Adaptive Partition-Based Address Allocation Protocol in Mobile Ad Hoc Networks

Ki-Il Kim, Bai Peng and Kyong-Hoon Kim, *Member, KIMICS*

***Abstract*—**To initialize and maintain self-organizing networks such as mobile ad hoc networks, address allocation protocol is essentially required. However, centralized approaches that pervasively used in traditional networks are not recommended in this kind of networks since they cannot handle with mobility efficiently. In addition, previous distributed approaches suffer from inefficiency with control overhead caused by duplicated address detection and management of available address pool. In this paper, we propose a new dynamic address allocation scheme, which is based on adaptive partition. An available address is managed in distributed way by multiple agents and partitioned adaptively according to current network environments. Finally, simulation results reveal that a proposed scheme is superior to previous approach in term of address acquisition delay under diverse simulation scenarios.

***Index Terms*—** Mobile Ad Hoc Networks, Address Allocation, Adaptive Partition

I. INTRODUCTION

MANET (Mobile Ad Hoc Networks) is defined as one of representative self-organizing networks, which initialize and maintain networks without any help of infrastructure such as wireless access point in WLAN (Wireless Local Area Networks). To create and maintain this kind of networks, many network technologies should be implemented. Among them, many researchers have paid attention to ad hoc routing protocol, which is to maintain the path information for each destination.

The basic assumption of these protocols is that each node is uniquely distinguished by identification system. In many network systems, identification system is controlled by address allocation scheme. So, it is essential to introduce and develop an efficient scheme in practical networks. Similarly, unique address allocation is an imperative step for nodes in MANET. Generally, static IP address assignment scheme is very hard to achieve in MANET because it needs to be done manually with prior knowledge about the current network configuration, which does not match the main property of MANET, that is, self-organization. In other way, when it comes to employ Dynamic Host Configuration Protocol[1] in MANET, we need to introduce the centralized server. However, due to nodes' mobility, the stability of the centralized server is not guaranteed in MANET.

To mention above problems, many researchers have paid attentions to develop a new address allocation scheme without centralized one. However, they have long configuration time delay or cause much control overhead. The largest portion on control overhead is derived from procedure to figure out whether duplication address is allocated and maintain available address pool consistently.

In this paper, we propose DAAP (Dynamic Address Allocation Protocol) that is entirely autonomous and distributed for address allocation in MANETs. In DAAP, address is allocated by multiple decentralized servers to solve path availability problem to the server. Even though similar approach have been proposed, a new scheme is different from the previous work in that address allocation accomplished by adaptive partitioning of available address block according to network environment such as how many nodes joins. In DAAP, adaptive means that the number of agent and address block are dynamically controlled by network environments. In addition, address block is carefully managed by merging and seeking the leakage of available address pool.

The remainder of this paper is organized as follows. In section II, the related work for address allocation in MANET is summarized. The proposed scheme is presented in section III. Simulation results and analysis are described in section IV. Finally, we make a conclusion in section V.

Manuscript received March 17, 2009 ; revised May 18, 2009. The authors are with the Department of Informatics, Research Institute of Computer and Information Communication, Gyeongsang National University, 900 Gajwa-dong, Jinju, 660-701, Korea (Tel: +82-55-751-6282, Fax: +82-55-751-6280, Email: kikim@gnu.ac.kr)

II. RELATED WORKS

Address allocation schemes are classified into centralized and distributed approaches according to which node is charged for this procedure, a centralized server or distributed many agents. More specifically, we can classify existing distributed address auto-configuration scheme according to whether there is procedure for DAD (Duplicated Address Detection) because it plays a great role in allocation in terms of time and complexity. We provide the complete taxonomy of the schemes. As this taxonomy is fairly rich, the remainder of this section analyzes representative example protocols using top-down approach.

The concept of the centralized scheme is that a node will be chosen and works as server. We see instances of centralized schemes in [1]-[4]. The server node refers to an agent node, a leader node[4], or an address authority[2] in each scheme, respectively, but the operation of the server nodes is similar to each other. The server node maintains an address pool and is responsible for the address allocation. In these methods, the duplicated addresses by network merger can be simply detected using the server node's address pool.

In contrast, the distributed schemes operate in such a way that every node must communicate with each other to get an unique address. Hongbo et al., proposed a conflict free distributed address configuration scheme named Prophet Address Allocation using a function that produces an integer sequence[5].

DAD-based schemes are categorized as how strictly DAD procedure is applied. The strictness is implied by the name of each scheme. Perkins et al. proposed a distributed DAD scheme called Strong DAD[10]. A new node randomly selects an IP address and examines whether it is used in a MANET. If the chosen address is already used, it retries until it gets an unused address. Sanket et al., suggested an agent-based distributed address auto-configuration, MANETconf[9], using the distributed agreement concept. Unlike Strong DAD, a new node, which is called a requestor, asks for an address to one of the neighbors in MANET, which is called as initiator. The initiator then randomly selects an address and gets agreements from all other nodes in MANET and assigns the address to its requestor.

Passive DAD, presented in [14], tries to detect duplicate addresses without disseminating additional control information. Based on classic link state routing, the following three schemes are proposed, PDAD based on sequence number, PDAD based on locality principle, and PDAD based on neighborhood. Weak DAD protocol requires each node in the network to have a unique key. Weak DAD[15] requires that packets "meant for" one node must not be routed to another node, even if the two nodes have chosen the same address. This is achieved by using the key information for duplicated address detection.

III. A PROPSED SCHEME : DAAP

A new proposed scheme belongs to distributed scheme to prevent a single of failure problem on server and reduce the overhead on it. The proposed approach solves scalability problem because the larger the number of hop counts from a new mobile node to the server node, the longer centralized schemes take to find the server node and to get an address. In addition, it doesn't need to consider how to maintain a single server in a MANET as long as the network exists where mobile nodes are randomly in and out.

We present a distributed auto-configuration mechanism (DAAP) that guarantees unique IP address assignment. DAAP adopts IP block partitioning mechanism and is completely free from DAD. In our approach, most of the address assignments require local message exchange leading to low communication overhead and latency.

In the distributed scheme, some of them adapt similar approach by splitting the address space to avoid conflict among address. The examples include distributed protocol[7] and buddy[8], which have limitation in adaption and scalability. On the other hand, DAAP deals with these shortcomings by considering network environments such as number of node in the network and number of requests. Since dynamic topology is observed so frequently, this point should be emphasized. The main point worthwhile mentioning is that above two schemes cannot guarantee unique address assignment, which is regarded as prime performance metric in this research field.

A. Overview of DAAP

A new node joining the networks is called a requester. The configured node responsible for assigning an IP address to the requester is called an agent. We assume that the MANET starts with just one node. We call this first node the initiator of the network and the configuration of this node as MANET initialization. The initiator node keeps the whole IP address block which consists of a set of consistent IP addresses. In our system for IP assignment every node has a disjoint set of IP address that can be assigned to a new node without consulting any other node in the network. Fig. 1 shows the flow chart of DAAP.

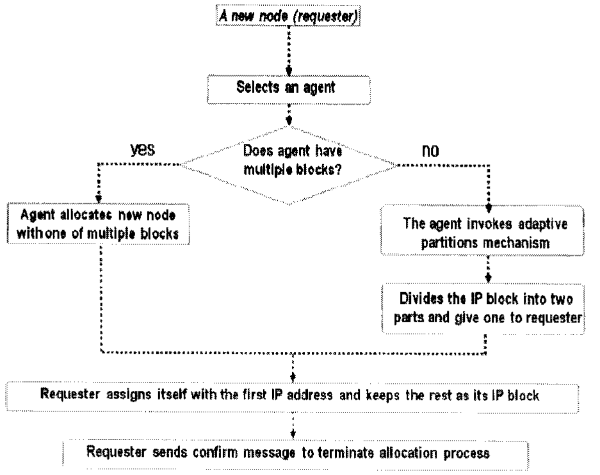

A new node (requester)
Selects an agent
yes
Does agent have multiple blocks?
no
Agent allocates new node with one of multiple blocks
The agent invokes adaptive partitions mechanism
Divides the IP block into two parts and give one to requester
Requester assigns itself with the first IP address and keeps the rest as its IP block
Requester sends confirm message to terminate allocation process

Fig. 1. Flow chart of DAAP

The operation in each step is described below.

1) In the beginning, there is only one node in the network that has the entire block of IP addresses.
2) When an un-configured node, ***K***, wishes to join the networks, it requests a configured neighbor node, ***L***, for an IP address. Node ***L*** assigns the requesting node ***K*** an IP address from its pool of IP address. It also divides the set of IP address into two (not even) and gives the part to the requesting node ***K***.
3) Nodes can leave the network either gracefully or abruptly at any time. When node *A* leaves a network gracefully, it returns its IP address and reserved IP pool to any node ***B*** nearby. Node ***B*** has the responsibility of handling this set of IP addresses. It can keep this block of IP address by itself or distribute to other nodes. On the other hand, when node *A* leaves the network abruptly it leads to IP address leak (because there is some IP address that is neither assigned to any node nor available for assignment to an un-configured node). This situation is handled by address reclamation mechanism presented later.
4) It is not necessary to do network range synchronization to keep track of the IP addresses assigned and detect any leaks in the available pool of IP addresses. We invoke these mechanisms only when all IP blocks are empty.

B. Acquiring New Address

Fig. 2 illustrates an example procedure for assigning new IP address. In Fig. 2, node *A* is initiator and node ***B***, ***C***, ***D***, and ***E*** join the network and request address sequentially. At the beginning, node *A* maintains table numbered 1. After ***B*** joins, *A* maintains table numbered 2 and ***B*** maintains table numbered 1 where node *A* shares its address pool with ***B*** according to proposed algorithm. Each entry in table consists of following form.

<IP_adddress, IP_block, agent_IP_Address, request_num, original_IP_block >

- IP_address : Node's IP address
- IP_block : range of IP address
- agent_IP_address : Node address which provides address block
- requester_num : The number of nodes it has configured
- original_IP_block : The total number of originally allocated address

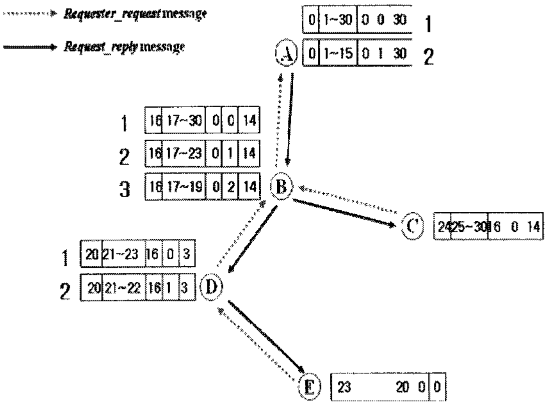


Fig. 2. Example of address allocation

More detailed message exchange and procedure is explained step by step.

Step 1) The requester periodically broadcasts requester_request message, then waits for a request_reply message until the timer expires. This continues for requester_retry threshold times. After all the failed retries, the node concludes that it is only node in the networks. It then, assigns itself the first IP address from the IP address block as its own and keeps the rest IP addresses as its reserved IP block, and then sets its network_id. We assume that this network identifier is universally unique.

Step 2) After MANET initialization, every time a new node (requester) requests an IP address, one of the existing MANET node (agent) within communication range of the requester initiates address allocation process for the requester. When a configured node receives a requester_request broadcast message, it first examines its IP block. If there are some available IP addresses there, it responds by sending a request_reply message. Otherwise, it replies with the negative message. It is possible that two or more configured nodes reply to the same broadcast message. In this case, the requester selects the configured node with the largest IP block as its agent.

Step 3) The requester sends an accept_ acknowledgement message back to one of the replied agents. In addition, the requester sends negative_ acknowledgement messages to the rest of the agents for the replied IP to be used for the other new nodes.

Step 4) When the agent node receives this message, it realizes that it is ready to assign a new IP address to the requester. If the agent node has multiple blocks of IP addresses, it assigns one of these blocks to the requester. Otherwise, it divides its set of available IP address into two disjoint parts. It then sends one subset to the requester and keeps the other subset with itself. Different from previous splitting protocol, we adopt a special function to calculate the proportion of partition of IP block. For example, a node many be chosen as an agent frequently in a short time. That means joins of new nodes are biased to a certain area or this busy node locates in the place where many high speed moving node pass by. Thus, this node does not divide its IP block into two equal parts as usual. For this, additional variable requester_num field is introduced.

Assume that requester_num is r and the threshold number is t. Agent node does not utilize equal division mechanism until $r >= t$. Instead, it partitions the IP block into two parts: one is $((r-1)/r)$ * IP_block_Size, the other is $(1/r)$ * IP_block_Size. Agent node updates its maintenance table by changing its IP_block range and increasing request_num.

Step 5) When the requester receives this set of available IP addresses, it assigns itself the first IP address from this set and keeps the rest as its available set of IP address.

Step 6) When the agent receives the confirm message, it terminates the IP assignment process.

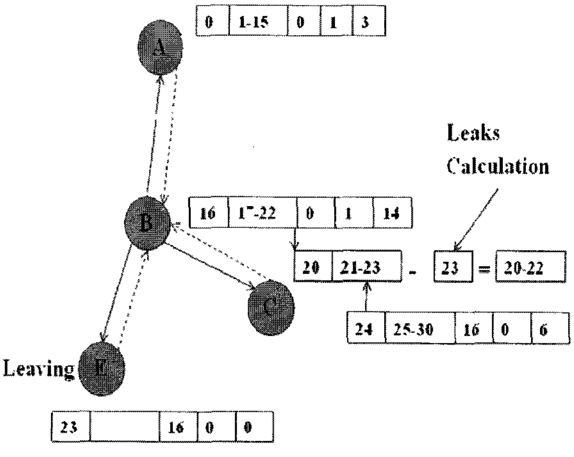


Fig. 3. IP address leaks detection

Table 1. Comparison of existing protocols

Protocols Features	AAA[10]	DAAP
Address block	The block can be the IPv4 private address block	A set of consecutive IP address
Agent selection	The neighbor with the largest proposed block	The configured block which has the largest IP block
IP block splitting	Static equal division	Dynamic proportional to network environments
Maintenance table	Each node in the network maintains a neighbor list	Every node just maintain its own IP block and the information of pre-agent
Protocol message type	There are 21 kinds of messages are used during the allocation process	There are only 3 kinds of protocol messages
Complexity of mechanism	High	Low

C. Node Synchronization

The synchronization process involves every node broadcasting its pool of IP addresses to keep record of IP address assignment in the entire network and detect any IP address leaks. This broadcast is used by every other node to update its maintenance table.

This detection only occurs when no free IP address is found during the requester_request period. The requester might fail to find an agent either because: 1) All IP addresses have been assigned to nodes currently in the MANET, or 2) some nodes have left the MANET without releasing their IP addresses and/or IP blocks (leaked-addresses). In the first scenario, no new node can be admitted (without expanding the address block range) as the MANET has reached its maximum size. In the second case, if a node leaves the network abruptly, or if the network partition happens, there are IP addresses that assigned to nodes which are no longer part of the network. We need to detect such IP address leaks, and then take corrective actions to reclaim those IP addresses.

Let there be node *A* wants to join the network and receives all the replies are negative before request_reply timer expires. Then it broadcasts leak_detect messages. In order to detect IP address leak every node scans that maintenance table for its state information, i.e., node ***B*** will scan the maintenance table for its agent_IP_address and requester_num as shown in Fig. 3. Then it informs the pre-agent its existence and waits for its requester's announcement. If node ***B*** discovers any requester's message missing, it concludes that this requester ***E*** departed without notification. It is easy to calculate the missing address range caused by ***E***. Node ***B*** then keeps node ***E***'s IP address block and takes the responsibility of allocating node ***A***. Otherwise, if no address is reclaimed, node ***A*** need to ask a new set of addresses from the system.

D. Discussion

In this section, we compare DAAP to other similar protocol in some features. Table 1 shows difference between them. The major difference is summarized as follows. 1) dynamic address pool, 2) small number of message, 3) low complexity without DAD and additional overhead.

IV. PERFORMACE EVALUATIONS

We used network simulator ns-2 as simulation tool to evaluate the performance of the protocol in terms of message complexity and latency.

A. Simulation Scenario and Parameters

The random waypoint mobility model was used. The speed of the nodes in the network was setting from 5 meters/second to 30 meters/second and the pause time was 10 seconds. The simulation duration was 5,000 seconds. The request_reply timer period was 5 seconds, and the requester_retry threshold was set to 3. The requester_num count was incremented by 1 after agent node successfully allocates a new node. In the simulation, the number of nodes varies from 60 to 100 nodes. The area of the networks was 500 * 500 m. The network was initialized with a single node. The inter-arrival time of new node was exponentially distributed with mean of 0.01 node arrivals/second. The inter departure time of nodes was exponentially distributed with a mean of 0.02 node departures/seconds. The address block size was varied from twice the node population to 10 times the node population. We used 60 and 100 node networks with 25 percent of node population generating CBR traffic to measure impact of traffic. The packet size was 512 bytes and rate varies 4 packets/sec to 20 packets/second.

We compare DAAP with MANETconf by implementing these protocols. The reason to choose MANETconf is because it guarantees unique address allocation and it is used for comparison in other researches so much time. Thus, it is so common to compare performance with MANETconf, which performance is considered as baseline. Furthermore, even though Buddy system is almost the same as DAAP, it cannot guarantee unique address allocation occasionally. Thus, comparison with buddy system is not accomplished.

B. Simulation Results

Fig. 4 shows a comparison of the average latency for address allocation to the number of nodes in terms of scalability. In DAAP, Fig. 4 shows the 95 percent confidence interval for mean latency for 60, 70, 80, 90 and 100 nodes. It was observed that around 95 percent of address allocations ware completed within 0.24 seconds at the requester had neighbors with nonempty IP blocks. A small fraction of the allocations require as much as 1.1 seconds where 100 nodes exist. This was when the new node had to perform a network-wide search for an agent.

Fig. 4 shows an increase in latency with increase in node population. This was because, as node population increased, for the same network density, the network diameter also increased. But, increase in latency is liner with respect to the increase in node population. This is because most of the addresses ware allocated locally. Only for few allocations, reclamations were needed. In MANETconf, the timeout to find an agent is almost as the same as DAAP, but DAAP has a better average of allocation latency than MANETconf because it doesn't need DAD to ensure unique IP address. As we known, MANETconf collected acknowledgement from all of MANET nodes in its list. MANETconf repeats the DAD procedure until the node succeeds in getting an unallocated address.

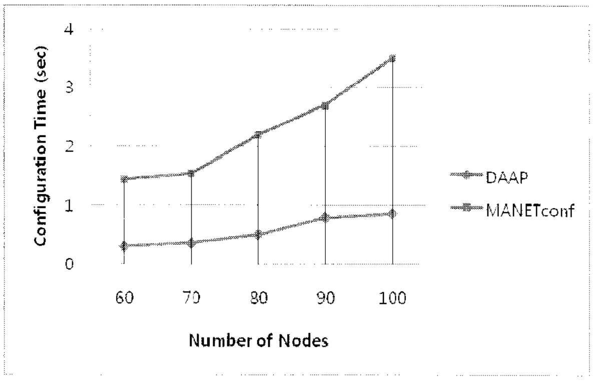


Fig. 4. Comparison of latency according to the number of nodes

Fig. 5 presents the comparisons of the average latency for address allocation according to the different mobility speed of nodes. As you can see, the configuration time increases as the speed gets faster in both of these two protocols. DAAP still performs better than MANETconf. Except the reason we mentioned above, DAAP has the advantage in managing the allocated addresses because every node in this approach is only concerned with its own pool of

IP addresses. To this extent, DAAP work like a stateless approach.

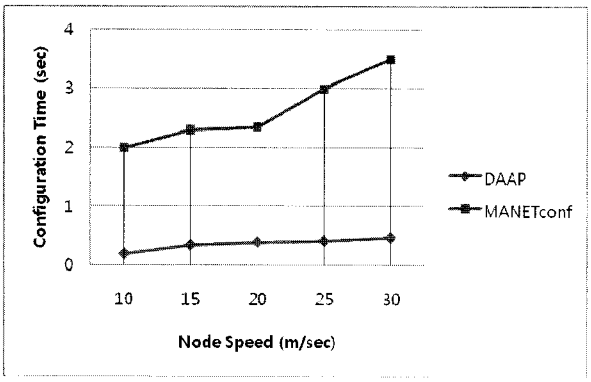


Fig. 5. Comparison of latency according to the node speed

Fig. 6 shows the traffic overhead according to the number of nodes. The result indicates that the number of packets is in proportion to the number of nodes in MANETconf. Regardless of the number of nodes, because a new node communications only with its agent to obtain its address. DAAP does not need DAD and reserved IP block is also helpful to reduce the number of DAD trials. Therefore, the total number of communication packets in DAAP is much smaller than in MANETconf. The node synchronization and reclamation of IP addresses mechanism used by DAAP aggravates communication overhead, but it has less impact than MANETconf because this mechanism is invoked if and only if all IP blocks are empty.

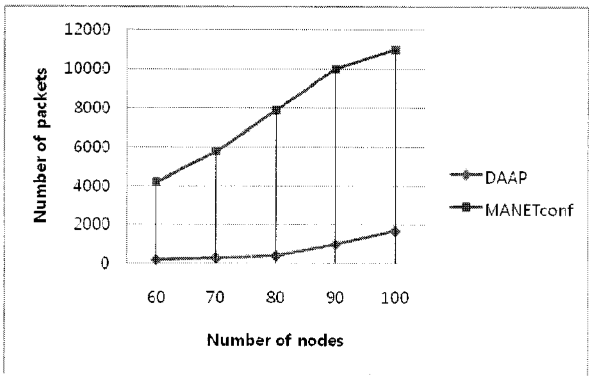


Fig. 6. Comparison of traffic overhead according to the number of nodes

We can observe the comparison of average communication overhead for address allocation according to different mobility of nodes in Fig. 7. The number of packets increases according to the node speed-up. That is because agent abrupt departure and network partition frequently happen as the node speed increases. Simulation result still proves that DAAP does well in this situation.

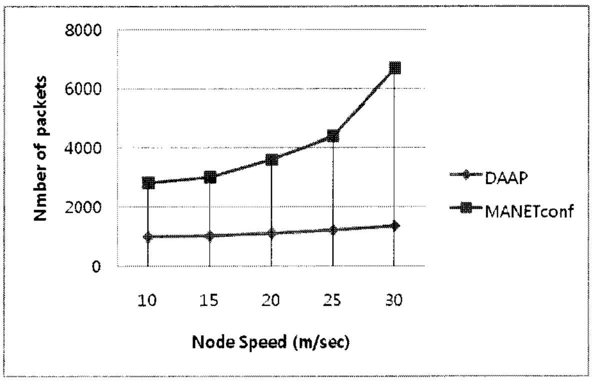


Fig. 7. Comparison of traffic overhead according to the node speed

V. CONCLUSIONS

We presented a simple and efficient protocol for dynamic allocation of nodes in MANETs. We have addressed unique IP address assignment to nodes in MANETs without centralized servers. Even though several solutions have been proposed however these approaches have different drawbacks. Our protocol is based on IP block splitting principle. The basic idea is to dynamically distribute the IP address blocks among the nodes in the network. Our approach guarantees unique IP address assignment under mobile ad hoc network conditions including network partitioning and merging.

Simulation results show that DAAP has low overheads, and is able to handle node arrivals and departures in an efficient way. We observed that the latency and communication overhead increased in a linear trend in node population and node mobility speed. Thus, the DAAP incurs low latency and communication overhead.

ACKNOWLEDGMENT

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-331-D00352).

REFERENCES

[1] R. Droms, “Dynamic host configuration protocol,” IETF RFC 2131, Mar.1997.

[2] Y. Sun and E. M. Belding-Royer. “Dynamic Address Configuration in Mobile Ad hoc Networks,” Technical Report, Computer Science Department, UCSB, Mar. 2003.

[3] Yuan-Ying Hsu and Chien-Chao Tseng, "Prime DHCP: A Prime Numbering Address Allocation Mechanism for MANETs" *IEEE Communications Letters*, Vol. 9, No. 8, Aug. 2005, pp. 712 – 714.

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

[5] H. Zhou, L. Ni, and M. Mutka, "Prophet Address Allocation for Large Scale MANETs," *Proc. of IEEE INFOCOM*, Mar. 2003.

[6] Z. Hu and B. Li "ZAL: Zero-Maintenance Address Allocation in Mobile Wireless Ad Hoc Networks," *Proc. of the 25th IEEE International Conference on Distributed Computing Systems*, Jun. 2005, pp. 103 – 112.

[7] M. R. Thoppian and R. Prakash, "A Distributed Protocol for Dynamic Address Assignment in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Vol. 5, Jan. 2006, pp. 4 – 19.

[8] M. Mohsin and R. Prakash, "IP Address Assignment in a Mobile Ad Hoc Network," *Proc. of IEEE MILCOM*, Oct. 2002,

[9] S. Nessargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," *Proc. of IEEE INFOCOM*, Jun. 2002, pp. 1059 -1068.

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

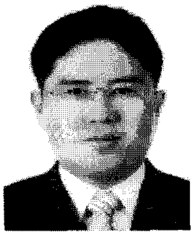
[11] N. Choi, C. K. Toh, Y. S. Lee, D. K. Kim and Y. H. Choi, "Random and Linear Address Allocation for Mobile Ad Hoc Networks," *Proc. of IEEE WCNC*, Mar. 2005.

[12] N. H. Kim, S. Y. Ahn and Y. H. Lee, "AROD: An address autoconfiguration with address reservation and optimistic duplicated address detection for mobile ad hoc networks," *Computer Communications*, Vol. 30, 2007, pp. 1913 – 1925.

[13] Y. Chen and E. Fleury, "Duplicate Address Detection in Wireless Ad Hoc Networks Using Wireless Nature," *Lectures Notes in Computer Science*, Vol. 3975, Feb. 2006.

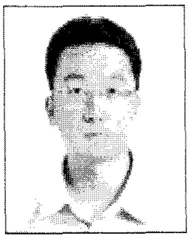
[14] K. A. Weinger, "Passive Autoconfiguration of Mobile Ad hoc Networks," *IEEE JSAC*, Vol. 23, Jan. 2005, pp. 507 – 519.

[15] N. H. Vadiya, "Weak Duplicate Address Detection in Mobile Ad Hoc Networks," *Proc. of ACM MOBIHOC*, Jun. 2002.



Ki-Il Kim received the M.S. and Ph.D. degrees in computer science from the ChungNam National University, Daejeon, Korea, in 2002 and 2005, respectively. He is currently with the Department of Informatics, Gyeongsang National University. His research interests include routing for MANET, QoS in wireless network, multicast, and sensor networks.

Bai Peng received the M.S. and in computer science from the Gyeongsang National University, Jinju, Korea, in 2008. His research interests include mobile ad hoc networks and sensor networks.



Kyong-Hoon Kim received his B.S., M.S., and Ph.D. degrees in Computer Science and Engineering from POSTECH, Korea, in 1998, 2000, and 2005, respectively. Since 2007, he has been a full-time lecturer at the Department of Informatics, Gyeongsang National University, Jinju, Korea. From 2005 to 2007, he was a post-doctoral research fellow at GRIDS lab in the Department of Computer Science and Software Engineering, the University of Melbourne. His research interest includes real-time systems, Grid computing, and security.