DHCP for IPv6

Charles E. Perkins and Jim Bound
Sun Microsystems, Inc.
Palo Alto, CA 94303
Digital Equipment Company
Nashua, NH 03062

Abstract

The Dynamic Host Configuration Protocol (DHCPv6) provides a framework for passing configuration information, via extensions, to IPv6 nodes. It offers the capability of automatic allocation of reusable network addresses and additional configuration flexibility. This protocol should be considered a stateful counterpart to the IPv6 Stateless Address Autoconfiguration protocol specification, and can be used separately or together with the latter to obtain configuration information.

1 Introduction

The Dynamic Host Configuration Protocol for IP version 6 (DHCPv6 [1], or in this document sometimes DHCP) provides configuration parameters to Internet nodes. DHCP consists of a protocol for delivering node-specific configuration parameters from a DHCPv6 server to a client, to be used for allocation of network addresses and other related parameters to IPv6 [2] nodes. DHCP supports optional configuration parameters and processing for nodes through extensions [7] that are added to various DHCPv6 messages, but do not affect the protocol control flow of those messages.

IPv6 provides new protocol features not available in the current version of IP, IP version 4 (IPv4) [9], and corrects a number of IPv4's deficiencies. These improvements are used to simplify and generalize the operation of DHCPv6 clients. IPv6 is more fully described in several documents undergoing standardization within the IETF, including the IPv6 Protocol [2], the IPv6 Addressing Architecture [3], IPv6 Stateless Address Autoconfiguration [10], and Neighbor Discovery [6].

IPv6 defines two classifications of address autoconfiguration, *stateless* and *stateful*. DHCPv6 is the currently specified protocol for stateful address autoconfiguration. It is expected to be used by IPv6 nodes which attach to the Internet at any IPv6 *managed link*, in other words a link on which nodes are expected to use stateful address autoconfiguration. Every IPv6 node equips itself with a *link-local* address, which is unique on the link to which the node is attached. This uniqueness is guaranteed by the operation of Duplicate Address Detection (DAD), using the methods of Stateless Address Autoconfiguration.

The next section provides an overview of the protocol design model that guided the design choices in the specification; Section 3 provides the message formats and field definitions used for each message. Details about how clients, servers, and relays interact are included at appropriate places in the description of the message formats. The general format of DHCPv6 extensions is described in section 4, and the IP address extension is detailed in section 5. Section 6 discusses some of the security considerations and techniques used with DHCPv6. Section 7 discusses the differences between DHCPv4 and DHCPv6, and the last section mentions a few directions about the future of DHCPv6.

2 Protocol Design Model

This section provides an architectural perspective for the DHCPv6 protocol design model. Goals, conceptual models, and some terminology are presented.

In DHCP, an *agent* is either a server or a relay. The server keeps a *binding* for each client, which records all the resources allocated to that client. DHCP client bindings are indexed by the client's link-local address and its agent's prefix. The latter is obtained from the DHCP Advertisement message which the DHCP server used to create the binding, and is typically built from the client's MAC address as described above.

2.1 Design Goals

Since the whole idea is to simplify administration and avoid additional configuration, DHCPv6 should not require manual configuration of DHCPv6 clients, except as dictated by security requirements. Each client should be able to obtain, from a DHCPv6 server, appropriate local configuration parameters without user intervention.

The DHCPv6 relay is enables clients and servers to carry out DHCP protocol transactions. DHCP Solicit messages are issued by the relay when initiated by prospective DHCPv6 clients. By default, the relay locates DHCPv6 servers by use of multicasting client solicitations to the All-DHCP-Servers multicast address. After initialization, clients typically rely on unicast messages for their new configuration needs or address lifetime, minimizing the need for multicast renewals.

DHCP is constructed so that a relay does not have to maintain any state when mediating DHCP client/server interactions, because relays are likely to be part of enterprise infrastructure routers which will need to operate as efficiently as possible. All relays must send DHCP Request messages using the source IPv6 address from the interface at which the DHCP request was received. This allows the DHCP server to identify the link from which the DHCP request was originated by the client, corresponding to the function available using the previous *giaddr* field in DHCPv4.

2.2 Processing Model

To find a server, a client multicasts a DHCP Solicit from the interface which it wishes to configure. The client then awaits a DHCP Advertise message, containing an IP address for a DHCPv6 server. This advertisement may be multicast by a DHCP relay to all servers, as illustrated in figure 1. Multiple, overlapping DHCPv6 servers can enhance reliability, reduce latency, and increase performance.

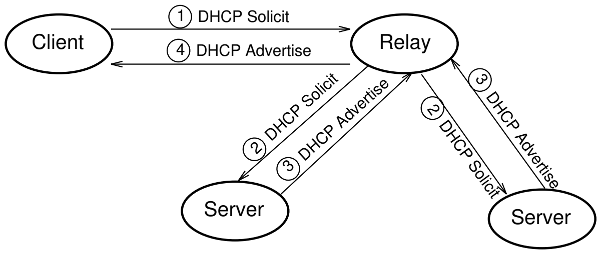

Figure 1. Getting DHCP Server addresses

After the client knows a DHCPv6 server's address, it starts a transaction by unicasting a DHCP Request to the server. The server then unicasts a DHCP Reply to the client (possibly via a relay), as illustrated in figure 2. After the Reply is received by the client, the transaction is complete. To provide a method of recovery if either the client or server does not receive its message, the client retransmits each Request message until it elicits the corresponding Reply, or until it can be reasonably certain that the desired DHCPv6 server is unavailable. Since UDP is used instead of TCP, a simple retransmission algorithm is required. Therefore, the responder in any transaction must be prepared to issue exactly the same answer when it receives such a retransmission. The responder optimistically assumes its answer was delivered unless it receives a retransmission of the original message.

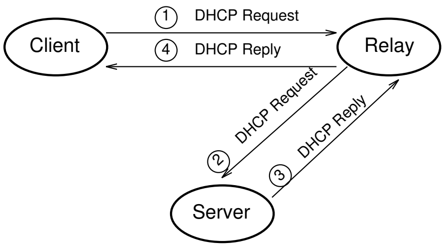

Figure 2. Sending a Request to a known Server

3 DHCPv6 Message Formats and Field Definitions

DHCPv6 uses the UDP [8] protocol (ports 546 and 547) to communicate between clients and servers. The source port for DHCPv6 messages is arbitrary. The following well-known multicast groups are joined by DHCP agents:

AllDHCPAgentsAddrOnLink	(FF02:0:0:0:0:0:1:2)
AllDHCPServersAddr	(FF05:0:0:0:0:0:1:3)
AllDHCPRelaysAddr	(FF05:0:0:0:0:0:1:4)

3.1 DHCP Solicit Message Format

A DHCPv6 client transmits a DHCP Solicit message (illustrated in figure 3) over the interface it is trying to configure, to obtain one or more DHCPv6 server addresses. Typically, when there is no DHCPv6 server on the link, such a request may be forwarded by a DHCPv6 relay attached to this link (if such a relay exists) on behalf of a client to a DHCPv6 server.

The DHCPv6 client sends the solicitation to the All-DHCP-Agents multicast address, setting the relay address to zero. Any DHCPv6 Relay receiving the solicitation must complete it and send it to the All-DHCP-Servers multicast address. The relay must copy a non-link-local address of its interface from which the client's solicitation was received into the relay address field.

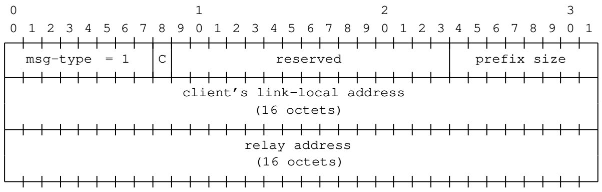

Figure 3. DHCPv6 Solicit Message Format

The soliciation instructs DHCPv6 servers to send their advertisements to the prospective client.

If a client sets the 'C' bit in the solicitation, previous allocations at forgotten servers can be reclaimed by those servers receiving the soliciation. This feature can help avoid the loss of resources allocated to clients that subsequently reboot or crash without issuing any DHCP Release message back to their servers.

3.2 DHCP Advertise Message Format

A DHCPv6 agent sends a DHCP Advertise message (illustrated in figure 4) to inform a prospective client about the IPv6 address of a server to which a Request message may be sent. When the client and server are on different links, the server sends the advertisement back through the DHCPv6 Relay whence the solicitation came.

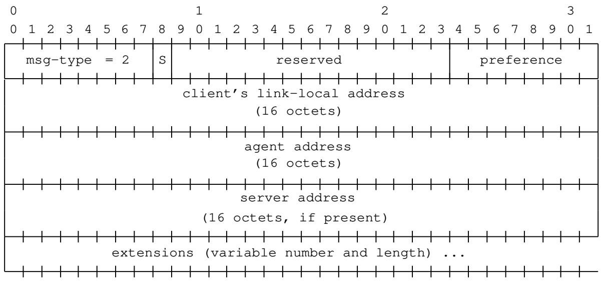

Figure 4. DHCPv6 Advertise Message Format

Suppose that a DHCPv6 server on the same link as a client issues the DHCP Advertise in response to a DHCP Solicit message sent to the All-DHCP-Agents multicast address. Then the agent address will be an IPv6 address of one of the server's interfaces on the same link as the client, and the 'S' bit will be set to zero. In this case, no server address will be present in the DHCP Advertise message.

The DHCPv6 server must copy the client's link-local address into the advertisement which is sent in response to a DHCP Solicit. Both the server address (if present) and agent address of the DHCP Advertise message must have sufficient scope to be reachable by the DHCPv6 client. Moreover, the agent address of any DHCP Advertise message sent by a DHCPv6 relay must not be a link-local address. In situations where there are no routers sending Router Advertisements, a DHCPv6 server must be configured on the same link as prospective clients. The DHCPv6 protocol design does not apply to situations where the client has no way to route messages to a server not on the same link.

Unless a DHCP Advertisement is received with a "server preference" field of 255, a DHCPv6 client must wait ADV_CLIENT_WAIT seconds after issuing the DHCP Solicit message in order to receive the Advertisement with the highest preference. After waiting for that period of time, a client must select the highest preference DHCPv6 server as the target of its DHCP request. If a DHCPv6 client receives an advertised preference of 255, it does not have to wait for any more advertisements. If a DHCPv6 client sends a DHCP Request to a highly preferred DHCPv6 server but fails to receive a DHCP reply from that server after following the retransmission algorithm, the client may subsequently attempt to send a DHCP Request to a less preferred server.

A DHCPv6 client is free to cache the result of any DHCP Advertisement it hears. However, it should be noted that this is purely a potential performance enhancement as the results need not be constant over time. Hence it may not get a response if it uses a cached server address, and it may have to subsequently transmit a new DHCP Solicit message.

3.3 DHCP Request Message Format

In order to request configuration parameters from a DHCPv6 server, a client sends a DHCP Request message (illustrated in figure 5), and may append extensions [7] to tell the server some details about the particular resources desired. Typically when a client reboots, it is not configured with a valid DHCPv6 server address, and it must first obtain one by multicasting a DHCP Solicit message (see Section 3.1). Moreover, since the client does not have a valid IPv6 address of sufficient scope for the DHCPv6 server to communicate with the client, the client must unicast the message to the local DHCPv6 relay and insert the DHCPv6 relay address as the agent address in the message header. The client could not send the message directly to the DHCP server because the server could not return any response to the client.

Otherwise, if the client can be reached by a server on the same link, the client may omit the server address in the DHCP Request message. In this case, the client must clear the S-bit in the DHCP Request message and send it directly to to the server, using the server address as the IPv6 destination address in the IPv6 header; no DHCP relay is needed.

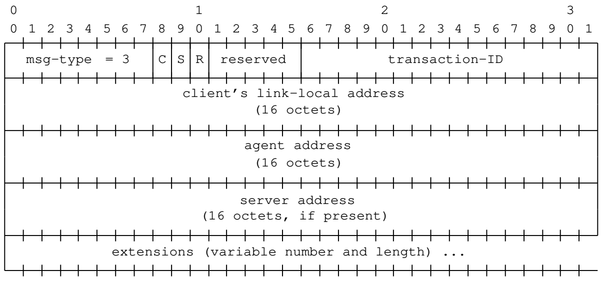

Figure 5. DHCPv6 Request Message Format

The transaction-ID included with the request is used to identify replies. The server is expected to remember recent transaction-IDs and send back the same reply to the client if it gets a retransmitted DHCP Request message. This requires that the server keep track of the transaction-IDs that it has received from the client. When the client reboots, it tells the server to wipe out its transaction-ID records by setting the 'R' bit. The request message also allows the client to tell the indicated server that all previous allocations can be reclaimed by setting the 'C' bit, in a way meant to resemble the meaning of the same bit in the DHCP Solicit message.

For each pending DHCP Request message, the client must make dutiful attempts to receive the server's reply message. In order to retransmit a request properly, a client must maintain the following information:

- The transaction-ID of the request message,
- The server address,
- The agent address (which can be the same as the server address),
- The time at which the next retransmission will be attempted, and
- All extensions appended to the request message.

3.4 DHCP Reply Message Format

The server sends one DHCP Reply message (illustrated in figure 6) in response to every DHCP Request or DHCP Release received. If the request comes with the 'S' bit set, the client could not directly send the Request to the server and had to use a neighboring relay agent. In that case, the server sends back the DHCP Reply with the 'L' bit set, and the DHCP Reply is addressed to the agent address found in the DHCP Request message. If the 'L' bit is set, then the client's link-local address will also be present, and the relay will use the link-local address to deliver the Reply message to the client.

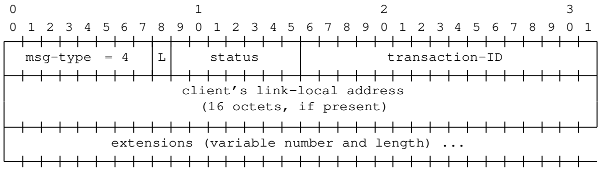

Figure 6. DHCPv6 Reply Message Format

The status code in the Reply message tells the client either that its request succeeded, or else the reason for the failure.

3.5 DHCP Release Message Format

The DHCP Release message (illustrated in figure 7) is sent without the assistance of any DHCPv6 relay. Other than that, the Release message is somewhat similar to the DHCP Request message, except that it asks the server to release resources instead of allocate them. When a client sends a Release message, it is assumed to have a valid IPv6 address with sufficient scope to allow access to the target server. If parameters are specified in the extensions, only those parameters are released; otherwise, all client allocations are released. The DHCPv6 server acknowledges the Release message by sending a DHCP Reply (see section 3.4).

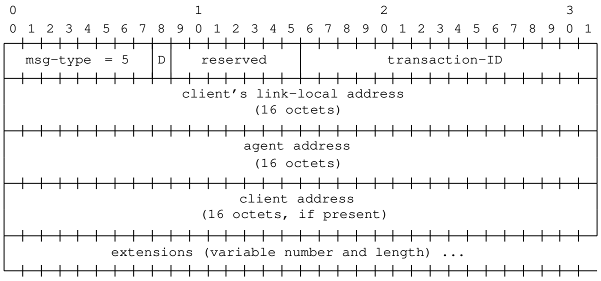

Figure 7. DHCPv6 Release Message Format

The client link-local address and the agent address are enough to allow the DHCP server to identify the correct client-server resource record.

Typically the client has an IPv6 address that will still be valid after the server performs the operations requested in the extensions to the DHCP Release message, and which has sufficient scope to be reachable from the server. In that case, and only then, the client should set the 'D' flag. When the 'D' flag is set, the server must send the DHCP Reply back to the client using the client address field of the Release message. Otherwise, when the 'D' bit is not set, the server must send its DHCP Reply message to the agent address in the Release message, so that the relay agent can subsequently forward the Reply back to the releasing client at the client's link-local address indicated in the Reply message. Note that it is an error to include within the DHCP Release message both the 'D' bit and an IP address extension (see section 5) which has the IPv6 address used as the client IP address field of the DHCP Release message header. If the client's link-local address and agent address do not match a client binding an error will be returned to the client.

3.6 DHCP Reconfigure Message Format

Reconfigure messages can only be sent to clients which have established a reachable IPv6 address. Hence, the DHCP Reconfigure message (illustrated in figure 8) is sent without the assistance of any DHCPv6 relay. Only the parameters which are specified in the extensions to the Reconfigure message need be requested again by the client. A Reconfigure message can either be unicast or multicast by the server. Using the appropriate extension, a server can assign a multicast address to its clients just for the purpose of enabling efficient transmission of a Reconfigure message to a large population of DHCP clients. If the client fails to listen for DHCP Reconfigure messages, it is possible that the client will not receive notification that its DHCPv6 server has deallocated the client's IPv6 address and/or other resources allocated to the client.

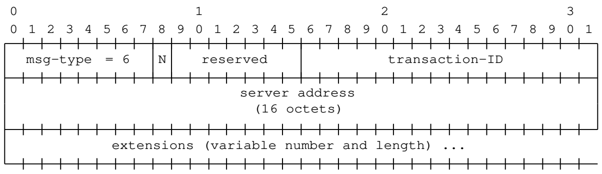

Figure 8. DHCPv6 Reconfigure Message Format

Transactions initiated by the reconfigure message are unique in that they typically use the same transaction-ID for three different message types – the reconfigure message itself, and the subsequent DHCP Request message followed by the DHCP Reply from the server.

If the client transmits a DHCP Request in response to a DHCP Reconfigure message (see section 3.6), the client can continue to operate with its existing configuration information and resources until it receives the corresponding DHCP Reply from the server. The same retransmission rules apply as for any other DHCP Request message from the client. When the server sets the 'N' bit in the DHCP Reconfigure message, a DHCP Request sent in response to a DHCP Reconfigure message will not elicit a DHCP Reply message from the server. In this case, the server agrees to take responsibility to insure that the client has received the Reconfigure message. This will be useful to speed site renumbering operations.

4 DHCPv6 Extension Field Format

Extensions may be fixed length or variable length. All extensions begin with a type field, which is two octets long and uniquely identifies the extension. Every extension except extension type 65535 has a length field. The value of the Length field does not include the four octets specifying the type and length, and it is present even if the length is fixed or zero.

Unrecognized extensions are to be skipped by ignoring the number of octets specified in the length field, and processing continued for subsequent extensions. By default, DHCP entities assume that that the maximum DHCP message size including extensions is 1500 octets.

5 IP Address Extension

The IP Address extension (illustrated in figure 9) is the most essential of all the DHCPv6 extensions, and the only one that we will describe in this paper. It can be used by both client and server in various ways. Since the IP Address extension can be used more than once in the same DHCP message, all information relevant to a particular IPv6 allocation has to be collected together in the same extension. Some of the fields within the IP Address extension enable DNS to be updated [12].

To renew or extend the lifetime of a particular IP address, the client puts that address in the client address field. To request the allocation of a new but unspecified IP address, the client omits the client address field. The IP address returned by the server in the latter case will be compatible with a subnet prefix of the link to which the client is currently attached. Each IP Address Extension can contain at most one IP address, but multiple extensions can be used.

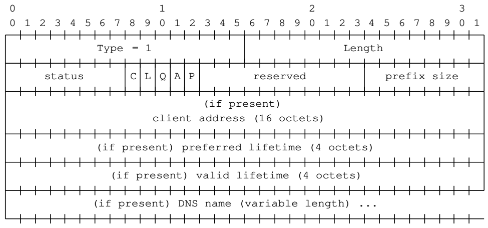

Figure 9. IP Address Extension for DHCPv6

If the 'C' bit is set, the field containing the IP address for the client is present in the extension. If the client address is present (the 'C' bit is set), a nonzero prefix size is the number of leftmost bits of the client's IPv6 address which make up the routing prefix. Otherwise, if the 'C' bit is not set, prefix size must be zero.

If the 'L' bit is set, the preferred and valid lifetimes are present in the extension. If the 'A' bit is set, the client requests a DNS update to the client's AAAA record, using the client's fully-qualified domain name (FQDN) [5, 12]. If the 'P' bit is set, the client requests that that the the server updates DNS with a new PTR record, as specified by the client's FQDN. The DHCP Server will usually perform such updates, because both the DHCPv6 server and the DNS server will be under tighter administrative control than random DHCPv6 clients will be. A number of failure conditions have to do with possible interactions between the DHCP Server and the DNS server in these circumstances.

If the 'Q' bit is set, the fields included by the client are required, and must be provided, or the extension must be rejected. If the 'A' bit is also set, the server must ensure that the DNS is updated with a new AAAA record, as specified by the client's FQDN, before responding with the corresponding DHCP Reply. Likewise, if the 'Q' and the 'P' bits are both set, the server must ensure that the DNS is updated with a new PTR record, as specified by the client's FQDN, before responding with the corresponding DHCP Reply. A DHCP client can include an IP address in its IP Address extension and set the 'A' bit and/or 'P' bit to ask the DHCP Server to use that address for updating DNS. This can be done even with IP addresses obtained by Stateless Address Autoconfiguration [10].

6 Security

DHCPv6 clients and servers often have the need to authenticate the messages they exchange. For instance, a server may wish to be certain that a request was originated from the client identified by the <link-local address, agent address> within the message. Conversely, it is quite often essential for a DHCPv6 client to be certain that the configuration parameters and addresses it has received were sent to it by an authoritative DHCPv6 server. Similarly, a DHCPv6 server should only accept a DHCP Release message which seems to be from one of its clients, if it has some assurance that the client actually did transmit the Release message. Again, a client might wish to only accept DHCP Reconfigure messages that are certain to have originated from a server with authority to issue them.

The IPv6 Authentication Header [4] can provide security for DHCPv6 messages when both endpoints have a suitable IPv6 address. However, a client often has only a link-local address, and such an address is not sufficient for a DHCPv6 server which is off-link. In those circumstances the DHCPv6 relay is involved, so that the DHCP message must have the relay's address in the IPv6 destination address field, even though the client aims to deliver the message to the DHCPv6 server. The DHCP Client-Server Authentication Extension [7] is intended to be used in these circumstances.

7 DHCPv6 Compared to DHCPv4

Many differences derive from use of IPv6. The link-local address permits a node to have an address immediately when the node boots, which means all clients have a source IP address at all times to locate a server or relay agent on the local link. DHCP's stateful autoconfiguration has to coexist and integrate with stateless autoconfiguration supporting Duplicate Address Detection and the two IPv6 lifetimes, to facilitate the dynamic renumbering of addresses and the management of those addresses. Multiple addresses per interface are inherently supported in IPv6. Some DHCPv4 options are no longer necessary because the configuration parameters are either obtained through IPv6 Neighbor Discovery or the Service Location protocol [11].

Other changes are due to a different architectural approach. IPv6 Address allocations are now handled in a message extension as opposed to the main header. Servers implement client/server bindings, taking advantage of the client's link-local address. Servers are discovered by a client solicitation and server advertisement model. The client will know if the server is on-link or off-link. The client receives the addresses of available servers in an advertisement either from the server itself, if the client and server are on the same link, or from a relay. The relay obtains server addresses from system configuration or by the use of a site wide DHCPv6 Multicast packet. The protocol is does not use ACKs or NAKs. The server assumes the client receives its responses unless it receives a retransmission of the same client request. This permits recovery in the case where the network has faulted.

The old function of DHCPINFORM is inherent in the new protocol design; a client can request configuration parameters other than IPv6 addresses in the optional extension headers. Clients are expected to listen to their UDP port for the new Reconfigure message from servers. Dynamic Updates to DNS are supported in the IPv6 Address extension.

With the changes just enumerated, we can support new user features, including

- Configuration of Dynamic Updates to DNS
- Address deprecation, for dynamic renumbering.
- Authentication
- Configuration of additional DHCPv6 addresses, for instance if the DHCPv6 client is itself a Web server.
- Reclaiming addresses allocated with too-long lifetimes using the Reconfigure message.
- Integration between stateless and stateful address autoconfiguration.

8 Summary and Future Work

IPv6, while flexibly specified to enable automatic and stateless address autoconfiguration, also needs to serve the needs of network administrators for more configuration control and options. DHCPv6 has been designed to work with the widest range of enterprise needs, from single links without any routers (or DHCP relays), to networks spread across multiple sites. The advanced features of IPv6 enable us to offer a simple yet powerful mechanism for all such configuration and resource allocation needs. Moreover, we have been able to use experience from DHCP for IPv4, and offer additional features and more freedom for the issuance and processing of client requests.

In the future, we expect that a DHCPv6 *server to server* protocol will be specified for tighter coordination of the handling of overlapping IPv6 address pools. Methods for configuring DHCPv6 servers and relays are also expected to become more important as deployment of DHCPv6 proceeds.

Acknowledgements

Thanks to the Dynamic Host Configuration working group (dhc) of the Internet Engineering Task Force for their many improvements to this protocol.

References

[1] J. Bound and C. Perkins. Dynamic Host Configuration Protocol for IPv6. draft-ietf-dhc-dhcpv6-14.txt, April 1998. (work in progress).

[2] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, December 1995.

[3] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 1884, December 1995.

[4] Stephen Kent and Randall Atkinson. IP Authentication Header. draft-ietf-ipsec-auth-header-03.txt, November 1997. (work in progress).

[5] P. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034, November 1987.

[6] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP version 6 (IPv6). RFC 1970, August 1996.

[7] C. Perkins. Extensions for the Dynamic Host Configuration Protocol for IPv6. draft-ietf-dhc-dhcpv6ext-10.txt, April 1997. (work in progress).

[8] J. B. Postel. User Datagram Protocol. RFC 768, August 1980.

[9] J. B. Postel, Editor. Internet Protocol. RFC 791, September 1981.

[10] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 1971, August 1996.

[11] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan. Service Location Protocol. RFC 2165, July 1997.

[12] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS). RFC 2136, April 1997.