Name-Based Autoconfiguration for Mobile Ad hoc Networks

Namhoon Kim, Saehoon Kang, Younghee Lee, and Ben Lee

ABSTRACT—In a mobile ad hoc network, difficulties exist in supporting address autoconfiguration and naming resolution due to the lack of centralized servers. This letter presents a novel approach, called name-based autoconfiguration (NBA), which uses host names to determine IP addresses and provides address autoconfiguration and name resolution as a single protocol.

Keywords—Autoconfiguration, naming service, MANET.

I. Introduction

Addressing configuration is a necessary step to facilitate communications among hosts (or nodes), and a naming service allows users to conveniently use applications such as telnet, ftp, email, http, SIP, and file sharing. However, administration of these hosts becomes difficult as the number of hosts grows in a network region. This is especially the case for ubiquitous networks based on mobile ad hoc networks (MANETs) due to the lack of administrative infrastructure such as dynamic host configuration protocol and domain name system servers.

In prior studies of address autoconfiguration for MANETs, some researchers have suggested mechanisms to avoid address conflicts before joining a MANET [1]-[3]. Other studies are not involved with address configuration, but instead suggest solutions to detect and solve the address conflicts caused by network merging using a special key or relying on the aid of a routing protocol [4], [5].

Manuscript received July 21, 2005; revised Nov. 29, 2005.
This work was financially supported in part by the KOSEF under Grant No. R01-2003-000-10562-0.
Namhoon Kim (phone: +82 42 866 6251, email: nhkim@icu.ac.kr), Saehoon Kang (email: kang@icu.ac.kr), and Younghee Lee (email: yhlee@icu.ac.kr) are with the School of Engineering, Information and Communications University, Daejeon, Korea.
Ben Lee (email: benl@eecs.oregonstate.edu) is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, USA.

Naming resolution schemes for MANETs are classified as either distributed or partially distributed [6]. Existing naming resolution schemes assume each node already has a unique address and name, but they require a time intensive procedure and consume considerable bandwidth to guarantee uniqueness. As mobility increases, the probability of address changes increases. This requires more frequent name resolutions than for applications in infrastructure mode. The existing distributed name resolution methods require broadcasting, which causes considerable message overhead. Therefore, a name resolution scheme that eliminates or minimizes broadcast messages is highly desirable in a MANET.

Since an IP address and host name are closely related in terms of the applications, a novel autoconfiguration mechanism, called name-based autoconfiguration, is proposed to support both address configuration and naming resolution as a single protocol. This is achieved by allocating IP addresses based on hashed values of host names rather than randomly generated addresses, which alleviates the need to perform a naming service. Therefore, name-based autoconfiguration (NBA) saves the time required for configuration and minimizes communication overhead.

II. Name-Based Autoconfiguration (NBA)

NBA focuses mainly on name-to-address resolution, because name-to-address resolution is more popular than address-to-name resolution for user applications. For address-to-name resolution, the proposed scheme would require messages to be broadcasted just as in existing name resolution schemes. In addition, all the nodes are assumed to employ identical hash functions, such as Message Digest 5 (MD5) [7].

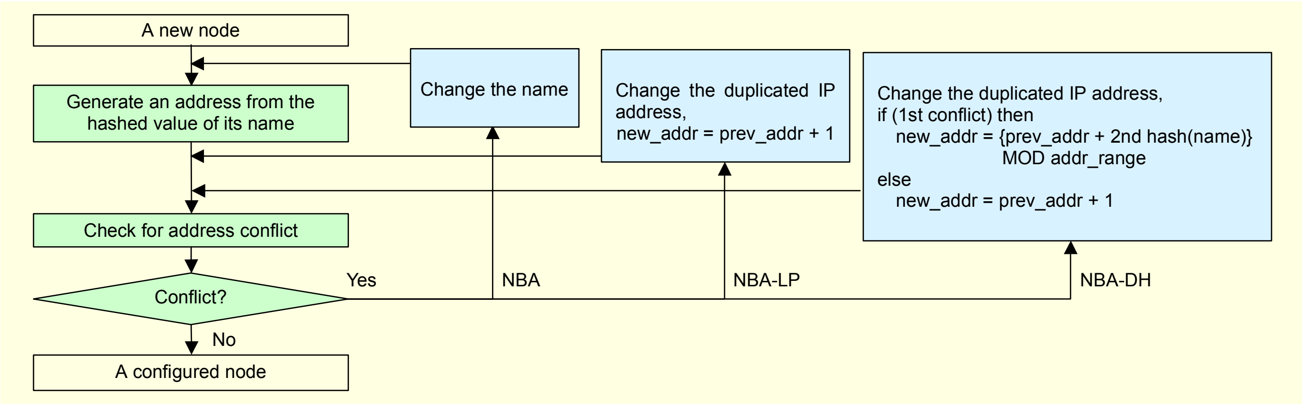

Fig. 1. Protocol description of NBA schemes.

1. NBA Operations

When a new node wants to join a MANET, it first selects a candidate address to configure its IP interface. The candidate address is obtained from the hashed value of its name. The new node then broadcasts the candidate address to check whether or not other nodes are already using this candidate address. If the address is not being used in the network, the new node can configure its IP interface with this address. Otherwise, the new node changes its name, obtains a new candidate address by hashing the new name, and checks for an address conflict. This procedure is repeated until the new node finds a unique IP address. Note that one simple way to perform a name change is to postfix a number to the name. For example, if the name of a new node is 'David-PDA' and the address obtained by hashing is already used, the name of the node will be changed to 'David-PDA1.' Figure 1 shows the protocol operations of NBA.

In NBA, name resolution is accomplished using the following steps. Since names of nodes may change due to an address conflict during the configuration phase, a node trying to resolve a name checks whether or not the correspondent node's name has changed by sending a probe packet. If the name has not changed, the name resolution is complete. Otherwise, the node broadcasts a message to find the changed name and address of the correspondent node.

2. Enhanced NBAs: NBA-LP and NBA-DH

The major challenge with NBA is that the hashing function used, that is, MD5, generates a 128-bit word while the address size in IPv4 or IPv6 is smaller. Thus, the number of bits in the hashed value must be reduced to match the address size. If the limited address range causes conflicts, the host name has to be changed. However, if the name is unique, it is advantageous to use it rather than requiring the host to alter its name. In order to eliminate the need to change host names, the following two enhancements based on hashing characteristics [8] have been incorporated into the basic NBA: (1) NBA with linear probing (NBA-LP) and (2) NBA with double hashing (NBA-DH).

As can be seen in Fig.1, the major difference between NBA and the two enhanced NBA schemes is that the latter do not unnecessarily change the host name when conflicts occur. In NBA-LP, if a new node causes an address conflict, instead of changing its name, the confirmation process is performed with another address, which is the previous address plus one. If a conflict occurs again, the confirmation process is repeated with the previous address plus one, and this process is repeated until the node finds a unique address. This causes a difference, which is referred to as a *displacement*, between the original address generated by the hash function and the final configured address. Thus, an exact match between the address and the hashed value of the name cannot be guaranteed.

Although NBA-LP avoids unnecessary host name changes, the cost of IP address resolution rises as the number of nodes increases due to the displacement problem. In comparison, NBA-DH employs two hash functions to reduce the likelihood of conflicts, where the second hash function is assumed to be secure hash algorithm (SHA1) [7]. Therefore, if a new node encounters a conflict during the addressing phase, the second hashed value of the name is generated. Then, the first value is added to the second value, and the result is adjusted to the fixed address size using a modular function. Finally, the node performs the confirmation process with the adjusted value. If the host encounters an address conflict again, it simply defers to NBA-LP.

III. Simulation Results

1. Simulation Environment

In order to study the performance of NBA, NBA-LP, and NBA-DH in terms of configuration time and communication overhead, simulations were performed using ns-2 (version

2.27) with modified random waypoint models [9]. The minimum and maximum speeds of nodes are 1 m/s and 5 m/s, respectively, and the pause time is set to 0 seconds. The address range is 1 to 127, where the first 15 addresses are used as temporary addresses for newly arriving nodes. Thus, the total number of available addresses is 112. The network size is 1000 m by 1000 m, and an ad hoc on-demand distance vector (AODV) was used as the routing protocol. The host names are randomly generated, and the simple convention of postfixing a number to the name was used to change host names (see section II.1). MD5 and SHA1 are well-made hash functions that generate 128-bit words, thus random names do not affect the number of address conflicts. The performance of NBA was compared against a hypothetical system that employs strong duplicated address detection (DAD) and a distributed naming resolution (DNR) mechanism (referred to as SDAD+DNR) [6].

2. Address and Name Configuration Cost

A. Average Configuration Time

Typically, three consecutive DAD processes are performed due to the possibility of message losses. After a new node succeeds in the DAD process, it can configure its IP interface [1]. Therefore, configuration time for the address and name depends on the number of conflicts. If an address conflict is encountered, the host must retry the DAD process with another address. Figure 2 shows that the three NBA-based schemes require at least 3 seconds for obtaining an address because the DAD timeout was assumed to be 1 second based on the assumption that the maximum hop count is 10 and the hop traversal time is 0.1 seconds. The performance of SDAD+DNR is worse than others because it needs to perform a name configuration, which takes as much time as an address configuration, while NBA-based schemes do not need to perform a name configuration. Among the three NBA-based methods, NBA shows the best performance, and the enhanced NBA schemes performed worse than NBA due to the number of address conflicts caused by displacement.

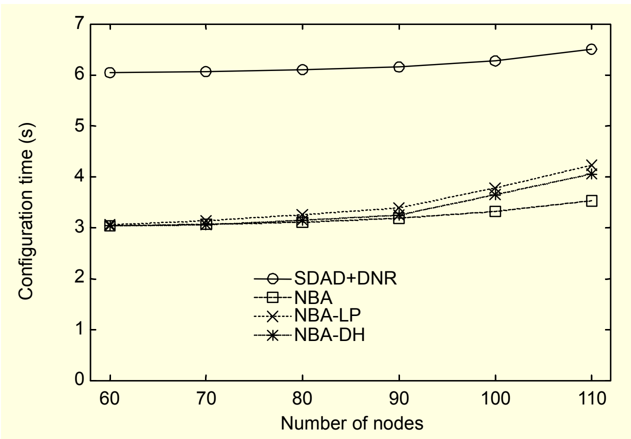

Fig. 2. Address and name configuration time.

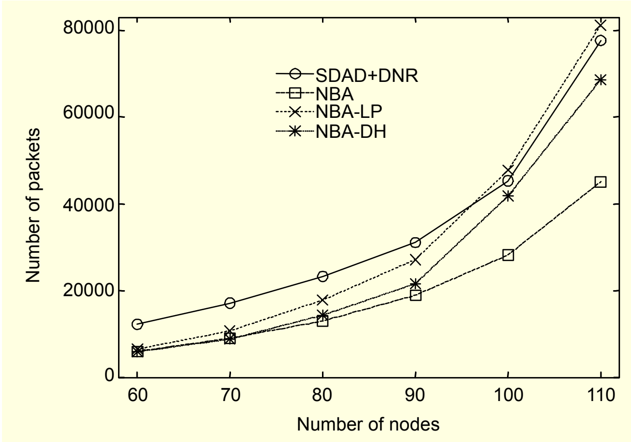

Fig. 3. Communication overhead for configuration.

B. Communication Overheads

Communication overhead depends on the total number of nodes and the number of address conflicts. Each node produces packets for address and name configurations; therefore, more packets are required as the number of nodes increases. The number of address conflicts also affects communication overhead because every node must perform DAD again if it encounters an address conflict. These results are shown in Fig. 3. In the range of 60 to 90 nodes, the number of packets required for SDAD+DNR is the worst because it needs to perform a name configuration. The overhead for NBA-LP is similar to NBA up to around 70 nodes. Beyond 70 nodes, the overhead increases rapidly due to the increase in the number of conflicts caused by displacement as well as the decrease in the number of available addresses. Moreover, the overhead for NBA-LP becomes worse than SDAD+DNR when the number of nodes is more than 90. Overall, the overhead required for NBA-DH is very similar to NBA in the range of 60 to 90 nodes. The overhead for NBA-DH also increases rapidly beyond 90 nodes due to displacement. However, the amount of overhead is less than NBA-LP or SDAD+DNR.

3. Name Resolution Cost

Table 1 summarizes the number of packets required to perform naming resolution for each method. For DNR, when a host wants to resolve the name of the correspondent node, a message with its name is broadcasted. The correspondent node then responds to the message with its address. In NBA, if the host name does not need to be changed, two unicast messages are sufficient to resolve the name. Otherwise, it additionally requires a broadcast message and a unicast message.

In contrast, broadcasting is not required in NBA-LP and NBA-DH. However, since the original hashed value may not be the same as the actual address of the correspondent node, unicast messages need to be sent to trace the address, where the number of messages required depends on the displacement.

In NBA, $p = (\sum_{i=0}^{n-1} \frac{i}{R})/n$ is the probability that a node changes its name, where R is the total number of addresses, and n is the number of nodes.

For NBA-LP and NBA-DH, the load factor a is given by n/R, where n is the number of nodes and R is the total number of addresses [8]. For example, if the number of nodes is 64 and the total number of addresses is 128, then the load factor is 0.5. Therefore, a node can find its correspondent address with an average of 1.5 trials in NBA-LP and 1.28 trails in NBA-DH. The expression is then multiplied by 2 to consider responses as well as probe messages.

Figure 4 shows the number of required messages to resolve its correspondent node's name as a function of the number of nodes. DNR incurs more cost than others because it uses message broadcasting. NBA requires a fewer number of messages than DNR because only the nodes that change their names send broadcast messages. The overhead for NBA-LP worsens as the number of nodes increases because the load factor is close to 1.

Table 1. Number of messages required for resolving a name.

Protocol	Number of messages
DNR	1 Broadcast + 1 Unicast
NBA	$(1-p)$.2 Unicast + p. (1 Broadcast + 3 Unicast)
NBA-LP	$(1+\frac{1}{1-a})$ Unicast
NBA-DH	$-\frac{2}{a}\ln(1-a)$ Unicast

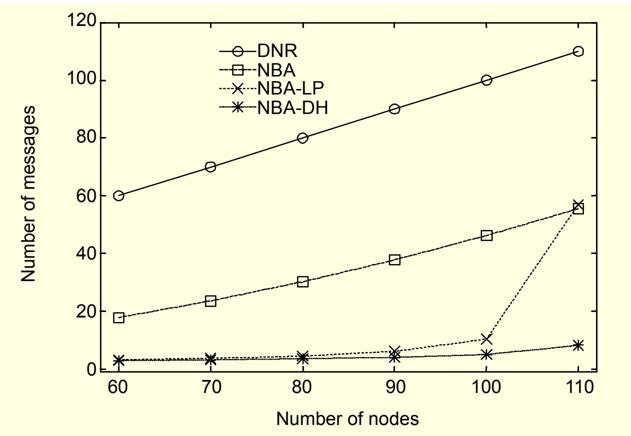

Fig. 4. Average messages for resolving a name.

IV. Conclusion

The proposed NBA scheme provides simultaneously an address configuration and naming service for a MANET. Thus, NBA reduces the total number of messages and the average latency to perform address allocation and naming resolution. Moreover, users can conveniently use the naming service due to early binding between the host name and IP address. This letter also proposes two enhancements, NBA-LP and NBA-DH, to solve the problem in which a host name needs to be changed even though the name is unique within the network. Although NBA-LP and NBA-DH incur some overhead for configuring the address due to displacement, they significantly reduce the number of needed messages for performing name-to-address resolution compared to DNR because they do not require broadcast messages. NBA-DH shows better results than NBA-LP as the number of available addresses decreases. Moreover, in terms of name resolution cost, NBA-DH gives better performance due to lower displacement than NBA-LP.

References

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

[2] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," *INFOCOM 2002*, vol. 2, June 2002, pp. 1059-1068.

[3] Hongbo Zhou, Lionel Ni, and Matt Mutka, "Prophet Address Allocation for Large Scale MANETs," *INFOCOM 2003*, vol. 2, Mar. 2003, pp. 1304-1311.

[4] Nitin Vaidya, "Weak Duplicate Address Detection in Mobile Ad Hoc Networks," *ACM MobiHoc 2002*, June 2002.

[5] K. Weniger, "Passive Duplicate Address Detection in Mobile Ad hoc Networks," *WCNC 2003*, New Orleans, USA, Mar. 2003.

[6] P. Engelstad, D. Thanh, and T. Jonvik, "Name Resolution in Mobile Ad hoc Networks," *ICT 2003*. vol. 1, Feb. 2003, pp. 388-392.

[7] Charlie Kaufman, Radia Perlman, and Mike Speciner, *Network Security, Private Communication in a Public World*, Prentice Hall, 2002, pp. 118-146.

[8] Knuth, Sorting and Searching, *The Art of Computer Programming*, Addison Wesley, vol. 3, 1973, pp 506-549.

[9] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," *INFOCOM 2003*, vol. 2, April 2003, pp. 1312-1321.