A Study of Mobility Support in IPv6

Ajith Prabhakara
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX 76019
ajith@uta.edu

Abstract

IP version 6 (IPv6) is being designed within the IETF as a replacement for the current version of the IP protocol, IP version 4 (IPv4), used in the Internet. One of the primary issues that the design of IPv6 addresses is mobility support in the wireless environment. This paper discusses the options that the current IPv6 specification [1] has proposed that support the requirements of wireless mobile environments and mobility management protocols. Further, these options have been discussed with reference to Mobile IP [2]. This paper also provides, wherever relevant, a comparison between the IPv4 and IPv6 implementations of the Mobile IP protocol.

This paper has been written for the partial fulfillment of the requirements of the CSE6392 – Mobile Computing Systems class, Fall 2002, offered Dr. Mohan Kumar of the Department of Computer Science and Engineering in the University of Texas at Arlington.

****

1. Introduction

The rapid growth of the current Internet, which operates using IPv4, has created a number of problems for the administration and operation of the global network. These problems include decreasing number of IPv4 addresses for network nodes, rapid growth of memory and performance requirements for network routers, and as the use of a variety of mobile nodes grows exponentially, the lack of support for issues concerned with mobility. While changes to IPv4 have extended the life of the current Internet, these changes have created new problems and require significant amount of overhead for network administration. IPv6 has been designed to support these extensions, and more, without creating additional problems.

The new IP is based on a simple philosophy: the Internet could not have been so successful in the past years if IPv4 had contained any major flaw [4]. IPv4 was a very good design and IPv6 has indeed keep most of its characteristics. However, based on the problems faced and improvements suggested in the last 10 years or so, IPv6 is not just a simple derivative of IPv4, but a definitive improvement.

In addition, the number of mobile devices/nodes accessing the Internet has been growing steadily over the last few years. This along with the growing importance of the Internet and the Web in day-to-day life clearly demonstrates the need to pay attention to mobility [5]. IPv6 has various extension headers, including Routing header and Destination options header, which alleviate mobility and routing problems experienced in IPv4. Also, it provides mechanisms, like Neighbor Discovery [6] and Stateless Address Autoconfiguration [7], that have been specifically

added to support mobility. Moreover, all IPv6 nodes are expected to implement strong authentication and encryption features to improve Internet security.

The tremendous proliferation of mobile devices/nodes equipped for connectivity in the past few years has established beyond doubt that the future of computing and the Internet lies in the mobile environment. With concepts such as Pervasive computing and "anytime-anywhere" connectivity, which have gone beyond Ubiquitous computing, taking off in a big way, the importance of network connectivity (wired or wireless) being transparent to the movement of mobile devices/nodes is more than ever before. In response to these challenges, researchers and standardization bodies alike have been working on protocols and mechanisms which will make the above mentioned concepts a reality. Mobile IP is one of the most predominant approaches in this direction. The initial proposals of this protocol was based on IPv4. With IPv6's proposed features for mobility support, Mobile IP will both be able perform much more efficiently and smoothly.

This paper discusses the changes in the new IP protocol, IPv6, with respect to mobility support. These changes are discussed with reference to Mobile IP. A comparison between the performances of Mobile IP based on IPv4 and the new IPv6, wherever relevant, has been provided.

Section 2 of this paper presents a description of IPv6's mobility support. This section also discusses, wherever possible, the changes that the IPv6 proposal has made to similar features that existed in IPv4 and the reason behind making those changes. Section 3 outlines the Mobile IP protocol and the changes in its operation due to IPv6. This section also includes a comparison with the earlier version of the above mentioned protocol which was based on IPv4. A discussion of the trends in IPv6 deployment and conclusions are presented in Section 4.

2. Mobility support in IPv6

In this section some of the basic characteristics of IPv6 that are particularly relevant to mobility have been outlined. The most visible difference is that IPv6 addresses are all 128 bits long, instead of 32 bits long as in IPv4. Within this huge address space a tiny part is reserved for all current IPv4 addresses, and another tiny part is reserved for the *Link-Local* addresses.

Link-local addresses are not routable, but are guaranteed to be unique on a link (i.e, on a local network). Nodes on the same link can communicate with each other, even without any routers, by using their Link-Local addresses.

IPv6 defines several kinds of *extension headers*, which may be used to include additional information in the headers of an IP packet. The defined IPv6 headers include:

- Routing header
- Destination options header
- Authentication header
- Hop-by-hop options header
- Fragment header and
- Encrypted security payload.

Of these the first three headers are of particular importance to mobility and have been discussed later in this section.

IPv6 nodes discover each other's presence, as well as each other's link-layer (i.e, MAC) addresses, by participating in the Neighbor Discovery protocol; IPv6 nodes also discover local routers and network prefixes by means of Neighbor Discovery. Stateless Address

Autoconfiguration allows mobile IPv6 nodes to automatically configure a globally routable address, irrespective of their point of attachment to the Internet.

2.1 Extension Headers

The IPv4 header had room for options, allowing special case treatment of some packets. Options however, fell gradually out of use, mostly because of performance effects.

The most common way to improve router performance is to concentrate on the most frequent packets, and allow them to travel through quickly. Packets with options, by definition, cannot travel through a router quickly because they require special treatment. This means that they will be placed in a secondary queue and handled when the router is relatively free. Thus using options results in a performance penalty, which explains their gradual falling out of use.

However, there are good reasons for packets to require special treatment. IPv6 specifies such special treatment by the use of extension headers.

All IPv6 headers are the same size and look pretty much the same. The difference is in the Next Header field. In an IPv6 packet with no extensions headers, the value of the field will refer to the next layer's protocol. In other words, if the IP packet contains a TCP segment, the Next Header field will contain an eight-bit value (six) from RFC 1700, Assigned Numbers. If the packet contains a UDP datagram the value will be 17. Table 1 shows some of the possible values for the Next Header field.

Next Header value	Description
0	Hop-by-hop header
43	Routing header (RH)
44	Fragmentation header (FH)
51	Authentication header (AH)
52	Encapsulated Security payload (ESP)
59	No next header
60	Destination Options header

Table 1. Some possible values that indicate an extension header in an IPv6 Next Header field. [8]

The Next Header field will indicate whether, and what, the next extension header is. Thus IPv6 headers can be chained together, starting from the IPv6 header itself and linking extension headers. Figure 1 shows how these header chains can develop. The first packet shown has no extension headers, but the second packet shows a Routing header extension, followed by the TCP header and the rest of the packet. The last packet shown demonstrates a more complex possible header chain, with a Fragment header extension appended to the IPv6 header, followed by an Authentication header extension, followed by an Encapsulating Security payload header extension, finally followed by the TCP heard and the rest of the packet.

A. Routing Header

The Routing Header replaces source routing as it was implemented in IPv4. Source routing allows the specification of the routers that the packet must traverse on its way to its destination. In IPv4 source routing, using IPv4 options, was limited by the number of intermediate routers that could be specified (no more than ten 32-bit addresses fit into the extra 40 bytes allowed for

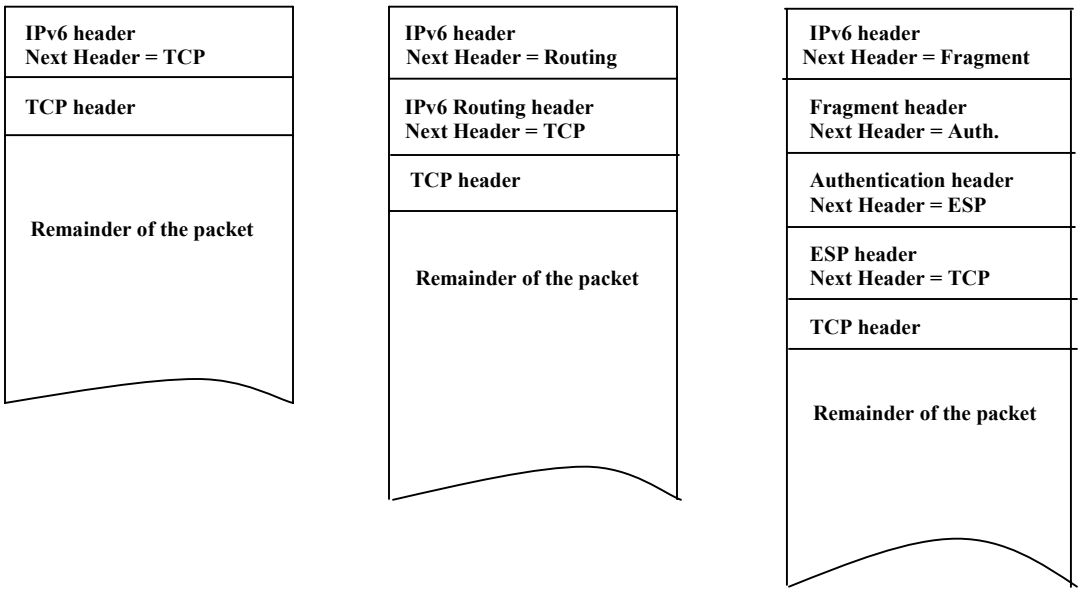

Figure 1: The Daisy-Chain of headers in IPv6. The figure shows 3 different IPv6 packets having different sets of headers.

IPv4 header extensions). Furthermore, processing source-routed packets tended to be slow, because every router in the path had to process the entire address list, whether the router was on the list or not.

IPv6 defines a generic Routing extension header, with two one-byte fields: a routing type field, indicating what kind of Routing header is in use, and a segments-left field, which indicates how many additional routers listed in the rest of the header must still be visited before the packet reaches it ultimate destination. The rest of the header is type-specific data, which may vary depending upon the type of Routing header specified. RFC 1883 defines one type, the type 0 Routing header.

The type 0 Routing extension header fixes problems, both major problems, with IPv4 source routing. The Routing header itself is processed only as it arrives at each router in the list, and there can be as many as 256 nodes specified in the list. The Routing header works something like this:

- The source node builds a list of routers that the packet must traverse and then builds a type 0 Routing header. This includes the list of routers and the final destination node address and the number of segments left (an eight-bit integer) indicating the number of specified routers that must still be visited before the packet can be delivered to its final destination.
- When the source node transmits the packet, it sends the IPv6 header destination address equal to the address of the first router in the Routing header list.
- The packet is forwarded until it arrives at the first stop in the path. Intermediate routers ignore the Routing header; only when the packet arrives at the IPv6 destination (a router) does that router examine the Routing header.
- At the first stop (and all other stops), the router checks to make sure that the number of segments left jibes with the address list. If the value of segments left is equal to zero, it means that the node is actually the final destination of the packet, as the node can then continue processing the rest of the packet.

- Assuming that the node is not the final destination and everything else checks out, it takes its own address out of the IPv6 header destination and replaces it with the node that is next in the Routing header list. The node also decrements the segments-left field, and sends the packet on its way to the next stop where the process is repeated.

B. Destination options header

There will be two ways to add functionality to IPv6. The first will be to define a new extension header type that will be carried transparently throughout the network and will be only be examined by the station specified in the destination address. This first extension method has two inconveniences:

i) It requires the allocation of header-type numbers. There are only 256 such numbers (its an 8-bit field), and they are used for extension headers as well as payload types such as UDP and TCP. They are a relatively scarce resource that should not be wasted.

ii) It requires that both source and destination understand the new option. If a station does not understand the type of a header, it can only reject the whole packet.

The IPv6 options header obviates these inconveniences. The Destination options header provides a mechanism to deliver optional information along with IPv6 packets. Rather than attempt to define additional specific extension headers, the Destination options header makes it possible to define new options intended for the destination node. The format used by the Destination options header is beyond the scope of this paper, and can be referred in [1] and [4].

Only two options, two padding options, have been defined so far. A description of these two padding options is not relevant to the scope of this paper. However, a few other options, which have not yet been published as RFC's, have been defined in Internet Drafts, particularly related to Mobile IP [9]. These options have been discussed later in this paper.

C. Authentication header

The IPv6 and IP Security specifications require all IPv6 nodes to implement authentication [1], which greatly improve security within the Internet. This also provides a secure environment in which mobile nodes can get attached to and leave different domains/sites. The primary security header used in IPv6 is the Authentication header. It is important to remember that this header has been defined for use in IPv4 also, in which case it is added to the normal IPv4 header as an option.

In IPv6, the Authentication header has been defined as a generic extension header. It is typically inserted between the IPv6 header and the end-to-end payload. For example, an authenticated TCP packet will contain an IPv6 header, an Authentication header, and the TCP packet itself. The Authentication header can be used to do the following:

- Provide strong integrity services for IP datagrams. This means that the AH can be used to carry content verification data for the IP datagram.
- Provide strong authentication for IP datagrams. This means that the AH can be used to link an entity with the contents of the datagram.
- Provide nonrepudiation for IP datagrams, assuming that a Public Key digital signature algorithm is used for integrity services.
- Protect against replay attacks through the use of a sequence number field.

2.2 Neighbor Discovery

IPv6 doesn't do Address Resolution Protocol (ARP) or Reverse ARP (RARP) anymore. These protocols are used in IPv4 to figure out what IP address is associated with what link-local network address, in other words, to link an Ethernet MAC address (as an example) with an IP address of its node.

A characteristic of IPv6 Neighbor Discovery is that it is defined in generic terms as a part of the IPv6 ICMP. This is a major difference from IPv4, for which a different address resolution protocol could be defined for each new media. An ICMP message, like any other IPv6 message, can only be transmitted if the host knows the media address of the destination. This is solved by the use of multicast transmission. As long as the media address of a destination remains unknown, messages are sent to multicast addresses. The media level multicast address is supposed to be determined by an algorithm which varies from media to media.

The description of the Neighbor Discovery procedures will assume the host maintains four separate caches:

- The Destination's cache has an entry for each destination address toward which the host recently sent packets. Its associates the IPv6 address of the destination with that of the neighbor toward which the packets were sent.
- The Neighbor's cache has an entry for the immediately adjacent neighbor to which the packets were recently relayed. It associates the IPv6 address of that neighbor with the corresponding media address.
- The prefix list includes the prefixes that have been recently learned from Router Advertisements.
- The router list includes the IPv6 addresses of all routers from which advertisements have been recently received.

The list of routers is infact built from the source addresses of the router advertisement messages received by the station. Routers, however, typically have more than one IP address, which is a potential cause for confusion. To avoid confusion, routers are thus requested to always use the link-local address assigned to the interface from which the advertisement messages are sent.

To transmit a packet, the host must find out the next hop for the packets destination. The next hop shall be neighbor directly connected to the same link as the host. The host should then find a valid media address for that neighbor.

In many cases, a packet will already have been sent to the destination, and the neighbor address will be found in the destination's cache. If this is not the case, the host will check whether one of the cached prefixes matches the destination address. If this is the case, the destination is local; the next hop is the destination itself. In other cases, the destination is probably remote. The host should then select a router from the table of routers and use it as the next hop.

Once the next hop has been determined, the corresponding entry is added to the destination's cache and the neighbor's cache is looked up to find the media address of that neighbor. At this stage there are four possibilities.

i) If there is no entry found for that neighbor in the cache, the host should send a Neighbor Solicitation message. The neighbor is then added to a new cache line whose status is set to incomplete.

ii) If there is already an entry for that neighbor, but its status is incomplete, the host should wait for the completion of the procedure to learn the media address and send the packet.
iii) If there is a complete line in the cache for the neighbor, the media address is known and the packet can be sent immediately.
iv) If the neighbor's entry in the cache has not been refreshed for a long time, its status is suspect. The media address can be used, but a neighbor solicitation message should be sent.

The source address of a solicitation message is always set to the link-local address of the interface. The destination address, however, is not the IPv6 address of the solicited neighbor, but a multicast address, the solicited node multicast address. The solicited node multicast address is formed by concatenating a fixed prefix and the last 24 bits of the node's IPv6 address. All nodes are expected to compute the solicited node multicast addresses corresponding to each of their configured addresses, including link-local addresses, and to join the corresponding multicast groups.

The solicitation message will be received by all hosts that have at least one address whose last 24 bits match those of the solicited neighbor address. Each of them will examine the target address parameter. If it matches one of its addresses, the solicited host will reply with a Neighbor Advertisement message.

The other special ICMP messages defined by Neighbor Discovery include:

i) Router Advertisement messages
ii) Router Solicitation messages and
iii) Redirect messages.

The first two messages will be discussed in detail in the context of Stateless Address Autoconfiguration. Redirect messages are sent by routers to notify hosts that they are not the best router (i.e., next hop) to use for a particular destination.

Neighbor Discovery as defined for IPv6 serves different purposes:

- Router discovery
- Prefix discovery
- Parameter discovery
- Address Autoconfiguration
- Next-hop determination
- Neighbor unreachability detection
- Duplicate address detection and
- Redirects

C. Stateless Address Autoconfiguration

This is mechanism provided by IPv6 that allows individual nodes to figure out what their IP configuration should be without having to explicitly query any server. This is the one of the features of IPv6 that is extremely beneficial with respect to mobile nodes.

Since IPv6 nodes use the IEEE EUI-64 link layer addresses [10], it is reasonably certain that the host ID will be unique. All a node has to do is find out what its own link layer address is, calculate its EUI-64 address and then figure out what is the IP address of its IPv6 network. This can be done by asking the nearest router for this information.

IPv6 nodes start initializing their behavior by joining the *all nodes* multicast group. This is done by programming their interfaces to receive all the packets sent to the corresponding

multicast address, FF02::1 [4]. They will then send a Router Solicitation message to the routers on the link, using the *all routers* address, FF02::2 [4], as the destination and their own link-local address as the source. When a router receives such a solicitation message, they are supposed to reply with a Router advertisement message. The message will be sent toward the link layer address of the requestor, using the content of the source address option as the media address.

Router Advertisement messages are not only sent in response to solicitations; routers transmit them at regular intervals on the *all hosts* multicast address.

To make sure that there are conflicts among addresses after a node autoconfigures its address, a Duplicate Address Detection mechanism is provided by IPv6. This mechanism uses the Neighbor Discovery procedures. Once an address has been configured, the host sends a Neighbor Solicitation message toward that address and waits for one second. If another station has been configured with the same address, it will reply to the solicitation exposing the collision. At this point the host will know that its number is not unique and that the corresponding address should not be used. It will the pick a new number (if that is possible) or display an error message an ask for human intervention.

The absence of a reply might be caused by the absence of a collision, but it may also be due to the loss of the initial Neighbor Solicitation message or of the reply itself. Stations are supposed to retry the procedure several times before being reasonably sure of the uniqueness of their token. These trials should be sufficiently spaced not to cause link layer congestion. According to the current specification [7], the message should be sent only once, after a delay chosen at random between zero and three seconds.

3. Mobile IPv6

The first few sub-sections of this section highlight the differences between the certain primary mechanisms of Mobile IP in IPv4 and in IPv6. This discussion is important to be able to understand and appreciate the changes in Mobile IPv6. Later on, the basic operation of Mobile IPv6 is described; then the Binding Update and Acknowledgement options are discussed and finally Proxy Neighbor Advertisements and Home Agent Discovery mechanisms are described.

Mobility Support in IPv6, as proposed by the Mobile IP working group, follows the design for Mobile IPv4. It retains the ideas of a home network, home agent, and the use of encapsulation to deliver packets from the home network to the mobile node's current point of attachment.

Each mobile node is assigned a (permanent) IP address in the same way as any other node, and this IP address is known as the mobile node's home address. A mobile node's home address remains unchanged regardless of where the node is attached to the Internet. The IP subnet indicated by this home address is the mobile node's home subnet, and standard IP routing mechanisms will deliver packets destined to a mobile node's home address only to the mobile node's home subnet. A mobile node is simply any node that may change its point of attachment from one IP subnet to another, while continuing to be addressed by its home address. Any node with which a mobile node is communicating is referred to as a correspondent node, which itself may be either mobile or stationary.

3.1 Discovery of a Care-of-Address

A mobile node's current location while away from home is known as its care-of address, which is a globally routable address acquired by the mobile node. While discovery of a care-of address is

still required, a mobile node can configure its a care-of address by using Stateless Address Autoconfiguration and Neighbor Discovery. Thus, foreign agents are not required to support mobility in IPv6. IPv6-within-IPv6 tunneling is also already specified [11]. The association of a mobile node's home address with a care-of address, along with the remaining lifetime of that association, is known as a binding.

3.2 IPv6 Encapsulation and Tunneling

While away from its home subnet, a router on the mobile node's home subnet known as its home agent maintains a record of the current binding of the mobile node. The home agent then intercepts any packets on the home subnet addressed to the mobile node's home address and tunnels them to the mobile node at its current care-of address. This tunneling uses IPv6 encapsulation [11], and the path followed by a packet while it is encapsulated is known as a tunnel.

3.3 Route Optimization

Once a correspondent node has learned the mobile node's care-of address, it may cache it and route its own packets for the mobile node directly there using an IPv6 Routing header [1], bypassing the home agent completely. The authors of Mobile IPv6 believe it is reasonable to expect every IPv6 node to perform the extra steps of caching and using the care-of address for each mobile node with which it is communicating, since the additional overhead of doing so is quite small; the problems of triangle routing are then avoided by having each correspondent node route its own packets for a mobile node directly to its care-of address.

Mobile IPv6 introduces a set of new IPv6 Destination options, called Binding Update and Binding Acknowledgement, to manage these cache entries as needed. The Binding Update and Binding Acknowledgement options conform to an option mechanism already present in IPv6 [1]. IPv6 mobility borrows heavily from the route optimization ideas specified for IPv4 [12], particularly the idea of delivering binding updates directly to correspondent nodes. Processing binding updates can be implemented as a fairly simple modification to IPv6's use of the destination cache [6]. The Binding Update and Binding Acknowledgement options have been discussed in detail later in this section.

3.4 Source Routing

In contrast to the way in which route optimization is specified in IPv4, in IPv6 correspondent nodes do not tunnel packets to mobile nodes. Instead, they use IPv6 Routing Header, which implements a variation of IPv4's *source routing* option. A number of early proposals for supporting mobility in IPv4 specified a similar use of source routing options, but two main problems precluded their use:

- IPv4 source routing options require the receiver of source-routed packets to follow the reversed path to the sender back along the indicated intermediate nodes. This means that malicious nodes using source routes from remote locations within the Internet could impersonate other nodes, a problem exacerbated by the lack of authentication protocols.
- Existing routers exhibit terrible performance when handling source routes. Consequently, the results of deploying other protocols that use source routes have not been favorable.

However, the objections to the use of source routes do not apply to IPv6, because IPv6's more careful specification eliminates the need for source-route reversal and lets routers ignore options that do not need their attention.

Consequently, correspondent nodes can use routing headers without penalty. This allows the mobile node to easily determine when a correspondent node does not have the right care-of address. Packets delivered by encapsulation instead of by source routes in a routing header must have been sent by correspondent nodes that need to receive binding updates from the mobile node.

Unlike IPv4 source routing options, in IPv6, the Routing Header is not examined or processed until it reaches the next node identified in the route. Thus processing of source routed packets in IPv6, as explained in Section 2, is much faster compared to those in IPv4.

It is a further point of contrast to route optimization in IPv4 that, in IPv6 mobility support, the mobile node delivers binding updates to correspondent nodes instead of to the home agent. In IPv6, key management between the mobile node and correspondent node is more likely to be available [13].

3.5 Basic Operation of Mobile IPv6

Each time the mobile node moves its point of attachment from one IP subnet to another, the mobile node will configure its care-of address by Stateless Address Autoconfiguration, or alternatively by some means of Stateful Address Autoconfiguration such as DHCPv6 [14] or PPPv6 [15]. The decision about which manner of automatic address configuration to use is made according to the methods of IPv6 Neighbor Discovery. A mobile node may have more than one care-of address at a time, for example if it is link-level attached to more than one (wireless) network at a time or if more than one IP network prefix is present on a network to which it is attached. The association between a mobile node's home address and its care-of address, along with the remaining lifetime of that association, is known as a binding. The central data structure used in Mobile IPv6 is a cache of mobile node bindings, maintained by each IPv6 node, known as a Binding Cache.

While away from home, a mobile node registers one of its bindings with a router in its home subnet, requesting this router to function as the home agent for the mobile node. The care-of address in this binding registered with its home agent is known as the mobile node's primary care-of address, and the mobile node's home agent retains this entry in its Binding Cache, marked as a “home registration” until its lifetime expires. While it has a home registration entry in its Binding Cache, the home agent uses proxy Neighbor Discovery, described later in this section, to intercept any IPv6 packets addressed to the mobile node's home address on the home subnet, and tunnels each intercepted packet to the mobile node's primary care-of address indicated in this Binding Cache entry. To tunnel the packet, the home agent encapsulates it using IPv6 encapsulation.

In addition, Mobile IPv6 provides a mechanism for IPv6 correspondent nodes communicating with a mobile node, to dynamically learn the mobile node's binding. The correspondent node adds this binding to its Binding Cache. When sending a packet to any IPv6 destination, a node checks its Binding Cache for an entry for the packet's destination address, and if a cached binding for this address is found, the node routes the packet directly to the mobile node at the care-of address indicated in this binding; this routing uses an IPv6 Routing header instead of IPv6 encapsulation, as this adds less overhead to the size of the packet. (The home

agent cannot use a Routing header, since adding one to the packet at the home agent would invalidate the authentication in any IPv6 Authentication header included in the packet by the correspondent node.) If no Binding Cache entry is found, the node instead sends the packet normally (with no Routing header), and the packet is then intercepted and tunneled by the mobile node's home agent as described above.

Mobile IPv6 introduces two new IPv6 Destination options to allow a mobile node's home agent and correspondent nodes learn and cache the mobile node's binding. After configuring a new care-of address, a mobile node must send a Binding Update option containing that care-of address to its home agent, and to any correspondent nodes that may have an out-of-date care-of address for the mobile node in their Binding Cache. Receipt of a Binding Update must be acknowledged using a Binding Acknowledgement option, if an acknowledgement was requested in the Binding Update.

3.6 The Binding Update Option

A mobile node sends a Binding Update to another node to inform it of its current binding. As noted in Section 3.3, since the Binding Update is sent as an IPv6 Destination option, the mobile node may include it in any existing packet (such as a TCP packet) that it is sending to that destination node, or it may send the Binding Update in a packet by itself. The Binding Update is used by a mobile node to register its primary care-of address with its home agent, and to notify correspondent nodes of its binding so that they may create or update entries in their Binding Cache for use in future communication with the mobile node.

A Binding Update should be considered a form of routing update; handled incorrectly, a Binding Update could be a source of security problems due to the possibility of remote redirection attacks. Therefore, packets which carry a Binding Update must also include an IPv6 Authentication header, which provides authentication and replay protection for the Binding Update.

The format of the Binding Update option is shown in , however, a discussion of the only those fields that relevant to the scope of this paper have been discussed below.

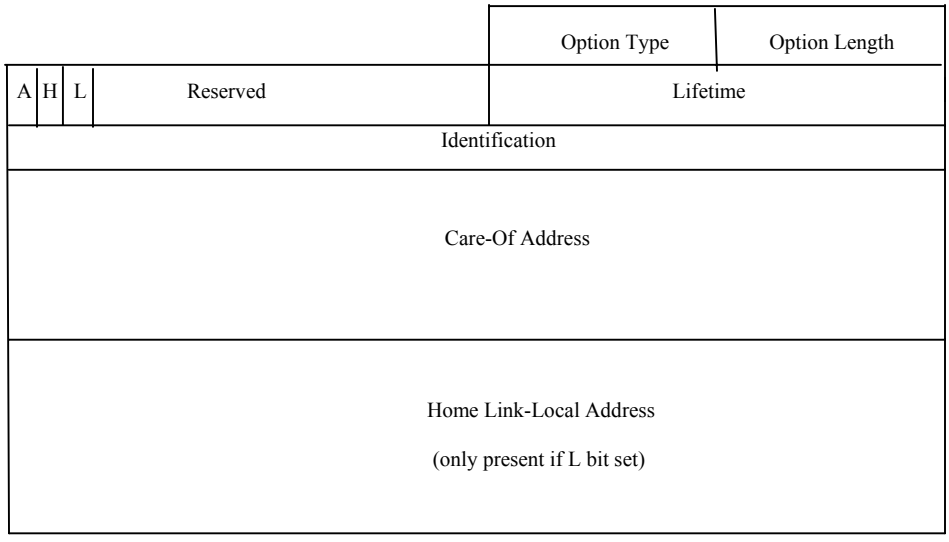

Figure 2: Binding Update Header format.

When the Home Agent (H) bit is set in the Binding Update, the mobile node requests the destination of this Binding Update to serve as its home agent, with the binding contained in the Binding Update as the mobile node's primary care-of address. This mechanism is used for router assisted smooth handoffs. The mobile node may also include the Link-Local address it used when last on its home network, to cause the home agent to send proxy Neighbor Advertisements for this address as well as for the mobile node's home address; the mobile node must set the Home Link-Local Address Present (L) bit to indicate that the Home-Link Local Address field has been included in the Binding Update. The mobile node may request a Binding Acknowledgement be returned for this Binding Update, by setting the Acknowledge (A) bit in the Update.

3.7 The Binding Acknowledgement Option

The Binding Acknowledgement option is sent by a node to acknowledge receipt of a Binding Update. When a node receives a Binding Update addressed to itself, in which the Acknowledge (A) bit is set, it must return a Binding Acknowledgement; other Binding Updates may also be acknowledged but need not be. The destination address in the IPv6 header of the packet carrying the Binding Acknowledgement must be the care-of address from the Binding Update, causing the Binding Acknowledgement to be returned directly to the mobile node sending the Binding Update.

In the case in which the mobile node is returning to its home subnet, the Binding Update sent to its home agent must specify the mobile node's home address as the care-of address. The mobile node must also send the appropriate IPv6 Neighbor Advertisement messages with the Override flag set, so that its neighbors on its home subnet will update the link-layer address for the mobile node in their Neighbor Caches. The mobile node must do this for both its Link-Local address and its home address. The Neighbor Advertisement messages can be repeated a small number of times to guard against occasional loss of packets on the home subnet. Receipt of this Neighbor Advertisement message overrides the previous proxy Neighbor Discovery actions taken by the home agent when the mobile node earlier left its home subnet.

3.8 Proxy Neighbor Advertisements

When a mobile node first registers with its home agent after leaving its home subnet, the home agent must send onto the home subnet a gratuitous Neighbor Advertisement on behalf of the mobile node, with the Override flag set, giving its own link-layer address as the associated link-layer address for the mobile node's IPv6 home address. All nodes on the home subnet receiving this Neighbor Advertisement will then update their Destination Cache entry for the mobile node to contain the link-layer address of the home agent, allowing the home agent to intercept any packets sent by them to the mobile node. Some correspondent nodes on the home subnet may have established connections with the mobile node by using the mobile node's Link-Local address, and thus the home agent must also send a gratuitous Neighbor Advertisement for the mobile node's Link-Local address as well. Each of these gratuitous Neighbor Advertisements should be repeated a few times to increase reliability, although any node that does not receive one of these Advertisements will be able to detect the change in the mobile node's link-layer address using IPv6 Neighbor Unreachability Detection.

In addition, while the mobile node is away from home, the home agent should act as a proxy for the mobile node, replying to any received Neighbor Solicitation messages received for the mobile node's home address or Link-Local address. These proxy Neighbor Advertisement replies ensure that correspondent nodes on the home subnet retain the home agent's link-layer address in their Destination Cache while the mobile node is away from home.

3.9 Home Agent Discovery

It is useful to be able to send a Binding Update to a mobile node's home agent without explicitly knowing the home agent's address. For example, since the mobile node was last at home, it may have become necessary to replace the node serving as its home agent due to the failure of the original node or due to reconfiguration of the home subnet. It thus may not always be possible or convenient for a mobile node to know the exact address of its own home agent.

Mobile nodes can dynamically discover the address of a home agent by sending a Binding Update to the IPv6 anycast address on their home subnet. A packet sent to an anycast address is received by exactly one router on the destination subnet. Any router on the home subnet which receives this Binding Update (responding to the anycast address) must reject the Binding Update and include its own (unicast) IPv6 address in the Binding Acknowledgement indicating the rejection. The mobile node will then repeat its Binding Update, sending it directly to the router that returned the rejection.

4. Conclusion

Information presented at the recent general assembly meeting of the Address Council that is part of the IANAs' Address Supporting Organization shows three scenarios for growing demand for mobile IP addresses. Each scenario indicates that the demand will increase steadily over the next few years, perhaps reaching 1 billion in 2003 and 1.27 billion in 2005.

With the usage of mobile devices/nodes growing rapidly, IPv6 is certainly going to play a major role in the future development of the Internet. Stateless Address Autoconfiguration requires much less manual less management than manual configuration, or even DHCP. Also Neighbor Discovery enables a lot more than just initial address configuration. A big difference between IPv4 and IPv6 is that IPv6 interfaces may be configured with several addresses. This allows organizations to connect to several providers at the same time, to arbitrage in real-time between these providers. This facility is missing from IPv4, in which one interface usually has one address. Also, IPv6's extension headers provide a rich set of options for IPv6 packets, yet make sure that they do not bog down the network core infrastructure.

IPv6 has security features integrated into it, which will play a critical role as transactions that occur in the mobile environment sky-rocket. It also takes care of establishing security associations and managing keys as mobile users join and leave wireless networks.

Work has also been progressing in adapting various other mobility management protocols to IPv6. Leading among such efforts is Cellular IPv6. Cellular IP [3] was proposed to the IETF by researchers from Columbia University and Ericsson in 1999. Apart from the Mobile IP protocol engine, Cellular-IP-enabled mobile hosts run an additional Cellular IP protocol engine. Thus, Cellular IP is a Mobile IP protocol extension and not a replacement. Cellular IP primarily overcomes the weaknesses of Mobile IP in case of mobile hosts that migrate between domains frequently. It attempts to alleviate this weakness by introducing hierarchies into the IP mobility

infrastructure. Currently Cellular IPv6 is still in its developmental stages and can be referred to in Internet Drafts [16].

Despite all this IPv6 has still not taken off as was predicted earlier. There are various reasons: the cost involved in upgrading software and infrastructure is prohibitive for most operators, in some regions (like North America) the need for IPv6 addresses is not foreseen in the near future. Its predicted that there are enough IPv4 addresses to satisfy the expected growth rate in the North American region until 2005, thus ISP's have been using mechanisms such as NAT and avoiding the transition to IPv6. In regions like Asia, due to the profusion of mobile devices, such as cell phones and PDA's, there is an acute shortage of IP addresses. However, sooner or later, IPv6 will be the universal standard.

With the future of computing clearly moving towards the mobile environment, IPv6 provides the necessary platform to develop new applications and technologies on.

References

[1] S. Deering, R. Hinden. "Internet Protocol version 6 (IPv6) specification". RFC 2460, December 1998.

[2] C.E. Perkins. "Mobile IP". IEEE Communications, Vol 35, No 5, 1997, pp. 84-99.

[3] A.G. Valko. "Cellular IP: a new approach to Internet host mobility". ACM SIGCOMM Computer Communication Review, Vol 29, Issue 1, January 1999, pp. 50-65.

[4] C. Huitema. "IPv6: The New Internet Protocol". Second edition. 1998, Prentice Hall Ptr. ISBN 0-13-850505-5

[5] S. Bradner, A. Mankin. "The Recommendation for the IP Next Generation protocol". RFC 1752, January 1995.

[6] T. Narten, E. Nordmark, W. Simpson. "Neighbor Discovery in IP version 6 (IPv6)". RFC 2461, December 1998.

[7] S. Thomson, T. Narten. "IPv6 Stateless Address Autoconfiguration". RFC 2462, December 1998.

[8] P. Loshin. "IPv6 Clearly Explained". 1999, Morgan Kaufmann Publishers. ISBN 0-12-455838-0

[9] C.E. Perkins and D.B. Johnson. "Mobility support in IPv6". In Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom'96), Rye, New York, USA, November 1996. ACM.

[10] IEEE. "Guidelines for the 64-bit Global Identifier (EUI-64) Registration Authority". Web Document, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html, March 1997.

[11] A. Conta, S. Deering. "Generic Packet Tunneling in IPv6 Specification". RFC 2473, December 1998.

[12] C.E. Perkins, D.B. Johnson. "Route Optimization in Mobile IP". Draft-ietf-mobileip-optim-11.txt. Mobile IP Working Group, September 2001.

[13] H. Orman. "The OAKLEY Key Determination Protocol". RFC 2412, November 1998.

[14] J. Bound, M. Carney, C.E. Perkins, T. Lemon, B. Volz, R. Droms. "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)". Draft-ietf-dhc-dhcpv6-23.txt, February 2002.

[15] D. Haskin, E. Allen. "IP Version 6 over PPP". RFC 2472, December 1998.

[16] Z. D. Shelby, D. Gatzounas, A. T. Campbell, and C-Y. Wan, "Cellular IPv6," draft-shelby-seamoby-cellularipv6-00, Work in Progress, November 2000.