Service Discovery based on Multicast DNS in IPv6 Mobile Ad-hoc Networks

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

Abstract— **This paper presents the mechanism for service discovery in IPv6 mobile ad-hoc networks, which uses the multicast DNS. When a user knows a service name and its transport protocol, he or she can find the IPv6 address and the port number of the service through this mechanism. The user can find the service information of multicast service as well as that of unicast service. The paper also suggests the architecture of name service system not only for service discovery, but also for DNS name resolution.**

I. Introduction

Mobile Ad-hoc Networks (MANET) are the networks where mobile nodes can communicate with one another without communication infrastructure such as base station or access point. When mobile nodes communicate in the environments such as battle field, airplane and boat where are separated from the Internet, they need to construct a temporary and infrastructureless network [1]. Recently, according as the necessity of MANET increases, the development of the ad-hoc routing protocols for multi-hop MANET has been being leaded very strongly by IETF MANET working group [2]. Also, the multicast routing protocols , multicast service such as video conferencing, DNS service, and service discovery in MANET have been being developed. With this trend, if IPv6 that has lots of good functions such as stateless address autoconfiguration for address configuration is adopted well in MANET, users in MANET will be able to communicate more easily through the zeroconfiguration that provides easy configuration [3].

This paper suggests the mechanism of service discovery in IPv6 MANET. The mechanism is performed through multicast DNS and service resource record (SRV) [4], [5]. The mechanism allows ad-hoc users to discover the service information that is necessary to connect or join to the service when the service name, transport protocol (e.g., TCP or UDP) and domain where the service is placed are given. Also, this paper suggests the architecture of name service system for DNS service as well as service discovery in IPv6 MANET, which is called Ad-hoc Name Service System for IPv6 MANET (ANS).

This paper is organized as follows; Sect. 2 presents related work. In Sect. 3, we explain the architecture and operation of the suggested name service system ,"ANS", and name service in ANS. In Sec. 4, the service discovery based on ANS is described. Sect. 5 show the result of the service discovery and name resolution through ANS in MANET testbed. Finally, in Sect. 6, we conclude this paper and present future work.

II. Related Work

A. 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 environments, such as small office home office (SOHO) networks, airplane networks and home networks, which is called zero-configuration or auto-configuration [3], [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.

B. 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 [7]. LLMNR comprises Responder and Sender. 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. Unless Sender receives any response during the limited amount of time (LLMNR_TIMEOUT, 1 [sec]), it retransmits LLMNR query by 3 times in order to assure itself that the query has been received by a node capable of responding to the query.

III. 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 [8], [9]. In this section, we described

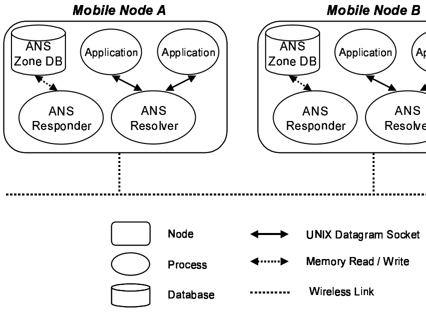

Fig. 1. ANS System for Name Service in IPv6 MANET

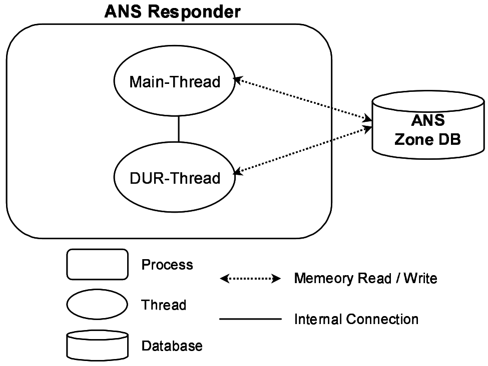

Fig. 2. Architecture of ANS Responder

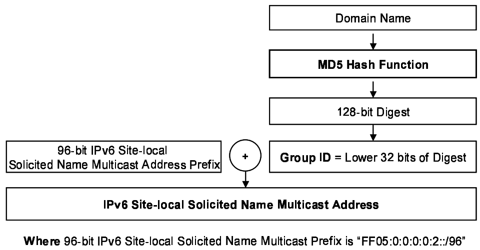

Fig. 3. Generation of IPv6 Site-local Solicited Name Multicast Address

the architecture and operation of ANS System, and the domain name service in ANS.

A. 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 show 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 the functions for name resolution through which they can communicate with ANS Resolver. ANS Resolver sends the DNS query for a name in the multicast address with which ANS Responder in each mobile node has joined for the name. 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.

B. 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. 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 the solicited name multicast addresss which is made from Name which is domain name through Fig. 3. 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 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. 4 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 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 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

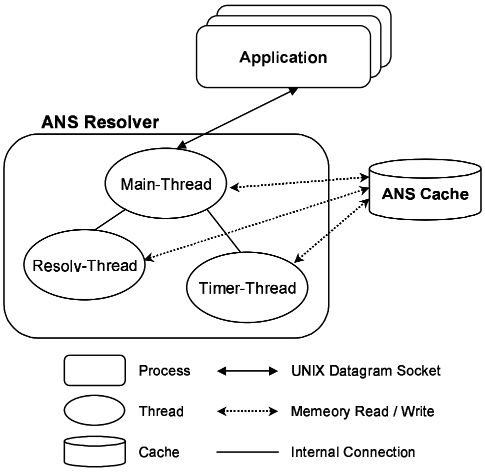

Fig. 4. Architecture of ANS Resolver

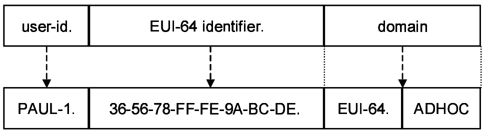

Fig. 5. Format of Unique Domain Name based on EUI-64 Identifier

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 unused for name resolution. After the work, Timer-Thread restarts ANS Cache timer.

C. Name Service in ANS

1) 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. 5 [10]. 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 placed, 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 the resource record related to the domain name although users in other nodes use the same user-id. But it is valid only when all the nodes in MANET should use the above name generation mechanism. Nevertheless, 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 [4]. user-id and domain are registered in options statement of

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

Fig. 6. Configuration File of ANS Responder (ans.conf)

```
$TTL    20
$ORIGIN   ADHOC.

PAUL-1.36-56-78-FF-FE-9A-BC-DE.EUI-64  IN   AAAA    FEC0:0:0:0:3656:78FF:FE9A:BCDE
PAUL-1                   IN    CNAME   PAUL-1.36-56-78-FF-FE-9A-BC-DE.EUI-64.ADHOC.

;; DNS SRV Resource Records
; Unicast Service : SERVICE-1
_SERVICE-1._TCP                    IN  SRV  0  1  3000  PAUL-1.ADHOC.
_SERVICE-1._UDP                    IN  SRV  0  1  3000  PAUL-1.ADHOC.

; Multicast Service : SERVICE-2
_SERVICE-2._UDP                    IN  SRV  0  1  4000  @.3.5.
```

Fig. 7. Zone File of ANS Responder

TABLE I
DNS SRV RESOURCE RECORD

Field	Description
service	symbolic name of the service preceded by an underscore (e.g., _SERVICE-1)
protocol	symbolic name of the protocol preceded by an underscore (e.g., _TCP or _UDP)
domain	domain the resource record refers to
ttl	time to live, the amount of time during which the resource record is allowed to be cached
class	class code of the resource record (e.g., IN)
type	type code of the resource record (i.e., SRV)
priority	priority of this target host
weight	relative weight for entries with the same priority
port	port number of the specified protocol
target	domain name that can be resolved into IPv6 address

the configuration file of ANS Responder (ans.conf) like Fig. 6.

2) Generation of Zone File: ANS Responder generates a zone file that contains the domain name generated above and site-local scoped IPv6 address of the network device corresponding to the name like Fig. 7. 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".

IV. SERVICE DISCOVERY

We assume that only mobile node running multicast or unicast service can register a DNS SRV resource record in its ANS zone file like Fig. 7 [5]. The SRV resource record is comprised of 10 fields as follows; (a) service, (b) protocol, (c) domain, (d) ttl, (e) class, (f) type (SRV), (g) priority, (h) weight, (i) port, and (j) target. Table. I explains the fields of the above fields.

A. Registration of SRV Resource Record

The service information of an application in mobile node is registered in ANS zone file by the manual or automatic configuration. In this point, we adopted the manual configuration for simplicity.

The information of unicast service is registered in a SRV resource record in ANS zone file like Fig. 7. In this fig-

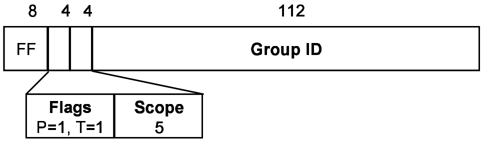

Fig. 8. Format of Multicast Address corresponding to Service Name

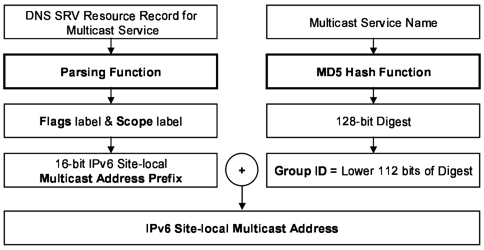

Fig. 9. Generation of Multicast Address for Service Name

ure, two unicast services are registered; (a) "_SERVICE-1._TCP.ADHOC." and (b) "_SERVICE-1._UDP.ADHOC.". Their resource data are the same. The class is "IN", the type is "SRV", the priority is "0", the weight is "1", the port is "3000", and the target is "PAUL-1.ADHOC.". Because the target is the alias of the canonical name "PAUL-1.36-56-78-FF-FE-9A-BC-DE.EUI-64.ADHOC.", it can be resolved into IPv6 address "FEC0:0:0:0:3656:78FF:FE9A:BCDE". Unlike RFC2782 about DNS SRV, we allow alias for the target field of SRV resource record [5]. Because the automatically generated domain name is too long to memorize. Of course, the uniqueness of the alias is verified through the dynamic update request [4], [7].

It is similar to the case of the unicast service to register the multicast service in ANS zone file. To represent the multicast service, we should support a new particular value for "target" field in SRV resource record like the resource record "_SERVICE-2._UDP.ADHOC." in Fig. 7. The format of the value is "@.flags.scope.". The special character @ indicates that this SRV resource record is for a multicast service. The label "flags" is a hex number consisting of two reserved bits, P-bit (Prefix bit) and T-bit (Temporary bit) of IPv6 multicast address. The label "scope" is a hex number that represents the scope field of IPv6 multicast address. Regarding a multicast service, the format of the multicast address corresponding to its name is like Fig. 8 [9]. The important fields of the multicast address are set as follows; The Flags and Scope fields are set to "flags" label and "scope" label of the SRV response. The Group ID field is set to the lower 112 bits of MD5 digest of a multicast service name like Fig. 9.

B. Procedure of Service Discovery

Fig. 10 shows the procedure of service discovery. Client sends the DNS SRV query to get the information of a service named as service-1 via site-local multicast through ANS Resolver. The mobile node that can serve the queried service responds to the client query and delivers the data of SRV resource record to the client via site-local unicast. When the client receives the response of SRV query, it checks whether the service is unicast or multicast. If the service is unicast, the client tries to connect the server with the server's IPv6 address, transport protocol and port number. If the service is multicast, the client makes the multicast address related to the multicast service through the procedure of Fig. 9 and joins the multicast group with the multicast address and UDP port number of the service.

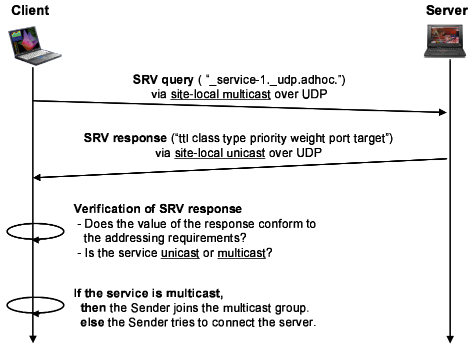

Fig. 10. Procedure of Service Discovery

TABLE II

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

C. 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 including service discovery 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 secret key for the trusted nodes and the use of the keyed hash method such as HMAC [7], [12].

V. EXPERIMENT IN MANET TESTBED

In this section, we show the service discovery as well as the name resolution through ANS System in IPv6 MANET.

A. Network Topology

Fig. 11 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 [13], [14], [15], [16]. Table. II describes the name information of mobile nodes in the testbed.

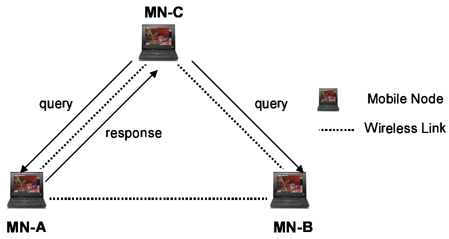

Fig. 11. Network Topology of MANET Testbed

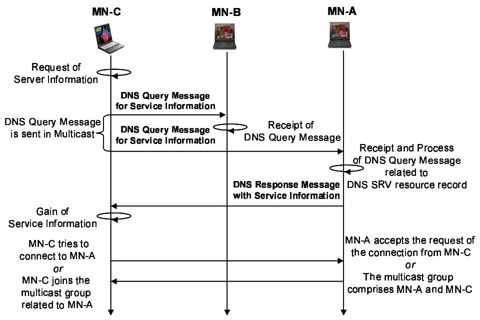

Fig. 12. Scenario of Service Discovery

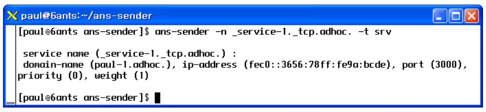

Fig. 13. Discovery of Unicast Service through ANS

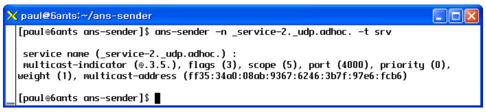

Fig. 14. Discovery of Multicast Service through ANS

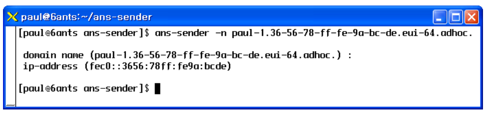

Fig. 15. Name-to-Address Resolution through ANS

B. Experiment

ANS allows mobile nodes to find out the information that is needed to use a service. Fig. 12 shows the scenario of the service discovery in MANET. "ans-sender" is a simple program that uses ANS name resolution functions in order to perform the service discovery and resolve domain name into IP address and through ANS. Through ANS, mobile node MN-C can find the information of the unicast service "_SERVICE-1._TCP.ADHOC." and that of the multicast service "_SERVICE-2._UDP.ADHOC." like Fig. 13 and Fig. 14.

Of course, ANS allows mobile nodes to resolve a domain name into IPv6 address. Like Fig. 15, mobile node MN-C can resolve the domain name of mobile node MN-A ("PAUL-1.36-56-78-FF-FE-9A-BC-DE.EUI-64.ADHOC.") into IPv6 address ("FEC0::3656:78FF:FE9A:BCDE").

VI. Conclusion

In this paper, we proposed an architecture of name service system called as ANS (Ad-hoc Name Service System for IPv6 MANET) for DNS service in IPv6 MANET. ANS allows mobile nodes perform the service discovery as well as the name-to-address resolution. The service discovery comprises the discovery of unicast service and that of multicast service, which are performed by the ANS and enhanced DNS service resource record (SRV).

Also, ANS provides mobile users in MANET with the 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. ANS can be adopted for the DNS service and service discovery not only in the ad-hoc network, but also in the unmanaged networks where there are no network manager and dedicated name server, such as home network and small office home office (SOHO).

In future work, we will enhance security function of ANS to provide securer name service agaist the various security attacks.

References

[1] Elizabeth M. Royer and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, April 1999.

[2] IETF Manet working group, http://www.ietf.org/html.charters/manet-charter.html

[3] A. Williams, "Requirements for Automatic Configuration of IP Hosts", (work in progress) draft-ietf-zeroconf-reqts-12, September 2002.

[4] Jaehoon Jeong, Jungsoo Park, Hyoungjun Kim and Kishik Park, "Name Service in IPv6 Mobile Ad-hoc Network", *17th International Conference on Information Networking*, February 2003.

[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] Levon Esibov and Dave Thaler, "Linklocal Multicast Name Resolution (LLMNR)", (work in progress) draft-ietf-dnsext-mdns-13, November 2002.

[8] Charles E. Perkins et al., "IP Address Autoconfiguration for Ad Hoc Networks", draft-perkins-manet-autoconf-01, November 2001.

[9] Jaehoon Jeong and Jungsoo Park, "Autoconfiguration Technologies for IPv6 Multicast Service in Mobile Ad-hoc Networks", *10th IEEE International Conference on Networks*, August 2002.

[10] Jae-Hoon Jeong, Jung-Soo Park and Hyoung-Jun Kim, "Generation of Unique Domain Name based on Network Device Identifier", draft-jeong-name-generation-00, October 2002.

[11] P. Vixie, S. Thomson, Y. Rekhter and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC2136, April 1997.

[12] H. Krawczyk, M. Bellare and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC2104, February 1997.

[13] 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-11, June 2002.

[14] 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.

[15] Elizabeth M. Royer and Charles E. Perkins, "Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing", draft-ietf-manet-maodv-00, July 2000.

[16] Implementation of IPv6 AODV and MAODV, http://www.adhoc.6ants.net/