Autoconfiguration Technology for IPv6-based Mobile Ad-hoc Network

Jaehoon Jeong, Jungsoo Park

Protocol Engineering Center, ETRI, 161 Gajong-Dong, Yusong-Gu, Daejon 305-350, Korea

{paul, pjs}@etri.re.kr

http://pec.etri.re.kr/

ABSTRACT

This paper presents the autoconfiguration technology that can be used for lots of network services including multimedia services in IPv6-based mobile ad-hoc network. A mobile node gets a unique unicast address through IPv6 unicast address autoconfiguration, some applications on the mobile node that need multicast address allocation can get a unique multicast address through IPv6 multicast address autoconfiguration, the applications can resolve a domain name through multicast name resolution and the client can find the information which is needed to access a server through service discovery. Thus, the user of mobile node can use network services easily with this autoconfiguraiton technology.

Keywords: Autoconfiguration, Ad-hoc Network, IPv6, Multicast, Address, DNS, Service Discovery

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 and access point [1]. In order that mobile nodes communicate in the environments such as battle fields, airplanes and boats where are separated from the Internet, we need to construct a temporary and infrastructureless network, namely ad-hoc network. Recently, according that the necessity of MANET increases, the development of ad-hoc routing protocols for multi-hop MANET is activated strongly. Multicast routing protocols for multicast service such as video conferencing are also being developed. With this trend, if IPv6 that has lots of good functions such as autoconfiguration for address configuration is adopted well in MANET, users in MANET will be able to communicate more easily. So we have become to want to maximize the autoconfiguration of IPv6 which is known as what is called zeroconfiguration that provides easy configurations, which have been developed by IETF Zeroconf working group [2].

This paper suggests four mechanisms related to the autoconfiguration technology for IPv6-based MANET like Figure 1 [3];

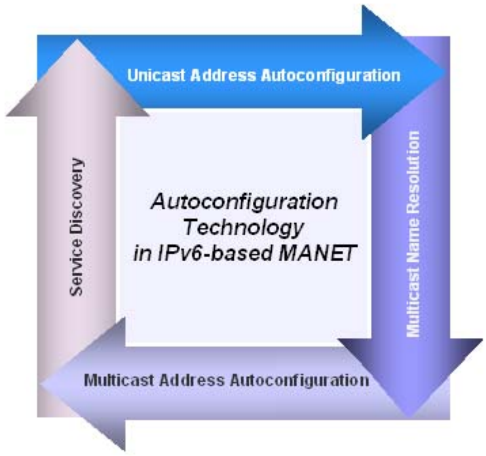

Figure 1. Autoconfiguration technology in IPv6-based MANET

The 1st is unicast address autoconfiguration through which a unique unicast address is configured in each network interface card (NIC) of mobile node (MN).

The 2nd is multicast address autoconfiguration through which a unique multicast address is allocated to application that needs a new multicast address.

The 3rd is multicast name resolution what is called mDNS (Multicast DNS) through which the resolution between domain name and IPv6 address is performed.

The last is service discovery through which a client gets the information which is necessary to access a server.

This paper is organized as follows; In Section 2, related work is presented. In Section 3, we describe four mechanisms related to the autoconfiguration technology for IPv6-based MANET. In Section 4, we suggest the method of applying these mechanisms to MANET with two service scenarios. In Section 5, we provide the guide to API implementation of the autoconfiguration technology. Finally, in Section 6, we conclude this paper and present future work.

2. RELATED WORK

In this section, we summarize the ad-hoc routing protocols that are being researched and developed for ad-hoc network, the autoconfiguration in IPv6 and the link-local scoped domain name service using multicast name resolution.

2.1 Ad-hoc Routing Protocols

As shown in Table 1, ad-hoc unicast routing protocols may be generally categorized as two kinds; (a) Table-driven routing protocols and (b) Demand-driven routing protocols [1][4][5]. In Table-driven, MNs exchanges routing information in advance and are ready to communicate. In Demand-driven, when MNs start to communicate, route discovery is initiated by source.

Table 1. Classification of ad-hoc routing protocols

Table-driven Routing Protocols	Demand-driven Routing Protocols
DSDV, CGSR, WRP	AODV, DSR, LMR, TORA, ABR, SSR

For ad-hoc multicast routing protocols, several multicast routing protocols such as 'MAODV' based on AODV and 'Simple protocol for multicast' based on DSR are being developed [6][7].

2.2 Autoconfiguration in IPv6

IPv6 unicast address can be configured in NIC through Neighbor Discovery Protocol (NDP) and Stateless Address Autoconfiguration [8][9].

In MANET, MN is not only router but also host. The current NDP that for unicast address configuration, sends RA (Router Advertisement) messages periodically in link-local all node multicast address is not suitable for MANET that has dynamic topology. The current Duplicate Address Detection (DAD) that checks whether tentative unicast address is being used by other MNs can also not work well in MANET without some modifications. Therefore, NDP is necessary to extend [10][11].

2.3 Link-Local Multicast Name Resolution

Link-Local Multicast Name Resolution (LLMNR) is the DNS mechanism which provides the domain name service of link-local scope with IP multicast in both IPv4 and IPv6 [12]. This mechanism is useful especially in the ad-hoc network where there is no fixed DNS server and users can access other computers by their DNS names.

3. AUTOCONFIGURATION TECHNOLOGY FOR IPV6-BASED MANET

In this section, first of all, the network configuration which is used to explain the mechanisms of this paper is described. Next, the protocol stack of MN and four mechanisms of the autoconfiguration technology are explained.

3.1 Network Configuration

In MANET shown in Figure 2, the following five MNs (i.e. A, B, C, D and E) are connected through ad-hoc routing protocol. A is adjacent to B, B is adjacent to A and C and so on. When A communicates with E, the intermediate MNs (i.e. B, C and D) should play the role of router. This topology will be changed according to the movement of MNs.

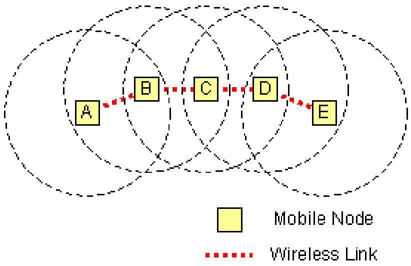

Figure 2. Network configuration

3.2 Protocol Stack of Mobile Node

The structure of the protocol stack of MN is shown in Figure 3.

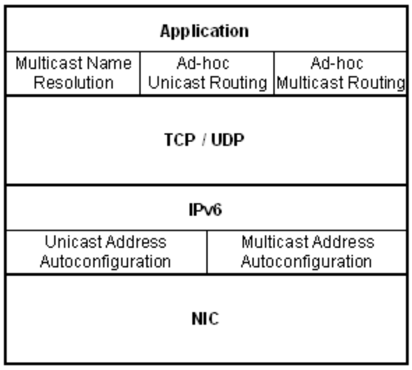

Figure 3. Protocol stack of mobile node

In Application layer, the ad-hoc unicast / multicast routing protocols and the multicast name resolution are executed. In IPv6 layer, unicast address autoconfiguration for unicast address configuration in NIC and multicast address autoconfiguration for multicast address allocation are executed. In this environment, the MN that executes the multicast application which needs multicast address allocation can configure a unicast address in its NIC through unicast address autoconfiguration and allocate a multicast address to the application through multicast address autoconfiguration by itself.

3.3 Unicast Address Autoconfiguration

Because we don't consider the global connectivity to the Internet in this paper, we assume that MANET is isolated network from the Internet and the scope of addresses used in MANET is site-local [10][11]. MANET exclusive prefixes are defined like Table 2.

Unlike [10], we use two prefixes of length 64; MANET_INIT_PREFIX is for temporary unicast address in DAD and MANET_PREFIX is for actual unicast address. The subnet ID of MANET_INIT_PREFIX is fffe and that of MANET_PREFIX is ffff.

Table 2. MANET prefixes

Prefix Name	Prefix Value
MANET_INIT_PREFIX	fec0:0:0:fffe::/64
MANET_PREFIX	fec0:0:0:ffff::/64

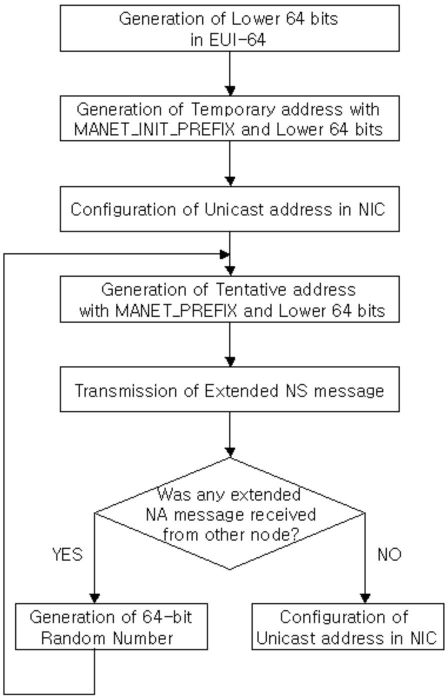

Figure 4. Procedure of site-local unicast address autoconfiguration

MANET_INIT_PREFIX is the prefix for temporary source address that is used temporarily in DAD in order that MN configure a unique site-local unicast address of prefix MANET_PREFIX in NIC. The lower 64 bits of the temporary source address are configured in EUI-64 [13]. With this temporary source address, MN starts the procedure shown in Figure 4 so that a unique site-local unicast address be reconfigured in NIC. In the last step of Figure 4, when the verified tentative unicast address is reconfigured as actual unicast address in NIC of MN, the temporary source address is not used any more as the source address.

The important point in Figure 4 is that when a unicast address is composed of MANET_PREFIX and the lower 64 bits, it is a tentative address of which the uniqueness has not been verified in MANET yet. MN must check the uniqueness with DAD [8][9]. Because the current DAD works only in link-local scope, NDP should be extended so that it work in site-local scope. Therefore, this paper uses the extended NS (Neighbor Solicitation) message and NA (Neighbor Advertisement) message which are suggested in [10][11].

3.4 Multicast Address Autoconfiguration

The format of site-local unicast address [14] and that of site-local multicast address [15] are shown in Figure 5. So that we indicate the multicast address of Figure 5 (b) is based on network prefix and is used temporarily, P-bit (Prefix bit) and T-bit (Temporary bit) that are Flags are set to 1. Subnet ID of Figure 5 (b) is set to that of the site-local unicast address of Figure 5 (a), ffff (the subnet ID of MANET_PREFIX). Interface ID of Figure 5 (b) is set to that of Figure 5 (a), which is the lower 64 bits of the unicast address configured in Figure 4. Group ID is set to a 32-bit random number generated by MN.

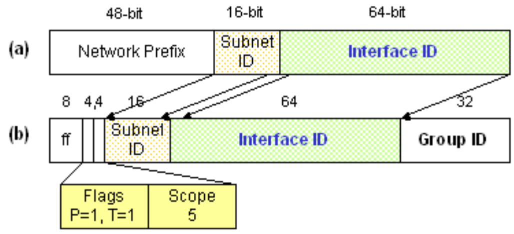

Figure 5. (a) Format of site-local unicast address & (b) Format of site-local multicast address

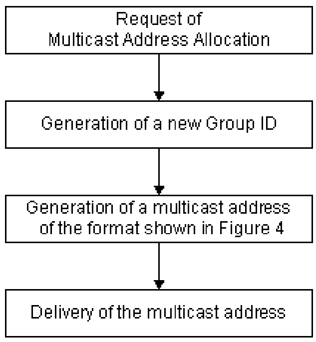

Figure 6. Procedure of site-local multicast address autoconfiguration

If a multicast application requests the allocation of a multicast address to the kernel, it is allocated a multicast address from the kernel in the procedure of Figure 6. In this way, as a unique site-local multicast address can be constructed on the MN's own without multicast address allocation server [16], this mechanism for multicast allocation is suitable for MANET.

3.5 Multicast Name Resolution

Multicast Name Resolution (MNR) has been extended from LLMNR in order that MNR perform the name resolution between domain name and IPv6 address with IPv6 multicast in the site-local scoped MANET [12]. Figure 7 shows the procedure of the resolution from domain name to IPv6 address. Sender is the resolver that sends MNR query in site-local multicast and Responder is the name server that sends the MNR 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 MNR cache and uses the result of it. Otherwise, Sender ignores the response and continues to wait for other responses. Unless Sender receives any response during a limited amount of time (LLMNR_TIMEOUT), it retransmits MNR query by 3 times in order to assure itself that the query has been received by an MN capable of responding to the query.

For MNR query, a site-local multicast address is generated by the method of Figure 8 per resource record name. The prefix of the multicast address is FF05:0:0:0:0:2::/96 and the lower 32 bits of it is that of MD5 hashing value for the resource record name such as domain name. In this way, Responder of MNR must join each site-local multicast address made out of each resource record name in MNR zone file.

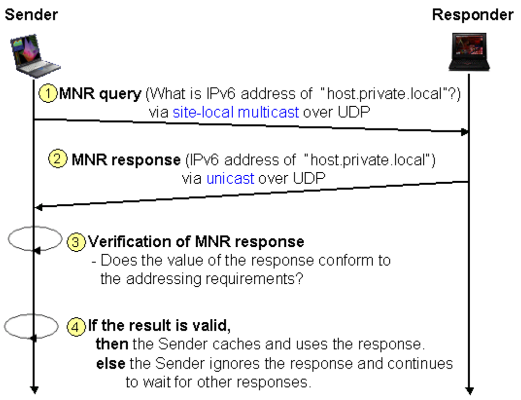

Figure 7. Procedure of the resolution from domain name to IPv6 address in MNR

For simplicity, we store AAAA resource records and PTR resource records together in an MNR zone file. The zone file for Responder of Figure 7 contains two resource records (AAAA and PTR) of its own site-local unicast address and domain name like Figure 9. It also contains two resource records (AAAA and PTR) of localhost and loopback address. We decided the domain for MANET as private.local. and TTL for caching resource records as 3600 seconds.

The utility which makes this zone file for MNR automatically instead of user is necessary for the auto-configuration of MNR. It can make the zone file with the system information of MN such as the domain name and the site-local address of NIC when MNR Responder Daemon is initiated.

FF05:0:0:0:0:2::/96 + Lower 32 bits of **MD5_Hash** (Resource Record Name)

Figure 8. Generation of site-local multicast address

```
$TTL 3600

;; Name to Address Lookups
localhost.private.local. IN AAAA ::1
; Localhost with Loopback Address
host.private.local. IN AAAA fec0:0:0:ffff::202:2dff:fe1b:e851
; MN's Domain Name with Site-Local Address

;; Address to Name Lookups using Nibble Format
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.int. IN PTR localhost.
; Loopback Address with Localhost
1.5.8.e.b.1.e.f.f.f.d.2.2.0.2.0.f.f.f.f.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int. IN PTR host.private.local.
; MN's Site-Local Address with Domain Name
```

Figure 9. Zone file of Responder

3.6 Service Discovery

The service discovery let a client get the information which is necessary to access a server. The specification for a service consists of service name, domain which the service is located in, IPv6 address and port number. We can provide service discovery for MNs through MNR and DNS SRV resource record which is used for specifying the location of services [17]. We register a DNS SRV resource record per service in MNR zone file like Figure 10.

```
$TTL 3600

;; Name to Address Lookups
localhost.private.local. IN AAAA ::1
; Localhost with Loopback Address
host.private.local. IN AAAA fec0:0:0:ffff::202:2dff:fe1b:e851
; MN's Domain Name with Site-Local Address

;; Address to Name Lookups using Nibble Format
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.int. IN PTR localhost.
; Loopback Address with Localhost
1.5.8.e.b.1.e.f.f.f.d.2.2.0.2.0.f.f.f.f.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int. IN PTR host.private.local.
; MN's Site-Local Address with Domain Name

;; DNS SRV Resource Records
_multimedia1._tcp.private.local. IN SRV 0 1 3000 host.private.local.
_multimedia2._udp.private.local. IN SRV 0 1 3001 host.private.local.
```

Figure 10. Zone file containing DNS SRV resource records for service discovery

In Figure 10, two services are registered in zone file; The information of the 1st is as follows; the service name is multimedia1, the domain where multimedia1 service is located is private.local., the protocol is TCP, the priority is 0, the weight is 1, the port number is 3000 and the domain name of the server host is host.private.local..

Thus, the client of multimedia1 can get the information of server through MNR SRV query for multimedia1 and it can connect to a multimedia1 server.

4. Service of Multimedia Applications

After MN configures site-local unicast address in its NIC through unicast address autoconfiguration of section 3.3, MNs can communicate with one another over ad-hoc routing protocols in the multi-hop ad-hoc network. When a multicast application of MN needs the allocation of a unique multicast address, it can be allocated a multicast address through multicast address autoconfiguration of section 3.4. When a client wants to get the service of a server, it can find the information for accessing the server through service discovery of section 3.6, resolve the domain name of the server through MNR of section 3.5 and finally connect to the server.

In this section, we show the usage of these four mechanisms presented in this paper through the service scenarios of video conferencing and video on demand (VoD).

4.1 Service Scenario of Video Conferencing

When the configuration of unicast address in each MN has been completed and ad-hoc routing protocols for unicasting and multicasting start to work, all the MNs in MANET that is shown of Figure 2 are ready to communicate. Figure 11 describes the service scenario of video conferencing in MANET. MN A creates a new conference session and MN E joins the session. Table 3 explains the service scenario of Figure 11 step by step.

Table 3. Steps of the service scenario of video conferencing

Step	Related nodes	Description of the step
1	A, B, C, D, E	Each MN autoconfigures its site-local unicast address through the extended DAD.
2	A	When a videoconferencing application of MN A creates a new session named as S, a site-local multicast address for session S is autoconfigured.
3	A, B, C, D, E	A multicasts the information for session S and the rest of the nodes receive it through multicast routing.
4	A	A joins session S.
5	E	When E joins session S, the multicast route between E and A is set up.
6	A, B, C, D, E	A sends videoconferencing data to E in multicast. B, C and D relay the data packet of A to E through multicasting.
7	A, B, C, D, E	E sends videoconferencing data to A in multicast. D, C and B relay the data packet of E to A through multicasting.

4.2 Service Scenario of Video on Demand

The MNs in MANET that is shown of Figure 2 are connected through ad-hoc unicast and multicast routing protocols.

MN E runs a server of which service name is multimedia1. The SRV resource record for multimedia1 is the same as that of Figure 10. A VoD client on MN A wants to connect to a VoD server named as multimedia1. The client can find the server and then be served by the server like Figure 12. First of all, the client on MN A finds the information for accessing the server through service discovery which is based on MNR and DNS SRV resource record and then it resolves the domain name of the server contained in the SRV resource record into IPv6 address.

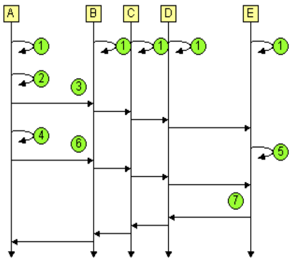

Figure 11. Service scenario of video conferencing

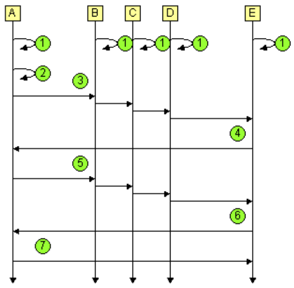

Figure 12. Service scenario of video on demand

Finally, it tries to connect to the server with IPv6 address and port number of the server. Table 4 explains the service scenario of Figure 12 step by step.

We can reduce four steps (from step 3 to step 6) into two steps if we adopt DNS implementation optimization in MNR, which returns the IPv6 address of the server together through the additional information field of the response message for the SRV query [17]. Because the client becomes to know the IPv6 address of the server through the additional information field, it is needless for the client to resolve the domain name of the server.

Table 4. Steps of the service scenario of video on demand

Step	Related nodes	Description of the step
1	A, B, C, D, E	Each MN autoconfigures its site-local unicast address through the extended DAD and makes its MNR zone file.
2	A	A VoD client is executed on MN A.
3	A, B, C, D, E	So that the client find the location of a VoD server named as multimedia1, it sends a SRV query for multimedia1 in multicast through MNR.
4	A, E	The MNR responder on MN E running a multimedia1 server responds to the query and sends the response message to the client in unicast.
5	A, B, C, D, E	So that the client resolve the domain name of the multimedia1 server into IPv6 address, it sends an AAAA query for the domain name in multicast through MNR.
6	A, E	The MNR responder on MN E responds to the query and sends the response message to the client in unicast.
7	A, E	The client on MN A tries to make a TCP connection with the server on MN E with IPv6 address and port number of the server.

5. Guide to API Implementation of the Autoconfiguration Technology

When a multicast application such as SDR (Session Directory) that is one of UCL Mbone conferencing applications needs a multicast address for a new session [18], it can be allocated a site-local multicast address with API allocmcastaddr() shown in Figure 13.

```
int allocmcastaddr (int socket, struct in6_addr* mcastaddr);

                                   Returns: 0 if OK, -1 on error
```

Figure 13. API for multicast address allocation

The parameter socket of which type is int is Socket Descriptor. The parameter mcastaddr of which type is struct in6_addr* is the buffer used for getting a site-local multicast address.

When VIC (Videoconferencing) on an MN tries to connect to VIC on other MNs with the domain name of the MN, it can resolve the domain name with API mnr_res_query() shown in Figure 14.

```
int mnr_res_query (const char* name, int class, int type,
                   u_char* answer, int anslen);

        Returns: the size of the response if OK, -1 on error
```

Figure 14. API for MNR query

The parameter name of which type is const char* is the resource record name such as domain name. The parameter class and type of which type are int are the class and type of query. The parameter answer of which type is u_char* is the buffer used for getting the response of the query. The parameter anslen of which type is int is the size of answer buffer.

6. CONCLUSION

Mobile Ad-hoc Network (MANET) that is temporarily constructed network and has dynamic topology supports the communication among mobile nodes in the infrastructureless environment. Until now the research of MANET has been focused on the development of ad-hoc routing protocols.

This paper presented four mechanisms for IPv6-based autoconfiguration technology that help users to communicate with one another easily in MANET, which is assumed as site-local network in this paper. The first is unicast address autoconfiguration which configures a unique IPv6 site-local unicast address in mobile node. The second is multicast address autoconfiguration which allocates a unique IPv6 site-local multicast address to a multicast application without a multicast address allocation server. The third is multicast name resolution which resolves domain name with multicast. The last is service discovery which provides a client with the information of a server which is needed for the client to access the server. If these mechanisms are used in MANET, users can not only communicate easily with one another in MANET but also use easily some kinds of multimedia services that need multicast address allocation and service discovery.

In near future, after we implement IPv6-based ad-hoc unicast/multicast routing protocols (IPv6 AODV and IPv6 MAODV) and the autoconfiguration technology presented in this paper, we will construct the integrated mobile ad-hoc environment that supports the multimedia services such as video conferencing and real-time collaborative applications.

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 Zeroconf working group, http://www.ietf.org/html.charters/zeroconf-charter.html

[3] M. Hattig, "Zeroconf IP Host Requirements", (work in progress) draft-ietf-zeroconf-reqts-09.txt, August 2001.

[4] 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-10, January 2002.

[5] David B. Johnson et al., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", (work in progress) draft-ietf-manet-dsr-07, February 2002.

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

[7] Jorjeta G et al., "A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks", (work in progress) draft-ietf-manet-simple-mbcast-01, July 2001.

[8] T. Narten, E. Nordmark and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC2461 , Dec 1998.

[9] S. Thompson and T. Narten, "IPv6 Stateless Address Autoconfiguration" , RFC2462 , December 1998.

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

[11] Jung-Soo Park, Yong-Jin Kim and Sung-Woo Park, "Stateless address autoconfiguration in Mobile Ad Hoc Networks using site-local address", draft-park-zeroconf-manet-ipv6-00, July 2001.

[12] Levon Esibov and Dave Thaler, "Link-Local Multicast Name Resolution (LLMNR)", (work in progress) draft-ietf-dnsext-mdns-09, February 2002.

[13] Matt Crawford, "Transmission of IPv6 Packets over Ethernet Networks", RFC2464, December 1998.

[14] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture", (work in progress) draft-ietf-ipngwg-addr-arch-v3-07, November 2001.

[15] Jung-Soo Park, Myung-Ki Shin and Yong-Jin Kim, "Link Scoped IPv6 Multicast Addresses", (work in progress)draft-ietf-ipv6-link-scoped-mcast-00, April 2002.

[16] B. Haberman and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", (work in progress) draft-ietf-ipngwg-uni-based-mcast-03, October 2001.

[17] A. Gulbrandsen, P. Vixie and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC2782, February 2000.

[18] UCL Mbone Conferencing Applications, http://www-mice.cs.ucl.ac.uk/multimedia/software/