Name Service in IPv6 Mobile Ad-hoc Network

Jaehoon Jeong, Jungsoo Park, Hyoungjun Kim, and Kishik Park

Protocol Engineering Center, ETRI, 161 Gajong-Dong, Yusong-Gu,
Daejon 305-350, Korea
{paul,pjs,khj,kipark}@etri.re.kr,
WWW home page: http://www.adhoc.6ants.net/

Abstract. In this paper, we propose an architecture of name service system which can provide mobile nodes in IPv6 mobile ad-hoc network with the name-to-address resolution and service discovery. Because mobile ad-hoc network has dynamic topology, the current DNS is not appropriate to name service in mobile ad-hoc network. We suggest the design and implementation of name service system suitable for the mobile ad-hoc network, autoconfiguration technology related to name service, and service discovery based on the name service system.

1 Introduction

Mobile Ad-hoc Network (MANET) is the network where mobile nodes can communicate with one another without communication infrastructure such as base station or access point. Recently, according as the necessity of MANET increases, ad-hoc routing protocols for multi-hop MANET have been being developed by IETF Manet working group [1]. With this trend, IPv6 that has many convenient functions including stateless address autoconfiguration [2] and multicast address allocation [3] has become mature and been being deployed in the whole world. The users in MANET will be able to communicate more easily through the IPv6 zero-configuration that provides easy configuration [3, 6]. Accordingly, if we adopt IPv6 as the network protocol of MANET, we will create a number of useful services for MANET.

DNS is one of the most popular applications in the Internet. It provides the name-to-address resolution among nodes in the Internet. DNS must be a necessity of MANET but the current DNS is inappropriate to MANET that has dynamic topology because the current DNS works on the basis of dedicated and fixed name servers.

In this paper, we propose an architecture of name service system which can provide mobile nodes in IPv6 mobile ad-hoc network with the name-to-address resolution and autoconfiguration technology for easy configuration related to name service including the generation of unique domain name of mobile node and generation of zone file for name service. We also suggest service discovery performed through the name service system of this paper and DNS service resource record (SRV) [5].

This paper is organized as follows; Sect. 2 presents related work. In Sect. 3, we explain the architecture of the suggested name service system, Ad-hoc Name Service System for IPv6 MANET (ANS), operation of ANS System, and name service through ANS. In Sect. 4, we show the result of name resolution through ANS in MANET testbed. Finally, in Sect. 5, we conclude this paper and present future work.

2 Related Work

2.1 Autoconfiguration Technology for Zero-Configuration Networking

IETF Zeroconf working group has defined the technology by which the configuration necessary for networking is performed automatically without manual administration or configuration in the environment such as small office home office (SOHO) networks, airplane networks and home networks, which is called zero-configuration or auto-configuration [6]. The main mechanisms related to the autoconfiguration technology are as follows; (a) IP interface configuration, (b) Name service (e.g., Translation between host name and IP address), (c) IP multicast address allocation, and (d) Service discovery.

2.2 Link-Local Multicast Name Resolution

Link-Local Multicast Name Resolution (LLMNR) has been devised for the resolution between domain name and IP address in the link-local scoped network [4]. The procedure of the resolution from domain name to IPv6 address in a subnet is as follows; Sender is the resolver that sends LLMNR query in link-local multicast and Responder is the name server that sends the LLMNR response to Sender in unicast. When Sender receives the response, it verifies if the response is valid. If the response is valid, Sender stores it in LLMNR cache and passes the response to the application that initiated the DNS query. Otherwise, Sender ignores the response and continues to wait for other responses.

3 Ad-hoc Name Service System for IPv6 MANET (ANS)

ANS is the name service system that provides the name resolution and service discovery in IPv6 MANET which is site-local scoped network. We assume that every network device of mobile node can be configured automatically to have site-local scoped IPv6 unicast address by ad-hoc stateless address autoconfiguration [3]. In this section, we described the architecture of ANS System, operation of ANS system and name service through ANS.

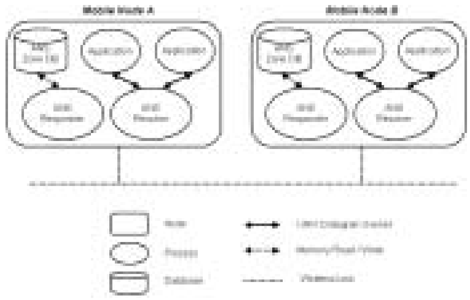

Fig. 1. ANS System for Name Service in IPv6 MANET

3.1 ANS System

ANS System consists of ANS Responder that works as name server in MANET and ANS Resolver that performs the role of DNS resolver for name-to-address translation. Fig. 1 shows the architecture of ANS System for name service in MANET. 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 name resolution functions through which they can communicate with ANS Resolver for name resolution.

ANS Resolver sends DNS query in the multicast address that ANS Responder in each mobile node has joined for name service. 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.

3.2 Architecture of ANS System

Architecture 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. The verification is initiated by ANS Resolver on the other 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 ANS Resolver sends the first response to every ANS Responder that sent a response except the Responder that sent a response first. Every

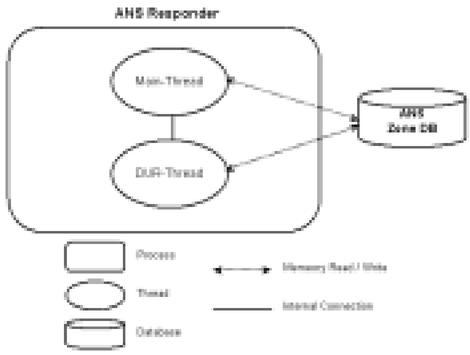

Fig. 2. Architecture of ANS Responder

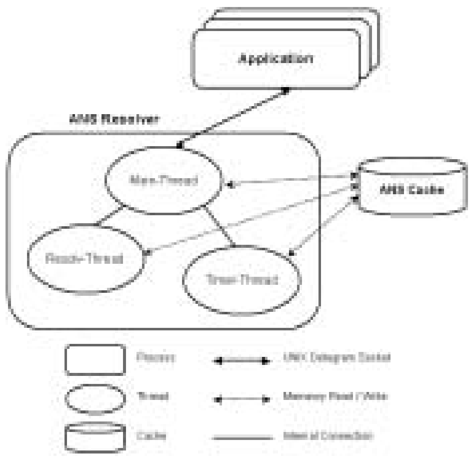

Fig. 3. Architecture of ANS Resolver

ANS Responder that receives a response managed by itself performs the verification of the uniqueness of the resource record related to the response through DUR-Thread. If DUR-Thread detects the duplication of the resource record, it invalidates the record in its ANS Zone DB.

Architecture 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 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 destination multicast address corresponding to the domain name of the query and then sends the message in the multicast address. If Resolv-Thread receives a response message from an ANS Responder, it returns the the result of the response to the application that asked 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.

Whenever ANS Cache timer expires, Timer-Thread checks if there are entries that expired in ANS Cache. Timer-Thread invalidates the entries in order to make the resource records of the expired entries unused for name resolution. After the work, Timer-Thread restarts ANS Cache timer.

3.3 Operation of ANS System

Operation of ANS Responder. ANS Responder performs the name service running as daemon process. ANS Responder operates in 9 steps as follows;

1) ANS Responder starts as daemon process.
2) ANS Responder generates a unique domain name per network device. Without the intervention of network manager to manage DNS, mobile node can generate a unique domain name per network device automatically on the basis of network device identifier [8]. We describe the generation of unique domain name in detail in Sect. 3.4.
3) ANS Responder generates zone file for name service. We explain the generation of zone file in detail in Sect. 3.4.
4) ANS Responder loads resource records of zone file into ANS Zone DB.
5) ANS Responder verifies the uniqueness of the resource records related to each domain name through dynamic update request [4, 7]. Fig. 4 shows the dynamic update request during the initialization of ANS Zone DB. Node 1 verifies the uniqueness of its resource records through dynamic update request. DUR query message is sent in multicast of which address is IPv6 site-local solicited name multicast address. The multicast address is generated through the procedure of Fig. 5. In the example of Fig. 4, Node 1 can register only RR-1 (Resource Record 1) in ANS Zone DB as valid resource record because RR-2 has been already managed by Node 2 and can not be managed by Node 1.
6) ANS Responder joins multicast group corresponding to each domain name [4]. The multicast address corresponding to each multicast group is generated through the procedure of Fig. 5. Because the multicast address which is used for sending and receiving the DNS query related to the domain name can be determined with the domain name, we can reduce the number of the packets flooded for DNS query.
7) ANS Responder waits for DNS message.
8) When ANS Responder receives a DNS message, it checks if the message is a DNS query.

9) If the DNS message is a query, ANS Responder processes the query and sends the response to ANS Resolver that sent the query and goes to Step 7. If the DNS message is a response of which the name belongs to this node, ANS Responder performs the dynamic update request to verify the uniqueness of the resource record related to the name [4]. If ANS Responder finds the name conflict through the dynamic update request, it invalidates the resource record corresponding to the duplicate name in its ANS Zone DB. After performing the dynamic update request, ANS Responder goes to Step 7.

Operation of ANS Resolver. ANS Resolver processes the DNS query as daemon process. ANS Resolver operates in 14 steps as follows;

1) ANS Resolver starts as daemon process.

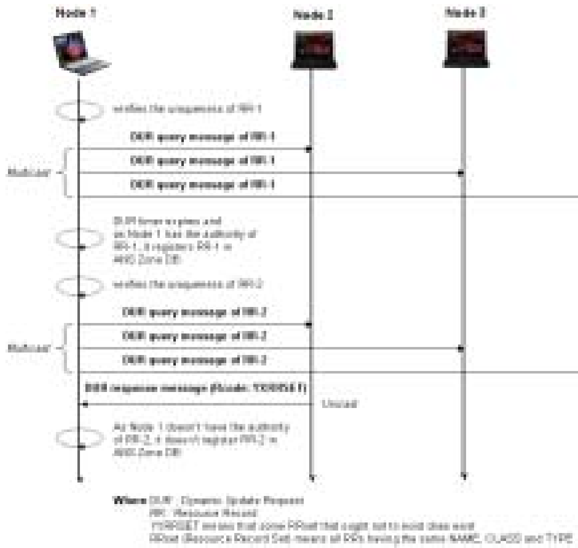

Fig. 4. Dynamic Update Request during the Initialization of ANS Zone Database

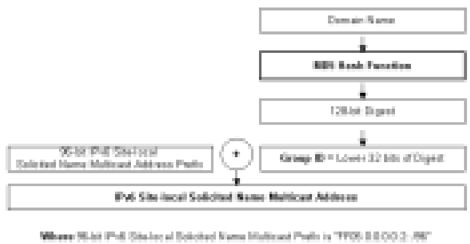

Fig. 5. Generation of IPv6 Site-local Solicited Name Multicast Address

2) ANS Resolver waits for the request of name resolution from an application on the same node.
3) When ANS Resolver receives the resolution request, it checks first if there is the result of the request in ANS Cache.
4) If there has been already the result of the request in ANS Cache, ANS Resolver passes the result to the application.
5) Unless there is the result of the request in ANS Cache, ANS Resolver creates Resolv-Thread that will perform name resolution and passes the request to Resolv-Thread.
6) Resolv-Thread receives the name resolution request from Main-Thread.
7) Resolv-Thread generates DNS query message.
8) Resolv-Thread makes the IPv6 site-local solicited name multicast address corresponding to the name like Fig. 5.
9) Resolv-Thread sends the query message in multicast and starts query timer.
10) Resolv-Thread waits for the response until query timer expires.
11) When Resolv-Thread wakes up, it checks if the response arrived.
12) If Resolv-Thread woke up because of the expiration of query timer, it checks if the number of retransmission of query message is less than or equal to the allowed maximum number of retransmission, MAX_RETRANS [4]. If the number of retransmission of query message, #Retransmission, is greater than MAX_RETRANS, Resolv-Thread returns the error to the application and exits. Otherwise, it goes to Step 9.
13) If Resolv-Thread woke up because of the receipt of response message, it checks if the response is valid. If the response is invalid, Resolv-Thread discards the response and checks #Retransmission. If #Retransmission is greater than MAX_RETRANS, Resolver returns the error to the application and exits. Otherwise, it goes to Step 9.
14) If Resolv-Thread received multiple valid response messages from other ANS Responders, it sends the first response to every ANS Responder that sent a response except the ANS Responder that has sent a response first. This triggers the dynamic update request at each ANS Responder that receives this response message. After this, Resolv-Thread stores the response in ANS Cache, passes the response to the application and exits. If Resolv-Thread received only one valid response, it stores the response in ANS Cache, passes the response to the application and exits.

3.4 Name Service in ANS

Generation of Unique Domain Name. The mechanism of name generation makes a unique domain name with user-id, device-id (network device's address extended into EUI-64 identifier) and domain like Fig. 6 [8]. user-id is the user identifier selected by user and device-id is EUI-64 identifier derived from the network device's built-in 48-bit IEEE 802 address. domain indicates the kind of network where a node is positioned, which should include "EUI-64" sub-domain which indicates that the domain name is based on EUI-64. We define the domain for ad-hoc network as EUI-64.ADHOC. For example, when user-id is “PAUL-1”, device-id is “36-56-78-FF-FE-9A-BC-DE”, and domain is “EUI-64.ADHOC”, a unique domain name would be “PAUL-1.36-56-78-FF-FE-9A-BC-DE.EUI-64.ADHOC”. The advantage of the above mechanism guarantees that no name conflict happens without the verification procedure of the uniqueness of domain name although users in other nodes use the same user-id. It is possible only when all the nodes in MANET should use the above name generation mechanism, but even though a node configures its domain name by manual configuration or other methods, mobile nodes can detect the name conflict through the dynamic update request. user-id and domain are registered in options statement of the configuration file of ANS Responder (ans.conf) like Fig. 7.

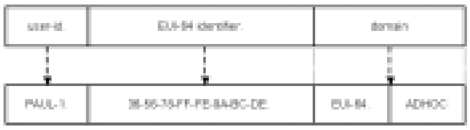

Fig. 6. Format of Unique Domain Name based on EUI-64 Identifier

```
/* Configuration File of ANS Responder (ans.conf) */
  options {
    user-id "PAUL-1";
    domain "ADHOC.";
  };
```

Fig. 7. Configuration File of ANS Responder

Generation of Zone File. ANS Responder generates zone file that contains the domain name generated above and the site-local scoped IPv6 address of the network device corresponding to the name like Fig. 8. The IPv6 address of the name, “PAUL-1.36-56-78-FF-FE-9A-BC-DE.EUI-64.ADHOC”, is “FEC0::3656:78FF:FE9A:BCDE”.

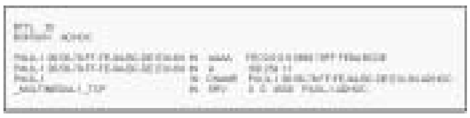

Fig. 8. Zone File of ANS Responder

Service Discovery through ANS. ANS can provide service discovery in use of DNS SRV resource record which has been defined for specifying the location of services [5]. The service discovery allows a client to get the information which is necessary to access a server. The specification for a service consists of service name, protocol of transport layer, domain which the service is located in, weight, priority, port number and domain name for IPv6 address. Fig. 8 shows an example of SRV resource record of the service named "_MULTIMEDIA-1._TCP".

Security Consideration. We assume the truested nodes generate their own domain name with the name generation mechanism of this paper. Therefore, there can not be the name conflict within the group of the trusted nodes.

In order to provide securer name service in ANS, we can use IPsec ESP with a null-transform to authenticate ANS response, which can be easily accomplished through the configuration of a group pre-shared key for the trusted nodes [4].

4 Experiment in MANET Testbed

4.1 Network Topology

Fig. 9 shows the topology of mobile network for the experiment of ANS name resolution. Three mobile nodes are connected through ad-hoc routing protocols. We implemented IPv6 AODV for unicast routing and MAODV for multicast routing [9–12]. Table. 1 describes the name information of mobile nodes in the testbed.

4.2 Experiment

The DNS query issued by mobile node MN-C could be resolved as follows; "ans-sender" is a simple program that uses ANS name resolution functions in order to resolve domain name through ANS Resolver. ans-sender on MN-C could

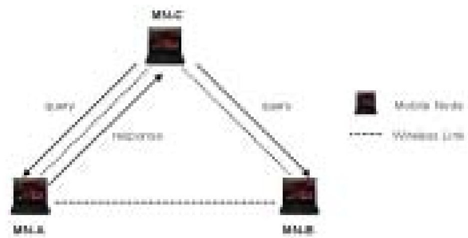

Fig. 9. Topology of MANET Testbed

Table 1. Name Information of Mobile Nodes

Node	User ID	EUI-64 ID	Domain	IPv6 Address
MN-A	PAUL-1	36-56-78-FF-FE-9A-BC-DE	EUI-64.ADHOC	FEC0::3656:78FF:FE9A:BCDE
MN-B	PAUL-2	02-01-02-FF-FE-FD-40-05	EUI-64.ADHOC	FEC0::0201:02FF:FEFD:4005
MN-C	PAUL-3	02-02-2D-FF-FE-1B-E8-51	EUI-64.ADHOC	FEC0::0202:2DFF:FE1B:E851

resolve the domain name of MN-A, “PAUL-1.36-56-78-FF-FE-9A-BC-DE.EUI-64.ADHOC”, into the IPv6 address, “FEC0::3656:78FF:FE9A:BCDE”.

5 Conclusion

In this paper, we proposed an architecture of name service system called as ANS (Ad-hoc Name Service System for IPv6 MANET) which can provide mobile nodes in IPv6 MANET with the name-to-address resolution and autoconfiguration technology for easy configuration related to name service including the generation of unique domain name of mobile node and generation of zone file for name service. We also explained service discovery performed through the name service system of this paper and DNS service resource record (SRV). With ANS, users can run name service easily in the unmanaged or unadministrated networks where there are no network manager and dedicated name server such as home network and small office home office (SOHO) as well as ad-hoc network.

In future work, we will append security service to ANS to prevent the security attacks.

References

1. IETF Manet working group, http://www.ietf.org/html.charters/manet-charter.html
2. S. Thompson and T. Narten, “IPv6 Stateless Address Autoconfiguration”, RFC2462 , December 1998.
3. Jaehoon Jeong and Jungsoo Park, “Autoconfiguration Technologies for IPv6 Multicast Service in Mobile Ad-hoc Networks”, ICON2002, August 2002.
4. Levon Esibov and Dave Thaler, “Linklocal Multicast Name Resolution (LLMNR)”, (work in progress) draft-ietf-dnsext-mdns-13, November 2002.
5. A. Gulbrandsen, P. Vixie and L. Esibov, “A DNS RR for specifying the location of services (DNS SRV)”, RFC2782, February 2000.
6. IETF Zeroconf working group, http://www.ietf.org/html.charters/zeroconf-charter.html
7. P. Vixie, S. Thomson, Y. Rekhter and J. Bound, “Dynamic Updates in the Domain Name System (DNS UPDATE)”, RFC2136, April 1997.
8. Jae-Hoon Jeong, Jung-Soo Park and Hyoung-Jun Kim, “Unique DNS Name Generation”, draft-jeong-name-generation-01, February 2003.
9. Charles E. Perkins, Elizabeth M. Belding-Royer and Samir R. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing”, (work in progress) draft-ietf-manet-aodv-13, February 2003.
10. Elizabeth M. Royer and Charles E. Perkins, “Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing”, draft-ietf-manet-maodv-00, July 2000.
11. C. Perkins, E. Royer and S. Das, “Ad Hoc On Demand Distance Vector (AODV) Routing for IP version 6”, draft-perkins-manet-aodv6-01, November 2001.
12. Implementation of IPv6 AODV and MAODV, http://www.adhoc.6ants.net