Name Directory Service based on MAODV and Multicast DNS for IPv6 MANET

Jaehoon Jeong, Jungsoo Park and Hyoungjun Kim
Protocol Engineering Center, ETRI,
161 Gajeong-dong Yuseong-gu, Daejeon, 305–350 Korea
Email: {paul,pjs,khj}@etri.re.kr
Telephone: +82–42–860–1664, Fax: +82–42–861–5404
WWW home page: http://www.adhoc.6ants.net/

***Abstract*— This paper proposes a name service called name directory announcing the information of neighbors who are reachable in IPv6 mobile ad hoc network partition. Neighbor information consists of ad hoc user profile, DNS name and IPv6 address. Through name directory, ad hoc users can know who is reachable in the connected network partition and connect to the user's mobile node with host DNS name.**

I. Introduction

Mobile Ad Hoc Network (MANET) consists of mobile nodes that take part in routing so as to communicate with one another in the environment where there is no communication infrastructure. Small Office Home Office (SOHO) networks, home-networks, and internal networks of transport vehicles such as airplane, train and bus can be constructed with MANET networking. Up to recently, MANET networking has been researched mainly on ad hoc routing protocols. However, for the deployment of MANET networking, other services are necessary, such as MANET IP address autoconfiguration and MANET DNS service [1]–[4].

In this paper, we focus on name service including DNS service in MANET, where the current dedicated DNS is impossible to adopt in order to provide name service for mobile nodes. Because MANET has dynamic network topology, the current DNS cannot be adopted in MANET, which has the hierarchical structure and consists of dedicated servers. Name Directory (NDR) proposed in this paper not only provides DNS name service in multi-hop MANET, but also allows ad hoc user of mobile node to perceive the neighbors and their user information that is necessary to decide whether or not the neighbor is the man with whom the user wants to communicate. We also suggest an autoconfiguration (zero-configuration) technology for generating DNS zone file associating host DNS name with autoconfigured IPv6 address. The naming suggested in this paper is useful in ad hoc network where there is no network manager to assign each user a unique domain name and there can exist mobile nodes with duplicate host DNS names. Accordingly, we suggest an Ad Hoc Name Service System (ANS) for MANET DNS and NDR service in this paper: (a) Auto-generation of DNS zone file, (b) Name-to-address translation, and (c) Collection of user information of neighbor nodes.

The remainder of the paper is organized as follows. In Section II, related work is presented. In Section III and IV, respectively, Multicast DNS and NDR are explained in detail. In Section V, we explain our testbed for testing NDR in IPv6 MANET. Finally, in Section VI, we conclude the paper with future research work.

II. Related Work

A. MANET DNS Service

An architecture of DNS service system is specified for mobile ad hoc network which might be connected to the Internet [1], [3], [4]. The resolution of DNS names of mobile nodes within mobile ad hoc network is performed by multicast DNS and that of DNS names of nodes in the Internet is performed through DNS autoconfiguration of recursive DNS server. In the former, each mobile node plays a role of DNS name server for the DNS resource records associated with DNS name of which authority it has. The latter allows mobile node to receive the global Internet service, such as web service, by providing global DNS resolution in mobile ad hoc network connected to the Internet. These two kinds of DNS name resolution are processed automatically without the intervention of users in mobile ad hoc network. Also, we can authenticate DNS message on the basis of "Secret Key Transaction Authentication for DNS" resource record called TSIG resource record [1], [3]–[5]. We can provide ad hoc users with service discovery based on multicast DNS and "Service Location" resource record called SRV resource record [4], [6].

B. Multicast Ad Hoc On-Demand Distance Vector (MAODV)

Multicast Ad Hoc On-Demand Distance Vector (MAODV) routing protocol is an extension of AODV routing protocol for ad hoc multicast routing [7], [8]. MAODV is a shared tree based multicast routing protocol like PIM-SM [9]. When a new group member joins a multicast tree corresponding to a multicast group, it finds a shortest path to the tree through AODV route discovery [7]. In every multicast tree, a group leader is selected to broadcast a Group-Hello (GRPH) message throughout the whole network to indicate the existence of that group and its current status. When receiving a GRPH message, each node updates its Group Leader Table which indicates the group leader and the route towards the group leader. Even

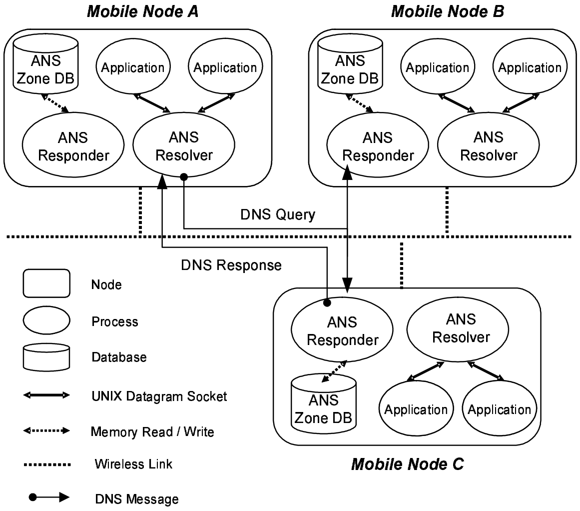


Fig. 1. DNS Name Resolution through Ad Hoc Name Service System (ANS)

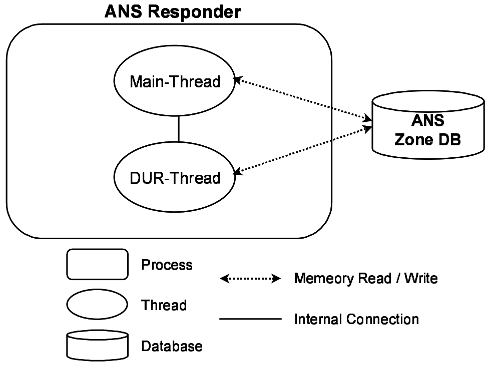


Fig. 2. Architecture of ANS Responder

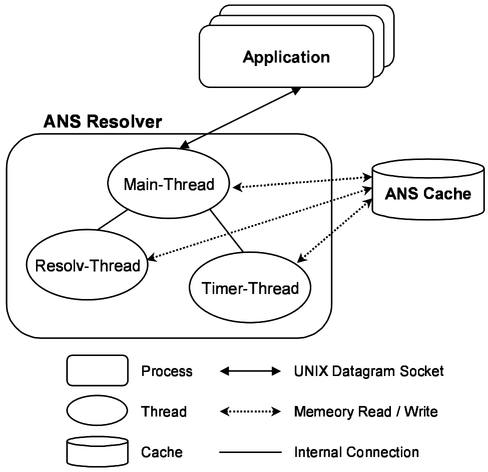


Fig. 3. Architecture of ANS Resolver

when the node is not a tree member, it rebroadcasts the first-time received GRPH message. This GRPH message is used in order to merge partitioned multicast trees.

III. Ad Hoc Name Service System (ANS) for Multicast DNS Service in IPv6 MANET

We developed Ad Hoc Name Service System for IPv6 MANET (ANS) that provides the name resolution and service discovery in IPv6 MANET which is site-local scoped network [1]. Every network interface of mobile node can be configured automatically to have site-local scoped IPv6 unicast address by IPv6 ad hoc address autoconfiguration. ANS System consists of ANS Responder that works as DNS name server in MANET and ANS Resolver that performs the role of DNS resolver for name-to-address translation. Mobile node registers an AAAA type DNS Resource Record (RR) of combining its unicast address and host DNS name with DNS zone file of its ANS Responder (ANS Zone File). Fig. 1 shows the architecture of ANS System for name service in MANET and DNS name resolution through ANS. Each mobile node runs ANS Responder and Resolver. An application over mobile node that needs the name resolution can get the name service through ANS Resolver because ANS provides the applications with the library functions for name resolution through which they can communicate with their ANS Resolver through UNIX datagram socket.

In Fig. 1, ANS Resolver of mobile node A sends DNS query in ANS multicast address, “ff05::224.0.0.251” or “ff05::e000:00fb”, which all ANS Responder should join for receiving DNS query [3]. When ANS Responder receives DNS query from ANS Resolver in other mobile nodes, after checking if it is responsible for the query, it decides to respond to the query. When it is responsible for the query, it sends the appropriate response to ANS Resolver in unicast. In Fig. 1, mobile node C responds to DNS query of mobile node A.

A. Architecture and Operation of ANS System

1) Architecture and Operation of ANS Responder: Fig. 2 shows the architecture of ANS Responder, which is composed of Main-Thread and DUR-Thread.

Main-Thread manages ANS Zone database (DB) for name service and processes DNS queries to send the corresponding response to the querier. It initializes ANS Zone file that contains DNS resource records into ANS Zone DB. When it receives a DNS query, it checks if it is responsible for the query. If it is responsible, it sends the response corresponding to the query to ANS Resolver that sent the query.

DUR-Thread performs the dynamic update request (DUR) during the verification of the uniqueness of DNS resource record [3]. The verification is initiated by ANS Resolver on another node that has receivied multiple responses with the same domain name and resource record type for the DNS query that it sent in multicast. The destination address of the multicast packet for the verification is also ANS multicast address, “ff05::224.0.0.251”. The ANS Resolver sends the first response to every ANS Responder that sent a response except the Responder that sent a response first. Every ANS Responder that receives a response managed by itself performs the verification of the uniqueness of the resource record

```
user-id = paul

domain = manet.

ttl = 10

user-name = Jaehoon Jeong

affiliation = ETRI

email-address = paul@etri.re.kr

phone-number = +82-16-711-1765

description = A researcher on IPv6 MANET networking
```

(a) NDR Configuration File (NDR.conf)

```
paul.manet.  10  IN  UI  $Jaehoon Jeong$ETRI$paul@etri.re.kr
                         $+82-16-711-1765$A researcher on
                         IPv6 MANET networking$
```

(b) User Information (UI) Resource Record

Fig. 4. Configuration File and Resource Record for NDR

included in the response through DUR-Thread. If DUR-Thread detects the duplication of the resource record, it invalidates the record in its ANS Zone DB.

2) Architecture and Operation of ANS Resolver: Fig. 3 shows the architecture of ANS Resolver, which consists of Main-Thread, Resolv-Thread and Timer-Thread.

When Main-Thread receives DNS query from application on the same node through UNIX datagram socket, it first checks if there is the valid response corresponding to the query in ANS Cache. If there is the response, Main-Thread sends the response to the application. Otherwise, it executes Resolv-Thread that will perform the actual name resolution and asks Resolv-Thread to respond to the application through the name resolution.

When Resolv-Thread receives the request of name resolution from Main-Thread, it makes DNS query message and then sends the message in ANS multicast address, "ff05::224.0.0.251". If Resolv-Thread receives a response message from an ANS Responder, it returns the result of the response to the application that asked for the name resolution through UNIX datagram socket. Whenever a new resource record is received by Resolv-Thread, it caches the response in ANS Cache. When a record is registered in ANS Cache, ANS Cache timer is adjusted for ANS Cache management. If Resolv-Thread receives the multiple responses for the query, it initiates the dynamic update request in the responders that sent the same response except the 1st responder.

Whenever ANS Cache timer expires, Timer-Thread checks if there are entries that expired in ANS Cache. Timer-Thread invalidates the entries and makes the resource records of the entries unusable any more for name resolution. After the work, Timer-Thread restarts ANS Cache timer.

IV. Name Directory Service

A. User Information (UI) Resource Record

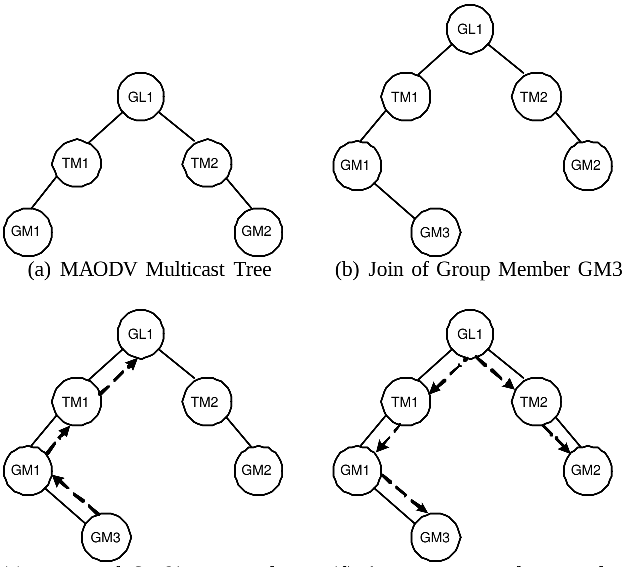
We define a new DNS resource record for User Information (UI). A DNS resource record is comprised of five fields: Name, TTL, Class, Type and Rdata. For a UI resource record, Name field is host DNS name of user's node. TTL field is caching time of this record. Class field is set to "IN", meaning Internet. Type field is set to "UI". Rdata contains user's information: (a) User name, (b) Affiliation, (c) Email address, (d) Phone number, and (e) Description. Each item is separated by a delimiter '$'. This user information is stored in NDR configuration file (NDR.conf). With NDR.conf, besides the above five fields, User ID and Domain are registered for making host DNS name like in Fig. 4 (a). Fig. 4 (b) shows UI resource record made out of NDR.conf of Fig. 4 (a).

B. Procedure of Name Directory Service

We assume that IPv6 MAODV is used for MANET multicast routing. When an ad hoc user joins MANET DNS service and wants to know what users are neighbors reachable in MANET partition, he should start NDR program. We will explain NDR service through Fig. 5. Like in Fig. 5 (a), five mobile nodes construct an ANS multicast tree: Group Leader GL1, Group Member GM1 through GM2, and Tree Member TM1 through TM2. In Fig. 5 (b), a new group member GM3 joins the ANS multicast tree. After joining the tree, GM3 receives broadcast GRPH message including GL1's IPv6 address. GM3 periodically unicasts its UI resource record to GL1 through unsolicited DNS response message like in Fig. 5 (c) and GL1 stores the UI resource record in its ANS Zone DB. Tree members, TM1 and TM2, forward to their upstream node the user information of their downstream nodes which are the group members of ANS multicast tree. Like this, a group leader of ANS multicast tree can gather user information of group members. Like in Fig. 5 (d), GL1 periodically multicasts neighbor directory including UI resource records throughout ANS multicast tree. When each group member receives UI resource records of neighbor directory and stores them in their own ANS Zone DB. NDR process running on GM3 can display neighbor directory stored in its local ANS Zone DB. Therefore, a group leader maintains neighbor information within its multicast tree and each group member can know who are reachable in the MANET partition. When a user can connect to another user with host DNS name, it can resolve the name into an IPv6 address through ANS Resolver. For the optimization of name resolution, we can piggyback AAAA resource record including IPv6 address associated with host DNS name of UI resource record through DNS additional section of DNS response message [10].

C. Optimization of Name Directory Service

We can optimize NDR service in the two aspects: (a) Periodic report of group members, and (b)Periodic announcement of name directory by group leader.



(a) MAODV Multicast Tree (b) Join of Group Member GM3

(c) Report of GM3's User Information to Group Leader GL1 (d) Announcement of User Information by GL1

Fig. 5. NDR Service over MAODV Multicast Tree

1) Optimization of Periodic Report of User Information: Only leaf group members of ANS multicast tree voluntarily report their UI resource records. When upstream group member node receives UI resource record from downstream node, it aggregates the received UI resource record with its UI resource record and forwards the records to its upstream node towards group leader. Tree members which consist of ANS multicast tree only forward aggregated user information to upstream node without adding their user information, because they are not group members. Therefore, the group leader can get an aggregated packet of UI resource records instead of each individual packet of group members.

2) Optimization of Periodic Announcement of Name Directory: Group leader announces total name directory of active nodes and partial name directory of out-of-range nodes at different periods. That is, group leader advertises the total name directory less frequently than the partial name directory. When a group member receives the partial name directory, it invalidates the entries of the partial name directory in its name directory. These separate announcements can reduce the amount of traffic advertised for name directory.

V. Experiment in Testbed

We have implemented IPv6 AODV and MAODV as ad hoc unicast and multicast routing protocols, which have been extended for the support of IPv6, on the basis of NIST AODV [7], [8], [11], [12]. These ad hoc routing protocols have been implemented in Linux kernel 2.4.22 version. Also, we have developed IPv6 Wireless Mobile Router (WR) for MANET testbed shown in Fig. 6, which is a small box with IEEE 802.11g interface and embedded linux of kernel version 2.4.22. In order that we can set up multi-hop MANET testbed and handle the topology easily, we have made the box regulate the signal range by controlling Rx and Tx power level of the wireless interface. In addition, we have implemented MAC filtering in wireless interface driver in order to filter adjacent node's packet in MAC level. With the Rx/Tx power control and MAC filtering, we can handle MANET topology at more liberty. With mobile routers, we tested the operation of ANS and NDR, changing the network topology variously.

Fig. 6. IPv6 Wireless Mobile Router

VI. Conclusion

In this paper, we suggest a new useful application called name directory, which can be categorized as peer-to-peer application in MANET. Through NDR, ad hoc users can notice who are reachable in MANET partition where they are placed. For future work, we will enhance our scheme of name directory service through ns-2 simulation [13].

References

[1] Jaehoon Jeong, Jungsoo Park and Hyoungjun Kim, "Auto-Networking Technologies for IPv6 Mobile Ad Hoc Networks", ICOIN 2004, February 2004.
[2] Jaehoon Jeong, Jungsoo Park, Hyoungjun Kim and Dongkyun Kim, "Ad Hoc IP Address Autoconfiguration", draft-jeong-adhoc-ip-addr-autoconf-02.txt, February 2004.
[3] Jaehoon Jeong, Jungsoo Park, Hyoungjun Kim and Kishik Park, "Name Service in IPv6 Mobile Ad-hoc Network", ICOIN 2003, February 2003.
[4] Jaehoon Jeong, Jungsoo Park and Hyoungjun Kim, "DNS Service for Mobile Ad Hoc Networks", draft-jeong-manet-dns-service-00.txt, February 2004.
[5] P. Vixie, O. Gudmundsson, D. Eastlake and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.
[6] A. Gulbrandsen, P. Vixie and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.
[7] E. Belding-Royer and C. Perkins, "Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing", draft-ietf-manet-maodv-00.txt, July 2000.
[8] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003.
[9] Deborah Estrin et al., "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, June 1998.
[10] P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC 1035, November 1987.
[11] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing for IP Version 6", draft-perkins-manet-aodv6-01.txt, November 2001.
[12] IPv6 AODV and MAODV Implementation, http://www.adhoc.6ants.net/
[13] Network Simulator: ns-2, http://www.isi.edu/nsnam/ns/