Advanced Disjoint Address Allocation for Mobile Ad Hoc Networks

Jinkyu Lee, Sehoon Kim and Ikjun Yeom
Dept. of Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
Email: {jklee, kimsh}@cnlab.kaist.ac.kr, yeom@cs.kaist.ac.kr

***Abstract*—Regarding to address allocation, a mobile ad hoc network (MANET) may suffer from the lack of address and network partition/merging due to mobility of nodes. In this paper, we propose a new address allocation protocol for dealing with those problems. The proposed protocol is developed based on disjoint address set distribution with binary splitting for scalability, and provides special treatments for resolving the lack of addresses. We also present an effective technique for handling network partition and merging. Through simulation, we show that the proposed protocol is effective to allocate addresses in a MANET with reasonable latency and communication overhead.**

I. INTRODUCTION

A MOBILE AD hoc NETwork (MANET) is a wireless network composed of mobile nodes without any infrastructure. One of the major obstacles to deploy MANETs in the real world is to assign unique addresses to mobile nodes and maintain their uniqueness. There have been several schemes proposed for address allocation in a MANET.

Based on who is in charge of address allocation, we may categorize them into three groups [1] as follows: (a) In decentralized schemes [2]–[4], each node itself configures its address with a global agreement; (b) in centralized schemes [5]–[7], a central server assigns an address to the new node; and (c) in neighbor-based schemes [8]–[11], a neighbor node of a new node is in charge of address allocation.

Whereas the previous schemes are mainly focusing on how to initially assign a unique address to a new joining node at the beginning state of a MANET, it has been paid less attention on how to maintain the uniqueness in spite of dynamically changing topology of a MANET. Regarding to maintaining unique addresses, there are two main challenges: (a) the lack of addresses; and (b) network partition and merging. The address space assigned to a MANET is finite and may drain unless it is properly managed. In a MANET, nodes are frequently in and out, and if a node leaves the network without returning its address properly, the address is wasted. As more nodes leave the network, more addresses are wasted, and eventually, the network becomes suffering from the lack of available addresses. Finding and reusing currently unoccupied addresses is not straightforward and may cause large traffic.[1] Frequent moving of a node also causes repeated partitioning and merging of the network. Partitioning itself is not harmful, but there may happen address conflict when partitioned networks are merged. Resolving this conflict also requires high communication cost.

[1]Some stateless schemes in [2]–[4] do not need this process. However, it has been addressed that those schemes are not suitable for large scale networks in [8] since they require global agreement for each address allocation.

In this paper, we propose a new address allocation protocol for MANETs. It adopts disjoint address set distribution (DASD) approach in [10], [11]. In this approach, each node in a MANET keeps a disjoint address set, and hands over a half of its set to a new joining node. Since the address set of each node is disjoint, uniqueness of address allocation can be guaranteed through communication only with neighbors, and it does not need any global agreement or centralized control. Hence, the DASD approach has advantage in its scalability compared to other address allocation schemes. The main limitation of this approach is inefficient utilization of address space: (a) When joins of new nodes are biased to a certain area of a MANET, address set can be unevenly distributed (skewed address distribution). As a result, some nodes in the MANET can suffer from the lack of addresses while others have enough addresses; and (b) Since address space of a MANET is distributed and maintained by individual nodes, addresses can be wasted when nodes leave the network suddenly without returning their address spaces properly (address leakage).

The main objective of this paper is to improve the DASD approach so that it works well in a large and dynamic MANET. To achieve that, we focus on improving utilization of address space and dealing with network partition and merging. To improve utilization of address space, we propose two techniques called *remote allocation* and *leakage collection*. Remote allocation is a process to allocate addresses from a remote node when local nodes (neighbors of a new node) do not have enough addresses. Leakage collection is for collecting unoccupied addresses and reusing them for the MANET.

To deal with network partition and merging, we consider two cases of merging as follows: (a) merging of two different networks from the beginning; and (b) merging of two networks which initially belonged to a network and was partitioned. Compared to case (a) in which two totally different networks are merged, case (b) is more likely to happen since in and out of a single node may induce merging and partition. In most previous protocols for address allocation, both cases are treated with the same way as follows: All the currently used addresses are collected and re-assigned to each node for avoiding conflict. For case (a), there seems no other scheme better than that. For case (b), however, it is too expensive to merge two networks which was just partitioned. In the proposed protocol, we present a simple technique to handle case (b) with a minimal cost.

The rest of the paper is organized as follows: we present detail description of the proposed protocol in Section II. In Section III, we present performance evaluation through *ns-2* [12] simulation. We conclude this paper in Section IV.

II. The Proposed Protocol

In this section, we propose a new address allocation protocol based on the DASD approach. Basic operations for address allocation are similar to other prior DASD schemes in [10], [11] except that the proposed one provides special treatments for skewed address distribution and address leakage. We also introduce a new technique for handling network partition and merging.

A. Address Allocation

In the proposed protocol, we consider two scenarios of address allocation depending on the state of neighbor nodes. When at least one neighbor node of a new node has enough addresses to hand over, the new node can obtain its address set from the neighbor. We call this process *local allocation*. Otherwise, the new node has to look for remote nodes with enough addresses. In this process, it is important to reduce latency and communication cost. We call this process *remote allocation*. In TABLE I, we present message types used in our protocol.

TABLE I
Message types in the proposed protocol

Message	Type	Content
NEIGH_REQ	one-hop broadcast	neighbor search request from a new node
NEIGH_REP	one-hop broadcast	reply for *NEIGH_REQ* from neighbors
LOCAL_REQ	one-hop broadcast	address request to neighbors from a new node
LOCAL_REP	one-hop broadcast	reply for *LOCAL_REQ* from one of the neighbors
AGENT_REQ	one-hop broadcast	address request to an agent from a new node
AGENT_REP	one-hop broadcast	reply for *AGENT_REQ* from an agent
REMOTE_REQ	network-wide broadcast	remote address search request from an agent to all nodes
REMOTE_REP	unicast	reply for *REMOTE_REQ* from some nodes to an agent

1) Local allocation: When a new node wants to join a network, it sends a one-hop broadcast message, *NEIGH_REQ*. Upon receiving the message, the neighbors reply back with the number of their addresses through *NEIGH_REP*. If there exists at least one neighbor with address set more than one, the new node obtains its address from the neighbor which has the largest address set through exchange of *LOCAL_REQ* and *LOCAL_REP*. This process is the same as the prior DASD schemes, and illustrated in Fig. 1(a). This process takes only four one-hop delays with $O(n)$ message complexity, where n is the number of neighbor nodes.

2) Remote allocation: When all the neighbor nodes have only their own address, the new node sends *AGENT_REQ* to one of its neighbors. The neighbor node (called an agent) broadcasts a message (*REMOTE_REQ*) to look for a node with more than one address. Upon receiving the message, each node replies back a unicast message (*REMOTE_REP*) to inform its address set. Then, the agent picks the node with the largest address set, and allocates the half of the set to the requester. The process of remote allocation is shown in Fig. 1(b).

In this process, we employ probabilistic reply for avoiding unicast replies in burst, that is, each node sends its reply with a probability. To determine the probability for reply, we consider the followings: (a) the number of replies should be limited for saving bandwidth; (b) a node with more addresses should have more chance to send its reply for even distribution of addresses; and (c) if there are not enough addresses for accommodating the current number of nodes, it should be notified in order to trigger the *leakage collection* process. Based on these considerations, suppose a probability μ for reply. For realizing (b), each node attempts to send its reply n_i times with μ, where n_i is the number of addresses belonging to that node. Then, the probability α that the agent does not receive any reply is given by

$$\alpha = (1 - \mu(1 - P_{un}))^{A_a} \tag{1}$$

where P_{un} is the probability of unicast packet loss, and A_a is the number of addresses belonging to the current nodes in the network. Here note that $A_t = A_a + A_l$, where A_t and A_l are the numbers of total addresses and leaked addresses, respectively. From (1), we can calculate μ for (a) by knobbing α with a given A_a. For (c), we substitute A_a with A'_a which is a configurable parameter and works as the threshold for determining whether there are enough available addresses or not. Then, we have

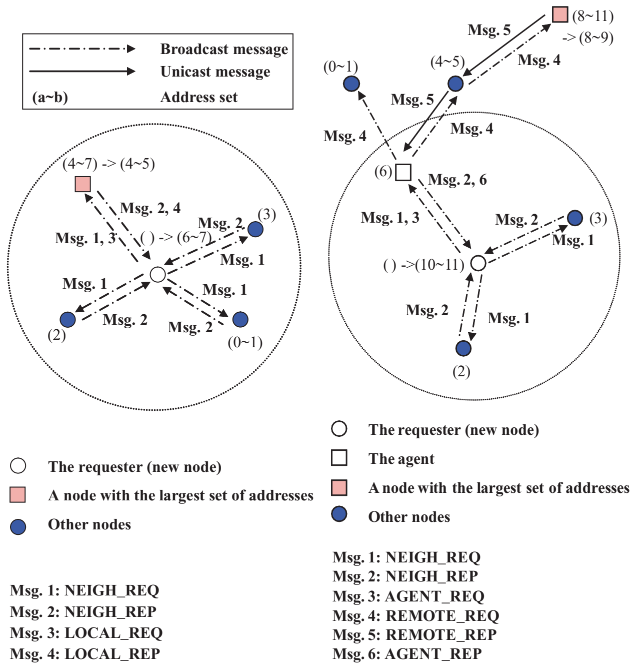
$$\mu = \frac{1 - \alpha^{\frac{1}{A'_a}}}{1 - P_{un}} \tag{2}$$

With μ in (2), if the agent does not receive any reply, we can judge that A_a is smaller than A'_a with the confidence interval $1 - \alpha$, and the *leakage collection* process is triggered. For properly setting A'_a, we need to know the current number of nodes (N) for preparing enough addresses. In the proposed protocol, N is measured in the *leakage collection* process, and then A'_a is set larger than N, i.e., $A'_a = \beta N$ for $\beta \geq 1$. Observe that α and β are inversely proportional to each other for a given μ. We will investigate the effect according to variation of α and β in Section III.

The time complexity of remote allocation is $O(d)$, where d is the diameter of the network, and one broadcast and several unicast messages are required.

B. Leakage Collection

Leakage collection is a process to collect currently unoccupied addresses (due to left of nodes) and redistribute them



(a) Local Allocation (b) Remote Allocation

Fig. 1. Local and Remote Address Allocation

to current nodes. In the proposed protocol, this process is triggered when the agent does not receive any response in the *remote allocation* process. At this time, the agent floods a message for requesting the currently occupied addresses. Upon receiving the request, each node forwards the request and sets its timer for response. Then, it waits for responses from its neighbors until the timer expires. After timeout, it sends a one-hop broadcast message containing cumulative address set which is the union set of its address set and address sets from neighbors. Here note that the timer of node i is set proportional to $d - h_i$, where h_i is the number of hops from the agent to the node. They can be measured roughly via flooding (i.e., simply using hop count in flooding messages). When the agent collects all the information on the currently occupied addresses, it floods the information to all the others.

Upon receiving the information, each node rearranges its address set in distributed fashion as illustrated in Fig. 2. Address rearrangement is performed by the reverse of binary splitting. Initially, the whole address space is divided into A_t/A_s subsets where A_s is the size of the smallest address set. Then, in the first iteration, a leaked address set is allocated to the node holding the address set which was immediately split with the leaked address set. In the next iteration, two neighbor address sets are merged, and we repeat the first iteration. This iteration is repeated until all leaked address sets are allocated. It takes at most $\log A_t$ iterations.

The time complexity of leakage collection process is $O(t \cdot d)$, where t is the average latency of one-hop communication, and three broadcast messages are required.

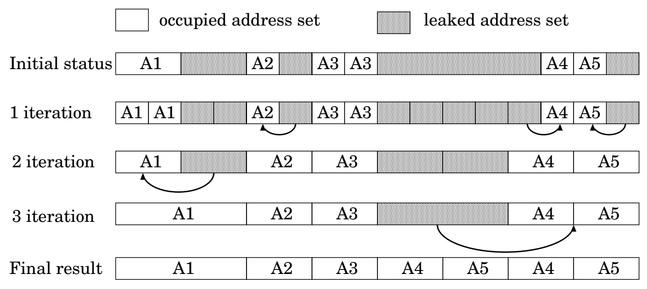


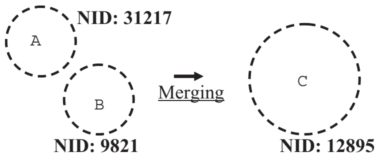
Fig. 2. Address Rearrangement in Leakage Collection

C. Network Partition and Merging

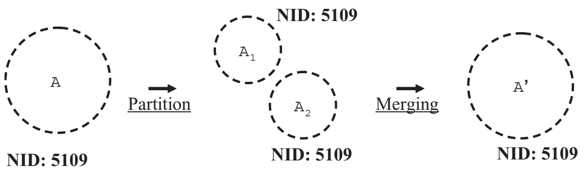
Mobile property of a MANET induces network partition and merging, and merging of two networks may raise address conflict. To deal with network merging, we may consider two types of merging. The first type is merging of two separate networks from the beginning. In this case, we should gather all used addresses in both networks, and re-distribute addresses to avoid address conflict. The second type is merging of two networks which stem from a network. Even though most merging events in practice belong to the second type, most previous schemes do not distinguish them, and treat as the first type. In this paper, we present a simple technique to distinguish them and deal with the second type of merging effectively.

In the proposed protocol, each node maintains two network IDs (NIDs) for the current and previous networks. An NID is randomly generated when a network is initiated, and changed whenever leakage collection occurs. Here we focus on handling merging of two networks since merging of more than two networks can be sequentially handled as several merging of two networks.

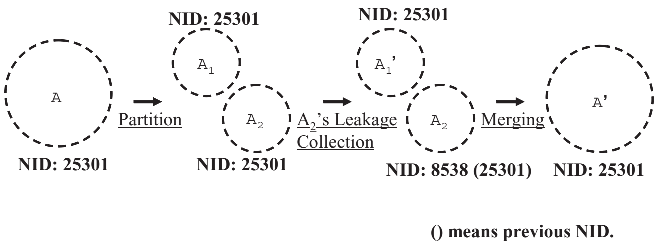
The NID pair (the current and previous NIDs) is periodically announced to detect network merging. When two networks are merged, we can consider three cases. If there is no match in the NID pair as shown in Fig. 3(a), the two networks are supposed to have been separate ones. Then, we just gather all the addresses and re-assign them. The process is similar to leakage collection, but in addition to reassignment of leaked addresses, address conflict should be removed. The time complexity is $O(t \cdot d)$, and the message complexity is $O(N_1 + N_2)$, where N_1 and N_2 are the number of mobile nodes in the two networks. When two just partitioned networks are merged before performing leakage collection in Fig. 3(b), the NID pairs are the same, and the merging event can not be detected. Here note that, however, the address sets are still disjointed, and there is no address conflict. When two networks partitioned from a network are merged after performing leakage collection once, there should be match in the NID pairs as in Fig. 3(c). In this case, each node gives up the addresses which was obtained from the previous leakage collection to avoid address conflict. If a node has been assigned an address set which was collected from the previous leakage collection, it re-performs address request again. Since it is required to re-assign addresses in one of two networks, the time complexity is $O(t \cdot d)$, and the message complexity is $O(min\{N_1, N_2\})$.



(a) Merging of two separate networks



(b) Merging of two same rooted networks without leakage collection



(c) Merging of two same rooted networks with leakage collection

Fig. 3. Network Merging Scenarios

III. Performance Evaluation

In this section, we present performance evaluation of the proposed protocol through *ns-2* [12] simulation. In simulation, DCF of the IEEE 802.11 is used as the MAC layer protocol. We use the standard values for the Lucent's WaveLAN as the radio model, and a nominal radio transmission range for each node is 250 meter.

A. Address Allocation with Leakage Collection

We look at how the proposed protocol performs address allocation and leakage collection in a MANET. The simulation scenario is as follows: A node joins the network at random time with 15 seconds average interval. The locations of the first 25 nodes are chosen for network connectivity, and the locations of the rest nodes are randomly selected in a 1000 meter by 1000 meter square region. To observe the impact of various N, life time of a node is changed over time as follows: In time between 0 to 1800 second, a node stays in the network for random time interval uniformly distributed on [400, 800] seconds. Then, to increase the number of concurrent nodes, life time of a node is increased up to [900,1800] seconds until 3600 second, and [1400, 2800] seconds until 7500 second. During the simulation, there have been 500 nodes joined the network, and the maximum number of concurrent nodes is about 180. The address space of the network is class C (256 addresses). The loss rate P_{un} for calculating μ in (2) can be directly measured in a MANET, but in the simulation, it is simply set to 0.2.

To discuss how to configure α and β, consider a MANET with a class C address space. When the network is managed properly so that there are more than 100 addresses available, the range between 0.01 and 0.05 is appropriate for μ to limit the number of replies to be less than ten. In Fig. 4, we present the relation of α and β in such network. N is set to 128. It is observed that we can adjust μ within the proper range (0.01 $\sim$ 0.05) by changing either one of α or β with the other fixed. In this simulation, we fix β as one, and change α to observe the impact of various μ.

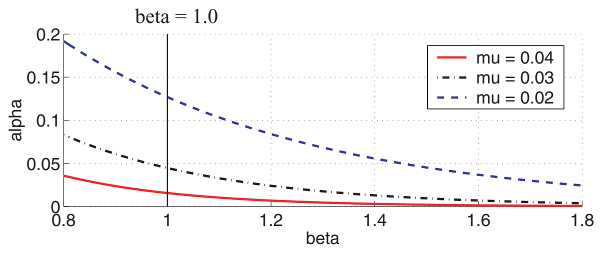


Fig. 4. Relation between alpha and beta

Simulation results are shown in Fig. 5. The solid line in the figure represents N over time, and dashed lines represent A_a with different α. It is observed that (a) A_a decreases as more nodes join and leave the network; (b) when A_a reaches N, *leakage collection* is properly performed; and (c) *leakage collection* is performed earlier with smaller α, but it does not much impact on operation of the proposed protocol.

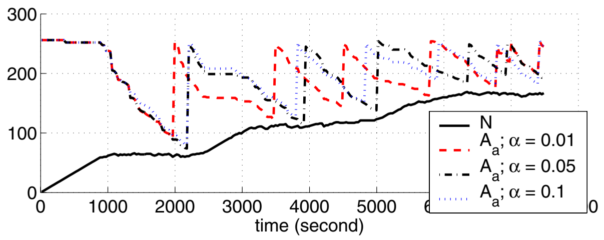


Fig. 5. Address Usage

In Fig. 6 and 7, we present latency and communication overhead in the simulation. When a new joins the network, we measure latency as the time interval for the node to obtain an address, and also measure communication overhead as the number of packets generated in the allocation process. To observe the impact of the number of nodes, we present data collected from three different periods as follows: *Period A* [800, 2400] second, *Period B* [3000, 5100] second, and *Period C* [6000, 7500] second. It is observed that both latency and communication overhead slightly increase as the number of nodes increases, but more than 90% nodes obtain their addresses within 0.5 second with less than one packet per a node in all the periods.

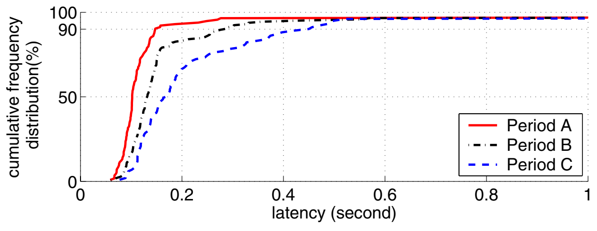


Fig. 6. Latency

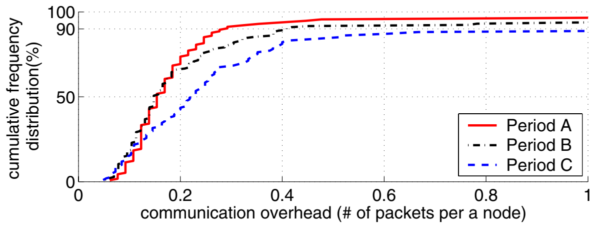
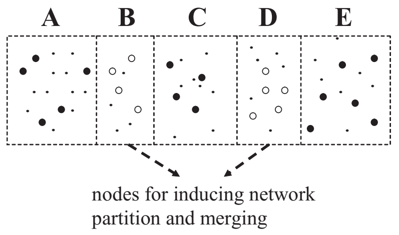
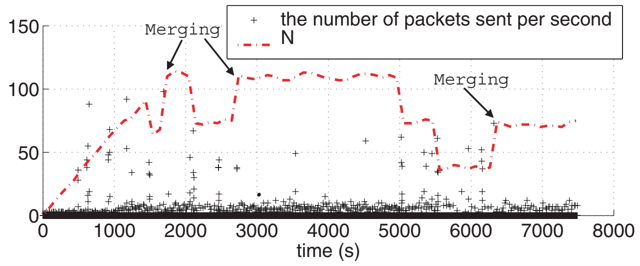


Fig. 7. Communication Overhead



(a) Topology



(b) Communication Overhead and N

Fig. 8. Network Merging

B. Network Merging

We consider a MANET which is partitioned and merged several times as follows: 500 nodes join the network in every 15 seconds on an average. As shown in Fig. 8(a), the total region of the network is 2750 meter by 1000 meter, and is divided by three 750 meter by 1000 meter regions (A, C and E) and two 250 meter by 1000 meter regions (B and D). Except some nodes which are in charge of connecting each network and inducing network partition and merging, the location of other node is randomly selected in A, C, or E region, and life time is uniformly distributed on [500, 1000]. Until 1497 second, all regions belong to the same network, and the network is partitioned or merged by eliminating or generating nodes in B or D region. We let A_t be 512, P_{un} be 0.05, and α be 0.05.

Fig. 8(b) represents the number of packets sent per second and N in the network including C region. The network is merged at 1687, 2715, and 6300 second, and partitioned at 1497, 2106, 5025, and 5527 second. Leakage collection is performed at 2435 and 6148 second. When network merging or leakage collection occur, the number of packets sent seems to be numerous, but considering N, the traffic is not seriously intensive.

IV. Conclusion

In this paper, we have made three main contributions as follows: (a) it has been addressed that it is needed to collect and reuse currently unoccupied addresses for MANETs; (b) we provide the method of handling network partition and merging, and thus, nodes in MANETs can keep address uniqueness in spite of dynamic topology; and (c) a new address allocation protocol has been proposed for MANETs. The proposed protocol has been devised for dealing with lack of addresses due to mobility of nodes, and provides two techniques, *remote allocation* and *leakage collection*. Through simulation, it has been shown that the proposed protocol performs well in allocating addresses in a MANET with limited latency and communication overhead.

Acknowledgment

This work was supported by SK Telecom Corporation.

References

[1] S. Kim, J. Lee, and I. Yeom, "Modeling and Performance Analysis of Address Allocation Schemes for Mobile Ad Hoc Networks," *to appear in the IEEE Transactions on Vehicular Technology*.

[2] C. Perkins, J. Malinen, R. Wakikawa, E. Royer, and Y. Sun, "IP Address Autoconfiguration for Ad Hoc Networks," IETF Internet draft, draft-ietf-manet-autoconf-01.txt, Nov. 2001.

[3] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," in *Proceedings of the 21st Annual Joint Conference of IEEE Conference on Computer Communication (INFOCOM'02)*, New York, NY, June 2002, pp. 206–216.

[4] J. Boleng, "Efficient Network Layer Addressing for Mobile Ad Hoc Networks," in *Proceedings of the International Conference on Wireless Network (ICWN'02)*, Las Vegas, NV, June 2002, pp. 271–277.

[5] S. Toner and D. O'Mahony, "Self-Organising Node Address Management in Ad-hoc Networks," *Springer Verlag Lecture notes in Computer Science 2775*, pp. 476–483, 2003.

[6] P. Patchipulusu, "Dynamic Address Allocation Protocols for Mobile Ad Hoc Networks," Master's thesis, Computer Science, Texas A&M University, August 2001.

[7] Y. Sun and E. Belding-Royer, "Dynamic Address Configuration in Mobile Ad hoc Networks," Computer Science, UCSB, Tech. Rep. 2003-11, June 2003.

[8] H. Zhou, L. Ni, and M. Mutka, "Prophet Address Allocation for Large Scale MANETs," in *Proceedings of the 22nd Annual Joint Conference of IEEE Conference on Computer Communication (INFOCOM'03)*, San Francisco, CA, Apr. 2003, pp. 1304–1311.

[9] Y.-Y. Hsu and C.-C. Tseng, "Prime DHCP: A Prime Numbering Address Allocation Mechanism for MANETs," *IEEE Communications Letters*, vol. 9, no. 8, pp. 712–714, Aug. 2005.

[10] M. Mohsin and R. Prakash, "IP Address Assignment in a Mobile Ad Hoc Network," in *Proceedings of IEEE Military Communications Conference (MILCOM'02)*, Anaheim, California, Oct. 2002, pp. 856–861.

[11] A. Misra, "Autoconfiguration, Registration, and Mobility Management for Pervasive Computing," *IEEE Personal Communications*, pp. 24–31, Aug. 2001.

[12] "The Network Simulator - ns-2," http://www.isi.edu/nsnam/ns/.