allocators* and nodes IP addresses if the network was temporarily partitioned and subsequently merged.

7.1 Merging Detection Mechanism

The first step of the merging process is the detection of it. So the network merging will be detected when a node receives a message with different network ID (NetID). After that, merging management will start.

7.2 Address Resource Preserving Mechanism

During the process of merging originally separated networks, one of these networks remain its address resources unchanged and others will free their address resources. The selection of which network be unchanged is a main challenge. The network that contains maximum number of *Allocators* and nodes can be a good choice. In fact, both of these factors may lead to a single network. The existence of maximum number of nodes in the network cause decreasing free address lists (that contains the address of depart nodes), which inevitably improves the utilization of system resources.

7.3 Address Resource Removing and Node Address Changing Mechanism

With regards to the pattern that *Allocators* obey for expanding the address resources [1], in the course of merging process, address resource of the network should be maintained integrated. Therefore, some nodes must change their addresses and of course some *Allocators* have to change their states. For this reason the proposed method, by using gradual method for address changing and also by limiting of the number of these changes, have good speed and efficiency.

There are some critical questions that protocol should answer:

- Which of sub-networks should free their IP addresses of nodes?
- How to remove IP address?
- Which of address resources should be remain unchanged?

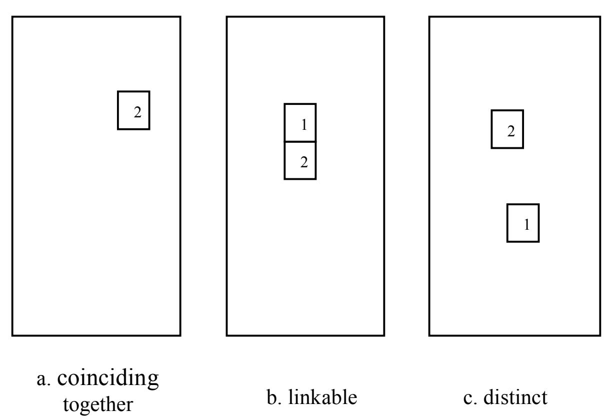

Figure 2. All possible state for *Allocators* address space

The figure 2. depicts all possible states that may be occurred. The proposed mechanism answers these questions with the help of situations stated in this figure.

This figure shows that there are only three states for the *Allocators* address resource: in the first state, the resources of two *Allocators* are coinciding together (figurer 2.a). In second state, according to the VASM address resource expansion policy, one can be child of another (figurer 2.b). In the last state, address resources of Allocators are distinct (figurer 2.c).

The main policy in the management of address conflicts follows these steps:

At the first step, the *Allocators* that having same address resources should be changed, and then, if the *Allocators* have a condition mentioned in figure 2.b, it just changes the state in one of the *Allocators*. Other *Allocators* must gradually change their addresses and normal nodes belonged to them.

8. SIMULATION EXPERIMENTS

Simulation experiments were performed using the behavioral analyzing to evaluate the performance of the protocol in terms of overhead and latency. The random waypoint mobility model was used. The speed of nodes in the network was selected randomly between 1 to 15 meters per second and the pause time was 10 seconds. The simulation duration was 3600 seconds. The JOINING_PKT_RETR threshold *was 2, and the* SQUARE_SIDE and SHEET_SIDE were set to 0x8 and 0xFFFF respectively. The network area was a square of 670m × 670m. Transmission range of the nodes selected randomly from 150 to 300.

Figure3 shows the overhead of partitioning and merging of networks.

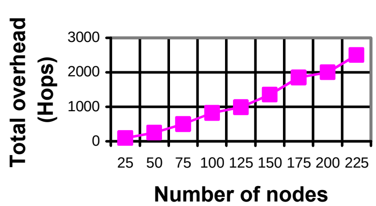

Figure 3. VASM Auto-Configuration traffic overhead

9. CONCLUSION

We presented a progressive method for managing of merging and partitioning of VASM protocol for dynamic configuration of nodes in MANETs. We have addressed the issue of unique IP address assignments to nodes in the absence of any static configuration or central servers. The basic idea is to dynamically distribute virtual address space among the dynamically selected *Allocator* nodes. This paper offers the high speed prediction of address space conflicts existence and it proposes a progressive method to avoid generating a huge amount of traffic on the wireless channels. The management of virtual address space conflicts is simple because each *Allocator* node holds the topology of squares corresponding to the other *Allocator* nodes. Currently we are working on the Integration of virtual address space after merging of different networks.

10. REFERENCES

[1] Taghiloo M., Dehghan M., Taghiloo J., Fazio M., "New Approach for Address Auto-Configuration in MANET Based on Virtual Address Space Mapping (VASM)," International Conference on Information and Communication Technologies: from Theory to Applications (IEEE ICTTA 2008), Damascus, Syria, 7-11 Apr. 2008.

[2] Jeff. Boleng, “Efficient Network Layer Addressing for Mobile Ad Hoc Networks”, in Proc. Of international Conference on Wireless Net-works (ICWN'02), Las Vegas, USA.

[3] N. Vaidya, “Weak Duplicate Address Detection in Mobile Ad Hoc Networks”, submitted for MobiHoc’02, June 2002.

[4] Archan Misra, Subir Das, Anthony McAuley, and Sajal K. Das, “Autoconfiguration, Registration, and Mobility Management for Pervasive Computing”, IEEE Personal Communication, August 2001, pp 24-31.

[5] S. Nesargi, R. Prakash, “MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network.” Proc. IEEE INFOCOM 2002, New York, NY, June 2002.

[6] Hongbo Zhou, Lionel Ni and Matt Mutka, “Prophet Address Allocation for Large Scale MANETS,” In Proceedings of the IEEE conference on Computer Communications (INFOCOM), San Francisco, 2003.

[7] Indrasinghe, S. Pereira, R. Haggerty, J., “Protocol Specification for Conflict Free MANET Address Allocation Mechanisms”, International Conference on Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008, pp 1434-1349.

[8] Navid N and Christian B, “Topology Management for Improving Routing and Network Performances in Mobile Ad Hoc Networks”, Mobile Networks and Applications 9, 583-594, 2004.

[9] P Mohapatra, Jiang Li, Chao Gui (University of California), QoS in Mobile Ad Hoc Networks, IEEE Wireless Communications, June 2003.

[10] Magnus Frodigh, Per Johanson, Peter Larson, Wireless Ad Hoc Networking – The art of networking without a network, Ericsson Review No 4, 2000.