Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/ULANC/DS/4.1.1/A1**
Contractual Date of Delivery to the CEC:	April 30th 2002
Actual Date of Delivery to the CEC:	May 2nd 2002
Title of Deliverable:	Survey and evaluation of MIPv6 implementations
Work package contributing to Deliverable:	WP4
Type of Deliverable*:	R
Deliverable Security Class**:	PU
Editors:	Martin Dunmore and Christopher Edwards
Contributors:	Tziouvaras Chrysostomos, Oliver Krämer, Piers O'Hanlon, Reinhard Ruppelt

* Type: P - Prototype, R - Report, D - Demonstrator, O - Other

** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Abstract:

This document provides a survey and evaluation of existing Mobile IPv6 (MIPv6) implementations. At the time of writing MIPv6 continues to be refined by the IETF mobileip working group and has not yet reached RFC status. MIPv6 is therefore not yet a standard and current implementations vary in both their completeness and to which draft version of the standard they are intended to support. This document provides a survey on how many host/router MIPv6 implementations are available and evaluates the MIPv6 features that are supported.

Keywords:

MIPv6, implementation, host, router

Executive Summary

Workpackage 4 of the 6NET project aims to identify and provide all the features necessary for supporting the sophisticated IPv6 applications and services to be trialled in Workpackage 5. Activity 4.1 is concerned with Mobile IPv6 (MIPv6) and, more specifically, the configuration, testing and possible enhancement of MIPv6 over the large IPv6 network testbed of 6NET.

In this relatively early stage of the 6NET project, one of the first tasks of Activity 4.1 is to investigate the available MIPv6 implementations that we may wish to deploy within the 6NET testbed. Consequently, this deliverable provides a survey and evaluation of existing MIPv6 implementations. The MIPv6 implementation survey consists of all the implementations that could be found at the time of writing. The ensuing evaluation is restricted to those implementations with which consortium partners have existing knowledge of and/or experience with. At this early stage of the project, there has been little opportunity, or available resources to evaluate previously untried MIPv6 implementations.

From our investigation, it is evident that existing MIPv6 implementations have varying levels of support for MIPv6 features and differ to which draft version they are based upon. In most cases, MIPv6 functionality does not come 'built-in' and must be either applied as a separate software patch or explicitly enabled on the target system. Furthermore, few implementations have support for IPsec and a suitable key distribution algorithm. However, the incomplete nature of current MIPv6 implementations is only to be expected at a time when implementers are at various stages of supporting a protocol that is still being designed. Despite this, our investigations have shown that the existing implementations will be of great benefit to Activity A4.1 in particular, and the 6NET project in general. Whilst the deployment of MIPv6 on a production network may be some years away, the deployment of MIPv6 within the experimental 6NET network will be greatly aided by the experimental MIPv6 implementations currently available.

Whilst reading this document, it is important to understand that this is a 'window in time' of the development of the MIPv6 protocol. Hence, both the status of the MIPv6 standard itself and the implementations that are reported in this deliverable may soon become outdated after the time of writing.

Table of Contents

1 Introduction

One of the main issues that 6NET (and the IPv6 community in general) has is the lack of a Mobile IPv6 "standard". The definition of the Mobile IPv6 protocol has been "work in progress" for a number of years, but has not, as yet, reached IETF RFC (Request for Comment) status. The reasoning behind this is rather subjective, and can be put down to a number of reasons, more recently in terms of the way the messaging that the protocol defines is made secure. At the time of writing, the Mobile IPv6 protocol is at the 16th version of Internet Draft.

What this implies is that it is difficult to ensure compatibility within the area. The changes taking place between subsequent releases of the Internet Draft have ranged from rather minor editorial enhancements to significant changes that effectively render successive implementations incompatible.

Whilst reading this document, it is important to understand that this is a snapshot of the status of the MIPv6 protocol at the time of writing. Leading figures within the Internet community believe that the protocol is very close to reaching standards basis, and that the current version of the draft could well be the one that moves to RFC. At the time of writing, this remains to be seen. Either way, the status of MIPv6 implementations reported in this document may soon become outdated as implementers strive for compliance with the latest MIPv6 IETF Internet Draft.

The rest of this document is structured as follows. The next section provides an overview of the MIPv6 protocol, describing the major concepts, the differences compared to Mobile IPv4 and recent changes to MIPv6. Section 3 gives a survey of MIPv6 implementations that are known about at the time of writing. Section 4 then goes on to provide a more detailed description and evaluation of the features offered by some of these implementations. Finally, a summary of the MIPv6 implementations that we investigated and associated conclusions are provided in section 5.

2 Overview of Mobile IPv6

The Mobile IPv6 (MIPv6) protocol [13] is a proposed standard by the IETF to provide transparent host mobility within IPv6. The protocol enables a Mobile Node to move from one network to another without the need to change its IPv6 address. A Mobile Node is always addressable by its *home address*, which is the IPv6 address that is assigned to the node within its home network. When a Mobile Node is away from its home network, packets can still be routed to it using the node's home address. In this way, the movement of a node between networks is completely invisible to transport and other higher-layer protocols.

2.1 Comparison with Mobile IPv4

The development of IPv6 [18] allowed the Mobile IP working group to start afresh, and with the benefit of hindsight, design a mobility support protocol which is more tightly integrated with IP than that of Mobile IPv4 (MIPv4) [19]. MIPv6 therefore has several differences to its IPv4 counterpart that provide a simpler, more streamlined protocol. Yet, the basic design of MIPv6 is largely based upon the original Mobile IP support in IPv4 and thus shares many features with Mobile MIPv4.

Although the basic concepts of protocol operation are the same between MIPv4 and MIPv6, there are two implicit assumptions made by MIPv6 that result in a much simpler and efficient protocol. Firstly, MIPv6 has no concept of a foreign agent. Instead it relies on standard IPv6 features, such as router discovery and stateless address configuration to enable the direct detection and utilisation of foreign networks by Mobile Nodes. Secondly, MIPv6 assumes greater intelligence in Correspondent Nodes, and resultantly, has route optimisation built in as a fundamental part of the Mobile IPv6 protocol. This dramatically improves the routing efficiency to Mobile Nodes, by effectively eliminating triangular routing.

The major differences between MIPv4 and MIPv6 can be summarised as:

- No need for foreign agents
- Route optimisation as standard
- Integrated support – COA and ingress filtering
- Destination options
- COA and multicast routing
- Detection of 'black holes'
- Use of the Routing Header – less overhead than tunnelling
- Use of Neighbour Discovery rather than ARP
- No need for tunnel soft-state
- Use of IPv6 anycast for Home Agent discovery
- Router Advertisement Interval option.

2.2 Functional Overview

Mobile Nodes participating in the MIPv6 protocol each have a persistent *home address*, which can be used to address the Mobile Node irrespective of its current point of attachment to the IPv6 network. The IPv6 network which matches the home address' prefix is known as the *home network*. Mobile Nodes also adopt a *Home Agent* - an IPv6 capable router directly connected to the home network. This process may either be static, or dynamic, via the MIPv6 Home Agent discovery mechanism. Like Mobile IPv4, the Home Agent is responsible for the interception and forwarding of IPv6 packets to the Mobile Node which are incorrectly routed to the home network while the Mobile Node is away from home.

As in MIPv4, Mobile Nodes attached to their home network operate as any other network node, so no special routing is required. Mobile Nodes migrating to a foreign network use IPv6 autoconfiguration to discover the new network, and to allocate a care-of address within the address space of that network. However, to ensure that IPv6 packets destined for the Mobile Node's home address reach the proper location as efficiently as possible, the routing information pertaining to the Mobile Node's home address must be updated in both the Home Agent and any relevant Correspondent Nodes. MIPv6 provides this functionality by the introduction of a bindings cache, and four new IPv6 destination options. It should be noted that the following information is based on the version 15 of the MIPv6 draft standard [12]. Changes made to MIPv6 in the latest draft version [13] are so recent (March 2002) that they are not featured in any of the implementations discussed in this document. For this reason, this functional overview of MIPv6 concentrates on version 15. For the record, the changes to MIPv6 in draft 16 are outlined below in section 2.3.

2.2.1 Bindings Cache

The relationship between a Mobile Node's home address and its current care-of address is known as a *binding*. All Nodes participating in MIPv6 are required to maintain a table of these bindings in a *binding cache*. One entry is held in the binding cache for each Mobile Node with which communication is currently taking place. The binding cache holds four pieces of information per binding which are central to the operation of MIPv6, as illustrated by Table 1 (other fields are present to ensure correct ordering of control messages, but these are omitted for clarity). The home address forms the key field of the cache.

Home Address	Care of Address	Lifetime	Home Agent
3ffe:2101:0:b00::10	3ffe:2101:0:a00:260:97ff:fe8b:4c56	120	Yes
3ffe:2101:0:b00::15	3ffe:2101:0:b00:a00:6aff:fe2b:137c	43	No

Table 1 Mobile IPv6 Bindings Cache

When a node wishes to transmit an IPv6 packet to a remote host, the home address field of the binding cache is searched for the IPv6 address of that host. If no match is found, the packet is transmitted according to the normal IPv6 routing tables. However, if a match is found, then the packet is encapsulated prior to transmission, to redirect the packet to the care-of address specified in the binding cache. This ensures optimal routing to the Mobile Node's current location. The form this encapsulation takes is dependant on the state of the 'Home Agent' flag stored in the binding cache entry.

2.2.2 Home Agent Operation

If the 'Home Agent' flag is set in a binding cache entry, then the node maintaining that cache is acting as a Home Agent for the Mobile Node. If this is the case, and the packet was entered through a forwarding context, then the packet is encapsulated using IPv6 in IPv6 tunnelling, as illustrated in Table 2.

IPv6 Header(Outer)	IPv6 Header (Inner)	Transport Header	Payload
Source Address: Home Agent	Source Address: Correspondent Node	TCP/UDP	Data
Dest. Address: Care of Address	Dest. Address: Home Address		
← 40 bytes →	← 40 bytes →		

Table 2 IPv6 in IPv6 Encapsulation

IPv6 tunnelling is used by Home Agents to forward IPv6 packets misrouted to a Mobile Node's home network while it is away from home, as illustrated in Figure 1. Tunnelling has the advantage of preserving the complete original IPv6 packet, which is important as any modification to an IPv6 header could cause problems with higher layer protocols, such as TCP. Intercepted packets are tunnelled directly to a Mobile Node's current care-of address, where they are subsequently decapsulated by the Mobile host.

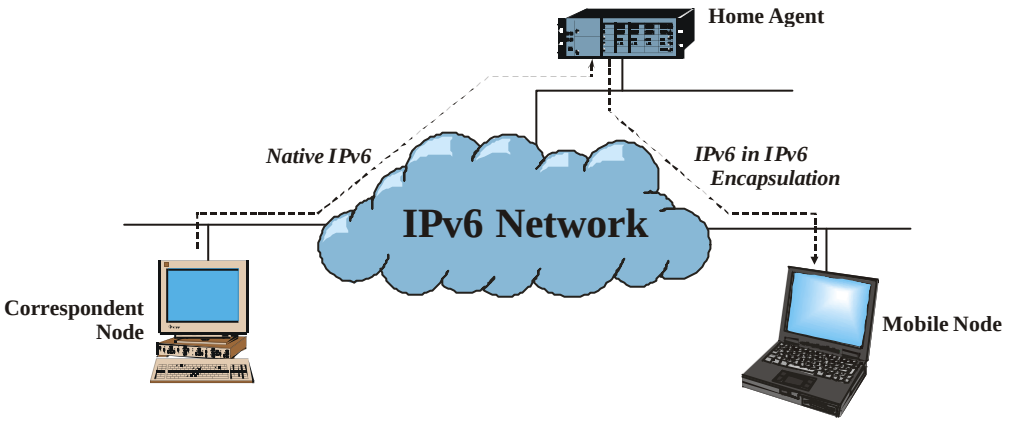

Figure 1 MIPv6 Routing to Mobile Nodes (Pre Route Optimisation)

2.2.3 Correspondent Node Operation

If the 'Home Agent' flag is cleared in a binding cache entry, or the packet was not received from a forwarding context, then an IPv6 routing header is used to redirect the IPv6 packet through the relevant care-of address, as shown in Figure 2.

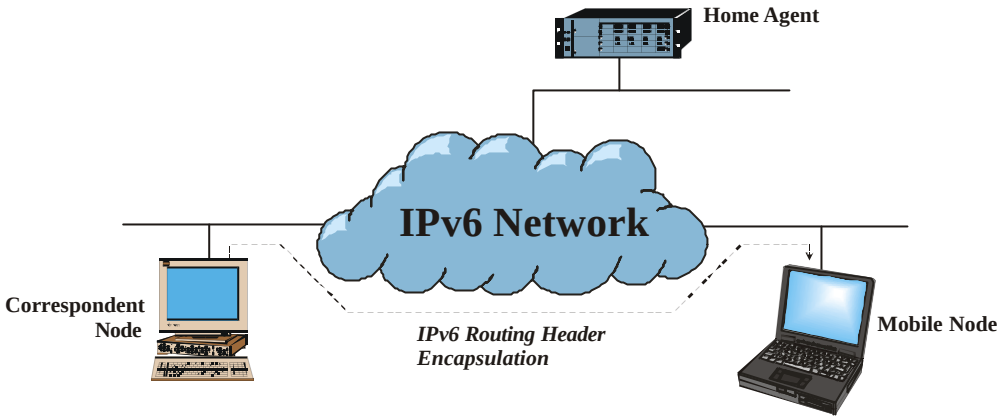

Figure 2 - Mobile IPv6 Routing to Mobile Nodes (Post Route Optimisation)

IPv6 Header	IPv6 Routing Header	Transport Header	Payload
Source Address: Correspondent Node Dest. Address: Care of Address	Next Hop Address: Home Address	TCP/UDP	Data
◄——— 40 bytes ———►	◄——— 24 bytes ———►		

Table 3 IPv6 Routing Header Encapsulation

As can be seen from Table 3, the use of the IPv6 routing header reduces the effective bandwidth required for the encapsulation of the packet in comparison to IPv6 in IPv6 tunnelling by 16 bytes. This reduction in packet size is possible due to spatial redundancy - i.e. if IPv6 tunnelling were used then the IPv6 source address of both the inner and outer IPv6 headers would be identical, resulting in a waste of bandwidth.

2.2.4 Binding Cache Coherence

The use of the binding cache and IPv6 encapsulation provides a mechanism to enable optimal routing to Mobile hosts. This mechanism, however, relies on the bindings contained within that cache being accurate and up to date. Indeed, to protect against total machine failure (which is common in a mobile environment due to battery life constraints, etc.) and long periods of network disconnection by Mobile Nodes, binding cache entries for a Mobile Node must persist even after a period of total disconnection or loss of state by Mobile or Correspondent Nodes.

Mobile IPv6 maintains binding cache coherence through the use of binding update, binding acknowledgement and binding request messages. The remainder of this section describes these messages in detail, and how they interoperate to provide accurate and timely binding cache coherence.

Binding update, acknowledgement and request messages are all carried inside IPv6 destination options, each with their own destination option type. Utilising IPv6 destination options gives several advantages over less integrated methods of control messaging. Firstly, messages can be placed inline with the header of existing IPv6 packets, thereby reducing the packet transmission overhead of the message. Secondly, as no transport layer protocol is involved in the transmission of the message, fewer issues exist concerning the blocking of control messages by firewalls.

2.2.4.1 Binding update messages

Binding updates are transmitted by Mobile Nodes to Home Agents and Correspondent Nodes to create or update the entry in their binding cache relating to that Mobile Node's home address.

Binding updates can be generated at any time by Mobile Nodes, but are always transmitted upon the detection of an IPv6 packet which has travelled via an IPv6 in IPv6 tunnel from that node's Home Agent. The reception of such a packet indicates that the Correspondent Node that generated the packet currently has no binding for this Mobile Node (else the packet would have been delivered via an IPv6 routing header). In order to guarantee that 'stale' bindings are not indefinitely maintained by binding caches, Mobile IPv6 employs a soft state mechanism to purge out of date bindings. Every binding update contains a lifetime field, which specifies, in seconds, how long the binding is valid for. After this lifetime expires, the binding is removed from the binding cache. The lifetime value is set and refreshed by the corresponding lifetime field contained within binding update messages. A lifetime value of zero in a binding update indicates removal of the relevant binding.

2.2.4.2 Binding acknowledgement messages

Unlike binding updates, binding acknowledgements are sent to Mobile Nodes by Correspondent Node and Home Agents. They provide control feedback to Mobile Nodes in response to binding updates, and are used to provide reliable binding update delivery and to indicate any errors which are generated during the remote processing of binding updates. Mobile Nodes match binding acknowledgements with their corresponding binding updates by the comparison of sequence numbers.

2.2.4.3 Binding request messages

Correspondent Nodes and Home Agents detecting an entry in their binding cache which is nearing expiry may decide to send a binding request message to the respective Mobile Node. The receipt of a binding request message by a Mobile Node results in the transmission of a new binding update to the source of that binding request. This mechanism enables Correspondent Nodes to avoid short periods of sub-optimal routing, due to the expiry of an accurate binding.

2.2.4.4 Binding Update List

As a Mobile Node roams from network to network, it is essential that binding update messages are transmitted to that node's Home Agent and Correspondent Nodes as soon as possible, in order to facilitate a fast handoff. Mobile Nodes therefore cannot rely on the soft state timeout mechanism used in binding caches to refresh stale bindings maintained by Correspondent Nodes (typical binding lifetimes are of the order of minutes). An additional data structure, the *binding update list*, is therefore kept by Mobile Nodes, which maintains state on any Correspondent Nodes or Home Agents. Figure 3 illustrates the binding update list.

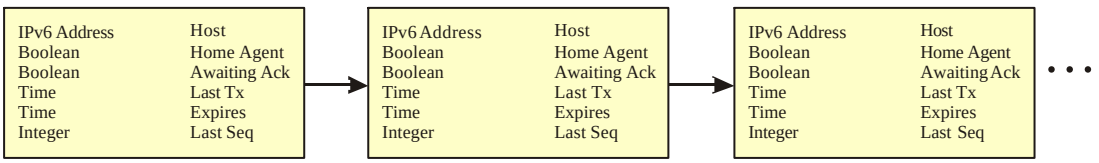

Figure 3 Mobile IPv6 Binding Update List

The binding update list contains one entry for every Correspondent Node or Home Agent to which a binding update has been sent. List entries contain information such as the address and time at which the binding update was transmitted, the state of any unacknowledged updates, the lifetime of the binding, a Home Agent flag, and the sequence number of the last transmission. Binding list entries are garbage collected from the binding update list as the respective binding expires.

The maintenance of the binding update list allows for significantly faster handoff performance. After a handoff has been detected and autoconfiguration has been completed, the binding update list is traversed, and a binding update message transmitted to every node contained within the list, thereby updating the binding caches of any active Correspondent Nodes.

2.2.5 Proxy Neighbour Discovery

Since packets destined for a Mobile Node may be incorrectly routed to its home network, placing Home Agents within an IPv6 edge router would allow the efficient interception of these packets, as they would likely travel through that router. However, the assumption that the packets will automatically reach this edge router cannot be relied upon. For example, consider the case of a Correspondent Node located on a Mobile Node's home network. If the Correspondent Node were to send a packet to that Mobile Node, its routing table would dictate that the Mobile Node was directly accessible, and did not require forwarding by a router. In this case, the Home Agent would not be able to intercept the packet.

Mobile IPv6 addresses this issue through a technique call *proxy neighbour discovery* (proxy ND). Neighbour Discovery [20] is a standard IPv6 protocol for the discovery of MAC addresses from IPv6 addresses, similar in concept to the ARP protocol for IPv4. Proxy ND involves an IPv6 node masquerading as another node at the MAC layer, by falsely responding to neighbour solicitations with its own MAC address. Home agents use proxy ND to ensure they intercept any IPv6 packets for a Mobile Node transmitted on its home network. To accomplish this, Home Agents also maintain a *proxy neighbour discovery table*, which contains the IPv6 addresses to which the Home Agent is acting as a proxy for. Entries to this table are added and removed as binding update messages with the 'Home Agent' flag set are added and removed from the binding cache.

2.2.6 Home Address Option

Mobile Nodes away from home have a choice of which global scope IPv6 address to use as a source for outgoing IPv6 packets. Either the node's home address could be used, or the current care-of address. However, neither of these choices are particularly desirable. If the current care-of address is used, then the source address for subsequent packets will change as a handoff takes place. This causes often irreparable problems for higher layer protocols such as TCP, which maintain transport layer identifiers and checksums based on network layer addresses. On the other hand, if the home address is used, then the outgoing IPv6 packet becomes susceptible to ingress filtering.

Ingress filtering is performed by many border routers to improve the security of the site to which they serve. Ingress filtering involves the inspection of the source address of all incoming IP packets, and verifying that the route to that address lies along the interface on which the packet was received. Any packets which fail this test are dropped as a security precaution. This can avoid many security attacks which use 'address spoofing'. Mobile Nodes sourcing their IPv6 packets with their home address on a foreign network can be mistakenly interpreted as a security threat by routers employing ingress filtering.

Mobile IPv6 defines a new IPv6 destination option, known as the *home address option*, which can provide a source address solution that is safe for transport protocols and is not susceptible to ingress filtering. This is achieved by a route optimised form of reverse tunnelling, which involves a level of minimal encapsulation when sending IPv6 packets from a Mobile Node. Table 4 illustrates the home address option.

IPv6 Header		Home Address Option	Transport Header	Payload
Source Address:	Care of Address	Home Address	TCP/UDP	Data
Dest. Address:	Correspon dent Node			
◄——— 40 bytes ———►		◄——— 18 bytes ———►		

Table 4 Mobile IPv6 Home Address Option

The Mobile IPv6 specification states that Mobile Nodes should source their IPv6 packets using a care-of address, thereby avoiding ingress filtering. However, any upper layer protocols should assume the source address of outgoing packets is the home address. All outgoing packets from a Mobile Node include a home address option. Upon receipt by a Correspondent Node, the address contained within the home address option replaces the source address of the packet, before any upper layer processing takes place.

2.2.7 Home Agent Discovery

Mobile IPv6 provides a mechanism for Mobile Nodes to automatically detect the presence of Home Agents on its home network. This mechanism involves all Home Agents joining the link local Home Agents anycast address. Mobile Nodes wishing to discover a Home Agent sends a binding update (with the 'H' flag set) to this anycast address. This message will be delivered to at most one Home Agent on the home network. Upon receipt of the message, the Home Agent responds with a binding acknowledgement, thereby informing the Mobile Node of the Home Agent's IPv6 address. The binding updates sent to the Home Agents anycast address are otherwise ignored by Home Agents.

2.3 Recent Changes to MIPv6

The latest version of the draft MIPv6 standard [13] includes some significant changes to the previous version of MIPv6. Although there are no MIPv6 implementations yet that support this draft version, the major changes are described here for the sake of completeness.

2.3.1 The Mobility Header

A new IPv6 protocol, the mobility header, has replaced the IPv6 Destination Options for some of the MIPv6 signalling. The motivation for this change was to allow the use of IPsec for:

- protecting RR packets (see below) as they are forwarded to the Mobile Node from the Home Agent
- protecting binding updates between the Mobile Node and the Home Agent

- possible future protection of binding updates between the Mobile Node and the Correspondent Node.

The mobility header is used by Mobile Nodes, Home Agents and Correspondent Nodes for all messaging related binding creation and management. The format of the mobility header is illustrated in Figure 4.

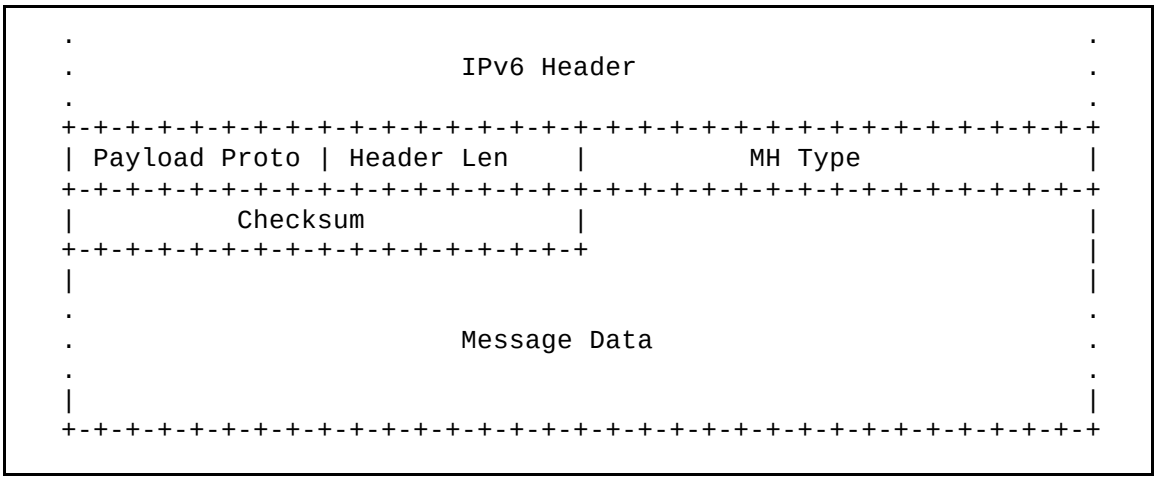

Figure 4 The Mobility Header Format

The mobility header is identified by a next header value of 62 in the IPv6 header (or an alternative preceding header if there is one). The 'MH Type' field identifies the specific mobility message in question and can be one of the following:

- Home Test Init (HoTI)
- Care-of Test Init (CoTI)
- Home Test (HoT)
- Care-of Test (CoT)
- Binding Request (BR)
- Binding Update (BU)
- Binding Acknowledgement (BA)
- Binding Missing (BM).

Due to the use of the new mobility header, the piggybacking of MIPv6 signalling with data is no longer possible. However, a separate extension to the protocol that will allow piggybacking in the presence of the mobility header may become available in the near future.

2.3.2 The Return Routability Method

The Return Routability (RR) method is a new binding update (BU) authorisation mechanism, suitable for use between Mobile Node and Correspondent Node peers, which have no previous knowledge of each other. It is based on the principle of exchanging 'cookies' that verify the Mobile Node is 'alive' at its claimed address. The cookies are used by the Mobile Node to cryptographically protect the eventual BU message.

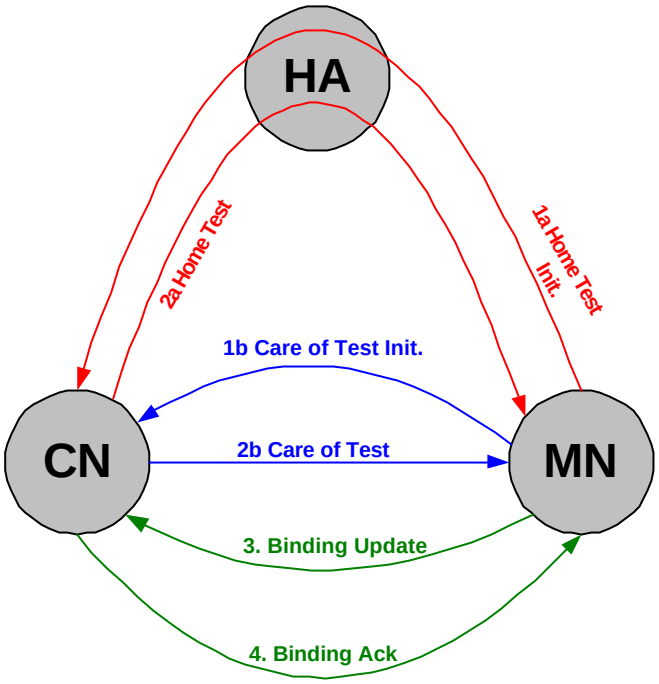

Figure 5 Return Routability Messaging

When a Mobile Node wishes to achieve route optimisation, it initiates the RR method as illustrated in Figure 5.

The HoTI and CoTI messages are sent simultaneously by the Mobile Node to the Correspondent Node.Upon the receipt of the HoTI and CoTI messages, the Correspondent Node computes two cookies based on the information contained in the messages, combined with its own secret key and nonce value. These cookies are inserted into the respective HoT and CoT messages, which are then sent back simultaneously to the Mobile Node.

Once the Mobile Node has received both the HoT and CoT messages, it has the cookies necessary to send the BU to the Correspondent Node. It hashes together the cookies to form a session key, which is then used to authenticate the BU that is sent to the Correspondent Node. When the Correspondent Node receives the BU, it can verify the information using its cookies and create a binding cache entry for the Mobile Node. The Correspondent Node may optionally acknowledge the BU with a BA.

2.3.3 Other Changes

Other changes brought by draft version 16 of the MIPv6 standard include:

- The Home Address Option (HAO) can now only be used when a binding already exists.
- A new message, Binding Missing (BM), has been added for the Correspondent Node to signal the Mobile Node that it has used the HAO but the Correspondent Node has no associated binding.
- MIPv6 now uses a Routing Header of type 2, instead of the general type 0. This will enable firewall administrators to allow Routing Headers specifically for MIPv6.

3 Survey of Existing Mobile IPv6 Implementations

In this section, a high-level survey of existing Mobile IPv6 implementations is provided. The survey includes the following implementations:

1. KAME
2. Monarch
3. NEC
4. 6WINDGate
5. MIPL
6. NUS
7. ULANC MIPv6
8. Bull
9. MSR
10. Cisco
11. Nokia
12. Ericsson
13. SFC-MIP6.

Some of the information provided in this section was acquired from the WWW, hence it is possible that it is now out of date.

3.1 KAME

The KAME (KAME means "turtle" in Japanese) project [1] is a joint effort of several companies from Japan to provide a free IPv6 and IPsec stack for BSD variant platforms. KAME is a part of a broader project, the WIDE project, which deals with the establishment of a "large-scale distributed computing environment"; KAME, as a project is more implementation oriented.

The KAME project is in close co-operation with the TAHI project, which develops conformance and interoperability tests concerning IPv6 implementations. KAME participated in the 3rd TAHI interoperability test event. In addition the KAME project co-operates with the USAGI project, which aims to improve the IPv6 environment on Linux.

The KAME MIPv6 implementation is based on the respective implementations of Ericsson, NEC and Keio University. It is based on the IETF specification draft version 15 [12].

3.2 Monarch

The Monarch (this name was derived from the migratory behaviour of the monarch butterfly) project [2] of the Department of Computer Science at Rice University focuses on developing open source networking protocols and protocol interfaces for wireless and mobile host networking. Most of the work through this project is in the area of ad-hoc networks, in which a Mobile Node can forward packets functioning as a router. The Monarch project has contributed to the IETF activities by publishing an RFC and several drafts concerning IPv6 and Mobile Ad Hoc networks.

The available Mobile IPv6 implementation of the Monarch project is based on September 3, 1997 release of INRIA's IPv6 code and supports FreeBSD.

The Monarch implementation is based on the IETF specification draft-ietf-mobileip-ipv6-03.txt.

3.3 NEC

The NEC Mobile IPv6 implementation [3] is an open source implementation that is based on the Mobile IPv6 implementation of the KAME project. Indeed, it is provided as a patch to the KAME implementation.

The NEC Mobile IPv6 implementation participated in the 3rd TAHI interoperability test and it took the TAHI MIPv6 conformance test.

The implementation is based on the IETF specification draft-ietf-mobileip-ipv6-13.txt.

3.4 6WINDGate

The 6WINDGate Mobile IPv6 implementation of 6WIND [4] implements an IPv6 Home Agent function. It participated in the 3rd TAHI interoperability test event.

The implementation is based on the IETF specification draft-ietf-mobileip-ipv6-13.txt.

3.5 MIPL

MIPL (Mobile IPv6 for Linux) is an implementation [5] that was originally developed as a software project course in the Helsinki University of Technology (HUT), with the goal to create a prototype implementation of Mobile IPv6 for Linux. After the course, the implementation was further developed in the context of the GO/Core project at HUT Telecommunications and Multimedia Lab. It is an open source implementation and has been released under GNU GPL.

The MIPL implementation has been tested in interoperability and conformance testing events such as the ETSI IPv6 Plugtest (November 19th - 23rd, 2001) and Connectathon 2002 (February 28th - March 7th, 2002)

The implementation is based on the IETF draft specification version 15 [12].

3.6 NUS

The “Mobile IP at NUS” project [6] is an open source project funded by National University of Singapore (NUS). It is part of the initiatives “Open Source at NUS” and “Networking Research at NUS”. The project focuses on the performance analysis of the Mobile IP architecture so as to characterise its efficiency and uncover any performance bottlenecks.

The project has implemented a Linux Mobile IPv6 implementation that is based on the IETF specification draft-ietf-mobileip-ipv6-04.txt.

3.7 Lancaster University’s MIPv6 for Linux

Lancaster University’s MIPv6 stack for Linux was first released to the public on 3rd April 1998. We believe this was the first publicly demonstrated implementation of Mobile IPv6, and also the first

version complying with the Mobile IPv6 specification draft-ietf-mobileip-ipv6-05.txt be released. The implementation has continued to be updated as the MIPv6 draft standard has progressed.

The current version of the MIPv6 code is based on IETF draft specification version 13 and is available from http://www.cs-ipv6.lancs.ac.uk/ipv6/MobileIP/. It is provided as a kernel installable module and works with Linux kernels 2.4.16 and 2.4.17. The implementation can support Mobile Node, Home Agent (mobile-aware router), and Correspondent Node functionality.

3.8 Bull

Bull continues developing IPv6 by experimenting MobileIPv6 with IPsecv6 in a common project with INRIA and France Telecom.

3.9 Microsoft Research MIPv6

The Microsoft Research (MSR) Mobile IPv6 implementation was produced in collaboration with Lancaster University as part of the LandMARC project [15]. During the project, Lancaster University's MIPv6 implementation for Linux (see section 4.6) was ported to the Windows 2000 operating system. The implementation is available in executable and source code format as a free download for research purposes from http://research.microsoft.com/downloads/. The MIPv6 implementation is a modified version of the MSR IPv6 stack, version 1.4 [21].

At the time of writing, the public release of the MIPv6 implementation supports version 12 of the IETF MIPv6 draft and provides Mobile Node, Correspondent Node and Home Agent functionality. Microsoft is currently working on supporting version 15 of the IETF MIPv6 draft [12] and also plan to integrate MIPv6 functionality into their IPv6 stack as standard in the near future.

At the time of writing, there is no public release of a MIPv6 implementation for other MS Windows operating systems. However, MIPv6 functionality in Windows XP and CE may be obtained from the Windows 2000 MIPv6 source code by re-compiling with appropriate changes made.

3.10 Cisco

The Cisco MIPv6 implementation began with a research collaboration with Lancaster University and the porting of their MIPv6 implementation for Linux. MIPv6 support is currently available as a Cisco "technology release". It is not yet fully supported but is available for testing via the Cisco IPv6 support team. The major functionality supported by the release is that of the Home Agent and the Correspondent Node.

The Cisco IOS release of the protocol has been demonstrated most recently at the Madrid IPv6 Summit, March 2002. The configuration comprised a Cisco 2600 acting as Home Agent, the Mobile Node was a Compaq iPAQ running Linux, and the Correspondent Node was an AlphaServer running Tru64.

At the time of writing, the Cisco IOS release supports version 13 of the IETF draft specification. The release works on any Cisco router that supports the 12.2.T software. For further information about this, see the Cisco IPv6 Statement of Direction [14].

3.11 Nokia

The Nokia Mobile IPv6 implementation [7] is based on the Symbian EPOC operating system, an open standards operating system for data enabled mobile phones, which takes into consideration the reduced processing power and the memory shortage on portable devices. Symbian [8] is a software licensing company, owned by Ericsson, Nokia, Matsushita (Panasonic), Motorola, Psion and Sony. It functions as the trusted supplier of the Symbian OS. The world's first Symbian OS phone became available in the first half of 2001 and it was the Nokia 9210 Communicator.

Nokia Research Center/Mobile Networks Laboratory has implemented a hybrid IPv4/IPv6 stack (host implementation). The protocol stack runs on various models of EPOC based PDAs e.g., on Psion Series 5mx, Series 7, netBook, and Revo.

According to a Nokia press release, Nokia was the first company to demonstrate IP multimedia Mobile IPv6 devices.

3.12 Ericsson

Ericsson's Mobile IPv6 implementation [9] is based on the Symbian OS. Symbian OS is already available in the Ericsson R380, R380e and R380 World Smartphones. It was also announced by Symbian that the Sony Ericsson P800 smartphone is the first multimedia smartphone to use Symbian OS v7.0 and the UIQ pen-based user interface, that will be shipped in the third quarter of 2002.

The Ericsson Mobile IPv6 implementation participated in the 3rd TAHI interoperability test event. It is based on the IETF specification draft-ietf-mobileip-ipv6-13.txt.

3.13 SFC-MIP6 by Keio University

SFC-MIP6 is an open Mobile IPv6 implementation [10] by Keio University based on the FreeBSD platform. They took part in an interoperability test between KAME, USAGI, and SFC. They also participated in the 3rd TAHI interoperability test event

The current implementation is based on the IETF specification draft-ietf-mobileip-ipv6-15.txt [13].

4 Evaluation of Mobile IPv6 Implementations

This section provides a high-level evaluation of various Mobile IPv6 implementations. It is not intended to be an in-depth, low-level technical evaluation. Rather, it provides an overview of a number of MIPv6 implementations, with which consortium partners have some knowledge of or experience with, and an evaluation of the key features that are supported by each implementation. As such, performance measurements are out of scope for this document. Not all of the implementations listed in section 3 are evaluated here due to the lack of suitable experience by the consortium partners. Time constraints have precluded the testing of previously untried implementations at this early stage of the 6NET project. In addition, it should be stated that because MIPv6 is a moving target, new releases of implementations may obsolete the versions described here soon after the time of writing.

4.1 Evaluation Criteria

The implementations are evaluated according to what features we considered a MIPv6 implementation (based on a relatively recent draft version such as 12 or higher) should support. The following criteria were identified:

- Proxy Neighbour Discovery
- IPv6 Encapsulation & Decapsulation
- Dynamic Home Agent Address Discovery
- Binding Management
- Home Address Option
- Movement Detection
- Smooth Handoff
- IPsec / AAA
- Key exchange
- Support for notebooks / PDAs
- MIP built-in
- Number of software patches necessary
- Set-up capabilities.

4.2 MIPv6 for Linux (MIPL)

4.2.1 MIPL overview

The MIPL implementation of the University of Helsinki [5] is currently available in version 0.9.1 for Linux. The officially supported kernel version is 2.4.16; however kernel 2.4.17 also seems to work without problems. The implementation is based on the draft-ietf-mobileip-ipv6-15.txt [12].

One of the major improvements compared to v0.9 is the support of preferences for multiple interfaces, which allows the user to manipulate the selection of the default router by specifying a preferred interface. This forces a vertical handover if a node is multihomed (for example having a LAN and WaveLAN connection).

4.2.2 Installation

The installation of MIPL 0.9.1 requires patching the Linux kernel (1 cumulative patch). The patch modifies the existing IPv6 code and adds a new subdirectory with the MIPv6 specific code, which is compiled into a separate kernel module. In addition to the kernel module, a userspace tool (mipdiag) is provided for configuration, for example setting the role of the node (Home Agent, Correspondent Node or Mobile Node) and displaying information.

The installation procedure on a SuSE Linux 7.3 distribution needed some minor fixes to run through without errors, among others the startup script causes problems because of a bug in the parameter evaluation of the mipdiag tool. Furthermore, the compilation of the mipdiag tool requires a header file to be copied from the kernel include directory to the user space include directory. However, this might be a distribution specific problem with the SuSE Linux distribution.

An additional change from MIPL version 0.9 is the introduction of an additional module, ipv6_tunnel, that provides the IPv6-in-IPv6 tunnelling code. This additional module is not loaded automatically after the initial installation, which prevents the mobile_ip6 module from being loaded as well. This problem can be fixed by loading the modules manually and running the Linux "depmod" command to resolve and save the module dependencies.

4.2.3 Features

This section evaluates the MIPL implementation against the criteria listed in 4.1.

4.2.3.1 Proxy Neighbour Discovery

The MIPL Home Agent supports Proxy Neighbour Discovery.

4.2.3.2 IPv6 Encapsulation & Decapsulation

IPv6-in-IPv6 encapsulation is required when the Home Agent has to forward packets secured with IPsec to the Mobile Node in a foreign network. In this case, it cannot modify the routing header of the received packet without invalidating the IPsec (or Authentication) header.

MIPL v0.9.1 supports IPv6-in-IPv6 tunnelling.

4.2.3.3 Dynamic Home Agent Address Discovery

According to the README of the installation, DHAAD is supported, but the Linux IPv6 stack currently has problems handling the anycast messages, but with "some tricks" it is still possible to enable this feature. This has not been verified in our testbed.

4.2.3.4 Binding Management

The "mipdiag" tool supports the viewing of existing Bindings.

4.2.3.5 Home Address Option

The generation of a Home Address option in packets sent to the Home Agent and Correspondent Nodes is supported by the implementation. The MIPL software has to be installed on every node that communicates with a Mobile Node since the Home Address option is not processed in the standard IPv6 stack and the packet would be dropped.

4.2.3.6 Movement Detection

Movement Detection in MIPv6 is based on the prefix information received in Router Advertisements. Additional movement detection mechanisms (that are not required by the draft) like link layer triggers are not implemented.

4.2.3.7 Smooth Handoff

The draft states that "to assist with smooth handovers, a Mobile Node SHOULD retain its previous primary care-of address as a (non-primary) care-of address, and SHOULD still accept packets at this address, even after registering its new primary care-of address with its Home Agent" [12]. This allows for (nearly) smooth handoffs (no packet loss), if the Mobile Node has 2 network interfaces, for example an Ethernet and a Wavelan interface. A simple test of a handover from Wavelan to the Ethernet interface after restoring the network link shows that the packet loss is negligible when streaming audio data, for example. The switch is not noticeable with a medium quality audio signal.

4.2.3.8 IPsec / AAA

Authentication via IPsec is supported with a "manually keyed" authentication header. The authentication data suboption is also supported.

4.2.3.9 Key exchange

Authentication keys can be either MD5 or SHA-1. The authentication algorithm to be used is specified in one of the configuration files.

4.2.3.10 Support for notebooks / PDAs

MIPL supports multiple interfaces and, from version 0.9.1 on, the interface priority can be configured with a userspace tool. MIPL has been installed on several Sony Vaio (sub-)notebooks.

4.2.3.11 MIP built-in

MIPv6 is not built into the current Linux 2.4.x kernels.

4.3 Cisco IOS MIPv6

4.3.1 Overview

Mobile IPv6 support is currently available as a Cisco "Technology Preview". Whilst not yet fully supported, this software is available for testing, via the Cisco IPv6 support team. The major functionality supported by the release is that of the Home Agent and the Correspondent Node.

The Cisco IOS release of the protocol has been demonstrated most recently at the Madrid IPv6 Summit, March 2002. The configuration comprised a Cisco 2600 acting as Home Agent, the Mobile Node was a Compaq iPAQ running Linux, and the Correspondent Node was an AlphaServer running Tru64.

Currently, the version supported is draft version 13. The Cisco IPv6 development team plans to produce a Technology Preview version of draft 17 of the Mobile IPv6 specification when it becomes available. These notes refer to the currently available Technology Preview based on draft 13.

The release works on any Cisco router that supports the 12.2.T software. For further information about this, see the Cisco IPv6 Statement of Direction [14].

4.3.2 Features

4.3.2.1 Proxy Neighbour Discovery

Fully supported.

4.3.2.2 IPv6 Encapsulation & Decapsulation

This includes IPv6 in IPv6 encapsulation.

4.3.2.3 Dynamic Home Agent Address Discovery

Included in the implementation is the functionality for Home Agents to keep lists of other Home Agents available on the same sub-network. However, at the time at which the development was carried out, IPv6 Anycast support (fundamental to the concept of Home Agent discovery) was unavailable. Thus, dynamic DHAAD is not supported.

4.3.2.4 Binding Management

The implementation supports a significant amount of binding management. On a general basis, the functionality is available to view bindings, as well as adding and deleting entries to the binding tables. There is also significant functionality in terms of determining the rights of nodes to roam to a foreign network, based on the Mobile Node's home address. For example, the Home Agent is able to decide which Mobile Nodes it will accept bindings for, and which networks the Mobile Node is able to roam to. This is achieved through the standard IOS access control list implementation for IPv6.

4.3.2.5 Home Address Option

The Home Address Option is correctly handled by this implementation of the protocol.

4.3.2.6 Movement Detection

This relates more to the Mobile Node than the Home Agent or Correspondent Node. However, in terms of more general IPv6 functionality, the way that Router Advertisements are issued has been altered in order to allow faster detection, i.e. Router Advertisements are allowed to be transmitted at a faster rate than in the Neighbour Discovery RFC.

4.3.2.7 Smooth Handoff

Home Agent functionality is suitable for supporting smooth handoff.

4.3.2.8 IPsec / AAA

Not supported, but is one of the areas that are being altered as the draft revision process takes place.

4.3.2.9 Key Exchange

Not supported, but is another one of the areas that are being altered as the draft revision process takes place.

4.3.2.10 Support for Notebooks / PDAs

Whilst this point probably refers more to the client-side implementation, it can be said that the "client type" is completely transparent to the Cisco IOS MIPv6 implementation. However, the lack of backwards compatibility between recent drafts means that Mobile Nodes implementing a draft later than draft 13 may not interoperate with this Home Agent and Correspondent Node implementation.

4.3.2.11 MIP Built-in

The implementation is an integral part of the particular IOS release.

4.3.2.12 Number of Patches Necessary

None, but as detailed above, it is only currently available as a technology preview release.

4.3.2.13 Set-up Capabilities

Standard command line commands are used to configure the release. The relevant configuration commands, along with an example configuration, are provided in Appendix A.

4.3.3 Further Information

The current implementation has gone through several phases of internal manual testing, and is available to third parties through the Technology Preview / Cisco EFT scheme.

4.3.4 Future Developments

As stated earlier, the Cisco IPv6 development team plans to produce a Technology Preview version of draft 17 of the Mobile IPv6 Specification when it becomes available.

4.4 Microsoft Windows Support for MIPv6

4.4.1 Overview

Microsoft has a working implementation of Mobile IPv6 for Windows 2000 that was produced in collaboration with researchers at Lancaster University as part of the LandMARC project [15]. During the project, Lancaster University's MIPv6 implementation for Linux (see section 4.6) was ported to the Windows 2000 operating system. The implementation is available in executable and source code format as a free download for research purposes (subject to licence terms) from http://research.microsoft.com/downloads/. The MIPv6 implementation is based on the MSR IPv6 stack, version 1.4 [21], which has had MIPv6 functionality added to it. At the time of writing, the public release of the MIPv6 implementation supports version 12 of the IETF MIPv6 draft and provides Mobile Node, Correspondent Node and Home Agent functionality. Microsoft is currently working on supporting version 15 of the IETF MIPv6 draft and plans to integrate MIPv6 functionality into their IPv6 stack as standard in the near future.

There is no public release of a MIPv6 implementation for other MS Windows operating systems. However, MIPv6 functionality in Windows XP and CE may be obtained from the Windows 2000 MIPv6 source code by re-compiling with appropriate changes made.

4.4.2 Installation and Configuration

Before one can install MIPv6 functionality on Windows 2000, one must install and configure version 1.4 of the MSR IPv6 stack so that the target machine has a working IPv6 stack. Once this is achieved, the MIPv6 binary distribution needs to be added to the relevant network connections in Start->Settings->NetworkAndDialupConnections->LocalAreaConnection.

The Mobile IPv6 stack can be dynamically configured to run in any combination of modes. These modes include Mobile mode, Correspondent mode, and Home Agent mode. When in Mobile mode, home addresses can be dynamically added and removed and the security settings for Home Agents can be configured. When any change takes place to the configuration, the new settings are stored in the Windows registry, where they are subsequently reloaded during driver initialisation.

In simple cases the MIPv6 Configuration service should succeed in automatically configuring the stack for Mobile Nodes, based on routers that advertise MIPv6 Home Agent service in their Router Advertisement messages. The operation of the MIPv6 Configuration Service can be controlled using the MIPv6Conf.exe utility. This utility can also be used to inspect and change the mobility parameters of the MIPv6 stack in a straightforward manner.

Beyond that the MIPv6 code is configured using a new version of the ipv6.exe utility. By default the MIPv6 stack assumes Mobile and Correspondent mode. Additional arguments to the new version of ipv6.exe are:

ipv6.exe hau *h-addr n ha-addr* Define home address *h-addr* with prefix length *n* using the Home Agent on address *ha-addr*

ipv6.exe hau *h-addr n ::0* Delete home address *h-addr*

ipv6.exe bc Inspect the state of the MIPv6 Binding Cache

ipv6.exe bc Inspect the state of the MIPv6 Binding Update List

ipv6.exe mip Inspect the mode (Mobile, Correspondent, Home Agent) of the MIPv6 stack

ipv6.exe mipu [MN] [CN] [HA] Set the mode of the MIPv6 stack to one or more of Mobile Node (MN), Correspondent Node (CN) or Home Agent (HA).

In addition, one can control the MIPv6 Configuration Service using the MIPv6Conf.exe utility. This provides a GUI interface for inspecting and manually configuring home addresses. Furthermore, MIPv6Conf.exe allows one to specify parameters dictating the automatic configuration of home addresses by the MIPv6 configuration service, such as the maximum number of home addresses that should be configured.

The stack will remember mobility parameters, in particular home addresses, in the registry under key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip6\Mobility. These setting survive reloads of the stack (via reboot or "net [stop|start] tcpip6") but are lost if the stack is reinstalled.

IPsec functionality requires manual configuration of its Security Policy Database (SPD), Security Association Database (SAD) and manual key distribution. Since this can be an extremely error-prone activity, the sagen.exe command line utility is provided for generating the relevant IPsec SPD, SAD and key files for a known set of MIPv6 hosts.

Sagen.exe sets up the following security policy:

- On End Systems: ICMP packets do not require authentication, but all other packets must be authenticated
- On Routers: Accept authentication, but don't require it (Routers must be able to forward transit traffic. They must also be able to forward packets destined for one of their interfaces that are received on a different interface.)

The algorithm used in this program does not scale to large networks. A key management protocol such as IKE/ISAKMP should be used instead of this program in production systems. Hence, this program should only be used in test environments. In addition, this program sets IPsec cryptographic keys to predictable values. Thus, it provides no actual security and again, should only be used for test purposes.

4.4.2.1 Microsoft Mobile IPv6 Configuration Tool(MIPv6Conf.exe)

The dialog-based WIN32 application enables users to conveniently configure the Microsoft Mobile IPv6 stack. Its primary purpose is to provide a friendly user-interface to common configuration settings and the new auto-configuration service.

The auto-configuration service enables Mobile Nodes to auto-configure home addresses as they start-up out-of-the-box or enter a (new) Mobile IPv6 network. A more detailed description of this service is provided below.

The *General* property sheet allows users to define the basic configuration of the Mobile Node (e.g., Correspondent Node, Mobile Node, Home Agent) and whether or not auto-configuration is used. As default settings (after installation), a node is configured to be a Correspondent and Mobile Node, and auto-configuration is enabled. If auto-configuration is enabled, advanced settings become available - here the user can define:

- if the first potential home address should be assigned as soon as a Home Agent is discovered (desperation mode)
- whether or not the user should be informed when a new home address becomes available

- the maximum number of auto-configured home addresses
- the heuristic which governs the responsiveness of the configuration service (how quickly to adopt a home address after a new address becomes available - aggressive vs. conservative).

Note that enabling and disabling auto-configuration support actually starts or terminates the auto-configuration service.

The *Home Address* property sheet allows users to define the preferred home address and to manually add or remove home addresses. Manual configuration of a new home address is simplified through the auto-configuration support, which allows users to pick a Home Agent from a list of *Known Home Agents* maintained by the auto-configuration service. The state is held in the registry. This list can also be used to define *default* or recommended Home Agents by system administrators or OEMs.

Furthermore, manual home address configuration is facilitated due to the proposition of likely home addresses as soon as the Home Agent is determined. The configuration tool suggests the stateless home address and all other currently configured link-local addresses for a given Home Agent network.

4.4.2.2 Microsoft Mobile IPv6 Auto-Configuration Service

This WIN32 service program, running as a background service, "listens" for Home Agent advertisements on the network and auto-configures home addresses if desired.

This is achieved through event notification from the underlying Mobile IPv6 stack. As soon as the Mobile IPv6 stack receives a router advertisement with the Home Agent flag set, it notifies the auto-configuration service, which then checks whether or not a new home address is desired. If the number of active home addresses[1] is less than the maximum number of home addresses defined by the user, a new home address is configured (or simply proposed to the user when user-interaction is desired). Otherwise, the service simply updates the list of *Known Home Agents* held in the registry.

Each entry holds the number of router advertisements received from this Home Agent, the time stamp of the last router advertisement and the *rank* of the Home Agent. The rank is determined based on an adaptive control function, which takes the number of router advertisements, the last time stamp, and the previous rank as input parameters.

Note that auto-configured home addresses are built from the 64-bit prefix of the Home Agent network and the 64-bit EUI of the Mobile Node.

4.4.3 Features

The major features supported by the Microsoft MIPv6 implementation are as follows.

4.4.3.1 Proxy Neighbour Discovery

Proxy Neighbour Discovery is supported in the Microsoft implementation. In order to provide the IPv6 address proxy functionality required by Home Agents, a new address type, the Proxy Address Entry (or PAE) has been defined[2]. When a Mobile Node requests a home registration for a specified

[1] Home addresses for which the binding updates have been positively acknowledged by the Home Agent within a certain time frame.
[2] The PAE is specific to the Microsoft MIPv6 implementation and does not affect the IETF draft standard.

home address, a PAE corresponding to that home address is added to the relevant interface on the Home Agent, and the relevant IPv6 Neighbour Discovery multicast groups are joined. PAEs are removed from interfaces upon the removal of the corresponding entry from the bindings cache (either explicitly via a registration, or indirectly via a cache expiry).

Upon reception of a packet destined for a PAE, the Home Agent code tunnels this packet to the Mobile Node's care-of address, as specified in the bindings cache.

4.4.3.2 IPv6 Encapsulation & Decapsulation

In the MIPv6 distribution, minor modifications were made to the MSR IPv6 stack, in order to allow the reception and generation of IPv6 in IPv6 tunnelled packets (as generated by Mobile IPv6 Home Agents). Upon reception of such packets by the Mobile Node, a new packet flag PACKET_GENERATE_BU is set in the relevant packet. Once fully parsed, this will stimulate a binding update message to be generated to the source of the packet.

4.4.3.3 Dynamic Home Agent Address Discovery

The MSR implementation does not implement DHAAD, However, it maintains a list of (32 maximum) recently seen Home Agents so that a service can be written to check this list when a new home address is required and one is not available on the local link.

4.4.3.4 Binding Management

The MSR MIPv6 stack maintains a binding cache and binding update list for the purpose of binding management. The generation and reception/handling of binding requests, binding updates and binding (n)acks are all implemented.

When in Home Agent mode, the MIPv6 stack will respond to binding update messages with the 'H' bit set, signifying home registrations. Upon reception of home registrations, the binding update is validated to guarantee that this node is capable of acting as a Home Agent for the Mobile Node originating the binding update. This is achieved by ensuring that the node has at least one unicast IPv6 address bound to an interface that corresponds to the same IPv6 network as the home address supplied in the binding update message.

The Home Agent functionality re-uses the 'mobile security' flag when parsing binding updates. A home binding with no authentication will be rejected if the mobile security flag is set, and accepted otherwise. Currently, no access control for Mobile Nodes is supported other than through IPsec, nor is a virtual interface for the provision of 'homeless' Mobile Nodes.

4.4.3.5 Home Address Option

The MSR MIPv6 stack deals with both sending and receiving a home address option, and has support for the Unique ID, Home Agents List and Care-of Address mobile sub-options. Any unrecognised sub-options are skipped silently. Other than the home address option, the implementation never sends any mobile sub-options.

4.4.3.6 Movement Detection

Movement detection is based on notification of NDIS media connect or disconnect and changes in the perception of the local link (e.g. new router advertisements, timeout of current router

advertisements). However, tests have shown that media sensing under XP is not reliable, hence movement detection by perception of the local link remains the more reliable method of movement detection.

Movement detection is achieved by listening to IPv6 router advertisement messages. If a router advert is received which does not match the previous one, then a handoff is deemed to have occurred. At this point, any IPv6 addresses bound to the interface which has performed the handoff are removed, and the associated routing state flushed.

Once a new address has been acquired (e.g. via stateless address configuration) and any necessary duplicate address detection has taken place, then the binding update lists are traversed and binding update messages sent to all relevant Correspondent Nodes.

4.4.3.7 Smooth Handoff

Handoffs are not usually attempted unless an old route becomes unreachable (e.g. due to media disconnect), or unless the routing policy/preference changes. In such a case, packets are delayed whilst a new route is found. If that route is found within a reasonable time (configurable, but based upon the maximum neighbour/router solicitation response times by default), packets are sent via that route without the application being aware of the handoff (unless it has requested handoff notifications via middleware, but this is not part of the stack Microsoft are using in their products). Smooth and transparent handoffs have been publicly demonstrated on Jameson (CE .NET 4.10) using Internet Explorer, and on XP using Internet Explorer, IIS, and at a Microsoft-internal event pre-release versions of Exchange and Outlook. TCP connections don't break immediately if no routes are available by design to give the user a chance to plug a cable back in.

4.4.3.8 IPsec / AAA

The MIPv6 stack has IPsec support, including integration with existing MSR IPv6 IPsec functionality for protection against misuse of MIPv6 control messages.

The authentication of binding updates is based on upon a choice between IPsec and CAM (Child-proof Authentication for Mobile IPv6) [17].

4.4.3.9 Key exchange

The command line utility sagen.exe (see section 4.4.2) is provided as a means to facilitate key distribution for IPsec.

4.4.3.10 Support for notebooks / PDAs

The MSR MIPv6 implementation (in Mobile Node and Correspondent Node modes) has been tested successfully on Sony Vaio and Dell notebooks running Windows 2000. It has also been tested successfully on a Compaq iPaq, and Pocket PCs (e.g. HP Jornado) running Windows CE version 3.

4.4.3.11 MIP built-in

MIPv6 support does not come built-in with any MS Windows operating system.

4.4.3.12 Number of software patches necessary

MIPv6 relies on the existence of a working IPv6 stack on the Microsoft operating system. In Windows 2000, the MSR IPv6 stack has to be downloaded and installed separately resulting in two software patches (including the patch for MIPv6 itself). In Windows XP, IPv6 is already available but needs to be activated by issuing an 'ipv6 install' command on the command line. However, the MIPv6 stack still needs to be patched on top of this.

4.4.3.13 Set-up capabilities

The MSR MIPv6 implementation provides an automatic configuration service in addition to a graphical configuration utility to enable the configuration of the MIPv6 stack (see above).

4.4.3.14 Other Features

Other features supported by the MSR MIPv6 implementation include:

- support for multiple home addresses, optionally involving multiple Home Agents on multiple home networks, per Mobile Node
- interface reconfiguration
- dynamic destination option packet generation
- route cache entry generation
- applications (ping6.exe, TCP/UDP)
- use of registry to store persistent home address information
- interoperability with MSRIPv6 IPsec
- Home Agent functionality
- transparent operation for IPv6 transport protocols, including TCP, UDP and ICMP

4.4.4 Known bugs and limitations

Some of the known limitations of the MSR MIPv6 implementation are:

- handoff times are highly dependent on the behaviour of network adapters and drivers
- no support for site-local (scoped) home addresses
- no support for remote Home Agent discovery ICMP messages
- no support for IKE
- no support for forwarding from a previous care of address.

4.5 FreeBSD Support for MIPv6 (Kame)

4.5.1 KAME Overview

The KAME Project is a joint effort to create single solid software set, especially targeted at IPv6/IPsec. Talented researchers from several Japanese major companies joined the project. This

joint effort will avoid unnecessary duplicated development in same area, and effectively provides high quality, advanced featured package.

The KAME Project aims to provide free reference implementations of:

- IPv6
- IPsec (for both IPv4 and IPv6)
- advanced internetworking such as advanced packet queuing, ATM, mobility, and whatever interesting on BSD variants.

Currently several BSD variants are being developed including FreeBSD, NetBSD, OpenBSD, and BSDI as commercial product. They are developing/improving network code separately but there is no single shared reference code for networking.

When consider to use IPv6, there are several choices already. The problem here is, even if *BSD projects choose a single IPv6 stack to merge into them, the code will be maintained by each project separately and these code likely to be quite different on each project tree.

We thus formed a project to implement and maintain the best available code for IPv4/IPv6/IPsec/whatever, which will be the basis of advanced internetworking in the 21st century.

The KAME project was started as a 2-year project (April 1998 - March 2000). It has got extension for 2 years TWICE, so will be until March 2004 at this moment. Core researchers are from the following companies (in alphabetical order):

- Fujitsu Limited
- Hitachi, Ltd.
- Internet Initiative Japan Inc. (IIJ)
- MGCS (Matsushita Graphic Communication Systems, Inc.
- NEC Corporation
- Toshiba Corporation
- YDC Corporation (former Yokogawa Digital Computer)
- Yokogawa Electric Corporation.

Core researchers have committed to work on the IPv6 stack more than 3 days per week, in full-time manner. Therefore, the project is the primary task for the core researchers. The primary task for them is to implement the best networking code possible, under BSD copyright. Also note that the code is available as free software on a “as is” basis without warranty and available for commercial use.

KAME's code is based on the WIDE Hydrangea IPv6/IPsec stack. Several other codes provided by the above companies are merged implementing and improving the software.

The basic specification has been implemented[1]:

- IPv6:
    - Basic specifications

[1] Kazuhiko Yamamoto, “IPv6 activities in Japan”, IIJ Research Laboratory, June 2001.

  - Routing: RIPng, OSPFv3, BGP4+, PIM-DM, PIM-SM
  - Translator: TCP-relay and protocol translator
- IPsec, IKE, Mobile IP
- Many IPv6 applications:
  - DNS, SMTP, POP, HTTP, FTP, TELNET, SSH...
- Adopted:
  - BSD/OS 4.2, FreeBSD 4.2, NetBSD 1.5, OpenBSD 2.8,
  - IIJ SEIL T1, Hitachi GR2000, Fujitsu NetVehicle.

4.5.2 MIPv6 Overview

The original KAME MIPv6 code is based on MIPv6 contributions from Ericsson, NEC[1] and SFC[2]. To benefit from the advantages of each implementation the codes had been merged into one common code, the KAME/MIP6 code. As a result now parts of the KAME MIPv6 implementation are from Ericsson, some from NEC and some from SFC. The development of the code is going on to support the latest MIPv6 specification and to provide a stable, full featured MIPv6 for KAME users.

The current code is based on draft 15 [12]. The release supports very basic functions of MIPv6. Currently, only FreeBSD and NetBSD have been checked to work as a Home Agent (HA), a Mobile Node (MN) and a Correspondent Node (CN). Other BSDs (OpenBSD and BSD/OS) have not been tested yet.

Currently the Ericsson code integrated before has been removed. The last KAME snap that includes the Ericsson MIP code is 20010604. A patch for the 20010611 snap is available from ftp://ftp.kame.net/pub/kame/contrib/mip6/ericsson/.

Also, NEC provides a Mobile IPv6 patch[1] (based on draft 13) available for the KAME stack.

4.5.3 Installation

The Mobile IPv6 implementation has been integrated as part of the FreeBSD kernel. It consists of three different parts: The CN, which is mandatory, must always be included for an IPv6 node to claim "IPv6 compliance". The HA part is optional and required if a router is acting as a HA at the home network. The MN part is optional and is required by a node (router or host) to be able to move between different sub-networks while maintaining transport and higher layer connections. A node may not act as MN and HA simultaneously. MIP6 is not enabled by default. The Mobile IPv6 protocol must be supported in the kernel before configuration starts.

One has to prepare a new kernel configuration file and rebuild the kernel to be able to speak the MIP6 protocol. Also, some user-space commands need recompilation.

The installation of KAME/MIP6 requires specifying the appropriate kernel options in the kernel configuration file and recompiling the kernel. The following options are available:

- `options MIP6`

[1] http://info.6bone.nec.co.jp/mipv6/
[2] http://neo.sfc.wide.ad.jp/~mip6/

- `options MIP6_DEBUG`
- `options MIP6_ALLOW_COA_FALLBACK`
- `# options MIP6_DRAFT13`
- `pseudo-device hif 1.`

If you specify MIP6_DEBUG, the kernel will print many debugging messages. These debug messages can be enabled/disabled at run time using the mip6control program.

MIP6_ALLOW_COA_FALLBACK enables the Care-o-Address (CoA) fallback feature. In the MIP6 specification, the author declares that all IPv6 nodes must support the home address destination option. But, there are not so many implementations which already support this option. If the peer doesn't recognize the home address destination option, the MN can't communicate with that node. If you specify MIP6_ALLOW_COA_FALLBACK, the kernel will try to use its home address as a source address without the home address destination option. If this approach fails, the kernel will use the CoA as its source address the next time to connect to the same peer. The former violates the MIP6 specification and the latter prevents the MN from moving from one network to another network, though, it is very useful to have this option during a transition period in which not all implementations support the home address destination option.

The use of the MIP6_DRAFT13 option enables MIP6 functionality which is compliant to draft-ietf-mobileip-ipv6-13.

The default home interface of the MN is hif0.

Users who are using draft-ietf-mobileip-ipv6-15 can use the following 3 functions:

- `rtadvd`
- `mip6control`
- `had.`

If a HA is required, `rtadvd` (The router advertisement code has been changed according to the requirements for Mobile IPv6) has to be rebuilt with the `MIP6` option and `had`, too. `mip6control` is a control command for the KAME/MIP6 functions. All users need to build `mip6control` to gain access to the KAME Mobile IPv6 functionality (The latest KAME/FreeBSD release will automatically compile `mip6control` and install it to the proper directory).

4.5.3.1 Set up a Home Agent

To establish a HA the following settings have to be completed:

- HA subnet anycast address
- prepare `rtadvd.conf` for a HA
- invoke `rtadvd` with `-m` flag
- invoke `had.`

4.5.3.2 Assign HA subnet anycast address

Assigning a HA subnet anycast address makes it possible to do DHAAD (Dynamic HA Address Discovery). The anycast address is calculated as follows.

If you have 64 bits length prefix, concatenated address of your prefix and 0xfdffffffffffffe is the HA subnet anycast address. If your prefix is not 64 bits, fill host part bits from curving proper bits from the value 0xffffffffffffffffffffffffffffe. For example, if your prefix is 2001:200:1:2::/64, the HA subnet anycast address is 2001:200:1:2:fdff:ffff:ffff:fffe.

```
# ifconfig fxp0 inet6 2001:200:1:2:fdff:ffff:ffff:fffe anycast
alias
```

This address must be configured before `had` is invoked.

4.5.3.3 Prepare `rtadvd.conf` for a HA

Here is the example of the `rtadvd.conf`.

```
fxp0:\
 :maxinterval#60:mininterval#40:
```

When `rtadvd` is invoked with the `-m` switch, it will automatically generate the proper router advertisement for mobile use. Basically, you need not to prepare `rtadvd.conf`. The above example is for users who want to change the advertising interval from the default value.

4.5.3.4 Invoke rtadvd

Invoke `rtadvd` with `-m` option. This option enables the MIP6 feature of `rtadvd`. For example, in case of a fxp0 interface,

```
# /usr/local/v6/sbin/rtadvd -m fxp0
```

4.5.3.5 Invoke had

Invoke `had` with the interface name that you want to enable the DHAAD on. Without had the DHAAD feature will not work.

```
# /usr/local/v6/sbin/had fxp0
```

4.5.3.6 Starting a Home Agent

To start a HA, the following command has to be issued as root. Prior to this, the setting of the anycast address and daemons described above must have finished.

```
# mip6control –g
```

To set a rule to avoid adding home address option when querying DNS, issue the following command:

```
# mip6control -u ::#53
```

4.5.3.7 Starting a Mobile Node

To make a node act as a MN, you must specify your home network prefix. To do this, issue the following command as root.

```
# mip6control –i hif0 -H2001:200:1:1:: -P64
```

Replace `'2001:200:1:1::'` with your home prefix. After prefix setting has finished, enable the mobility function using the `-m` option.

```
# mip6control -m
```

To detect movement, a MN needs to receive Router Advertisement packets. The easy way is to invoke the `rtsol` command. You may want to run `rtsold` with `–a -m` options to make the node detect its location quickly.

4.5.3.8 Setting up Security Features

The KAME/MIP6 can protect the binding update/binding acknowledgements using a security mechanism. By default, the KAME/MIP6 uses the authentication sub-option defined in the draft-ietf-mobileip-ipv6-15 to protect them. To protect them, you must set up the security associations between the nodes. Currently the KAME/MIP6 re-uses the security association database for the IPsec stack of the KAME. So, you need to use the `setkey` program to set up the security associations.

For example, if you want to protect the binding update/acknowledgements between the MN whose address is A and the HA whose address is B, set up the security association as described below:

```
add A B ah 1500 -m transport -A hmac-sha1 "AH SA configuration!";
add B A ah 1600 -m transport -A hmac-sha1 "AH SA configuration!";
```

Also, you must set up the security policy as follows:

```
spdadd ::/0[any] ::/0[any] ipv6-opts -P out ipsec
ah/transport//require;
```

If you don't want to protect them, you can disable this feature using `mip6control` program. To disable the authentication data protection, type the following:

```
# mip6control -T 0
```

4.5.4 Features

The command `mip6control` controls the KAME/MIP6 features (The newest version dates from March 12, 2002). The `synopsis of the command is`

```
mip6control [-iifname] [-abcghlmnw] [-Hhome_prefix -P prefixlen]
 [-Ahome_agent_global_addr -L home_agent_linklocal_addr]
 [-uaddress#port] [-vaddress#port] [-S0|1] [-T0|1] [-D0|1]
```

`mip6control` sets/gets KAME/MIP6 related information as follows:

`-i ifname`

Specify home interface of the MN. The default value is `hif0`.

`-H home_prefix`

Set `home_prefix` as a home prefix of the MN. You must specify the prefix length of `home_prefix` with -P option.

`-P prefixlen`

Specify the length of the prefix to be assigned to the MN. Use with `-H` option.

`-A home_agent_global_address`

Specify the global address of the HA of this MN. If your HA supports DHAAD (Dynamic Home Agent Address Discovery), you need not use this switch. Use with `-L` option.

`-L home_agent_linklocal_address`

Specify the linklocal address of the HA of this MN. If your HA supports DHAAD (Dynamic Home Agent Address Discovery), you need not use this switch. Use with `-A` option.

`-m` Start acting as a MN.

`-g` Start acting as a HA.

`-u Address#Port`

Add a rule that MN doesn't add a Home Address option to the outgoing packet.

`-S 0|1` When set to 0, the IPsec protection check of the incoming binding updates and binding acknowledgements will not be performed (always pass the check).

`-T 0|1` When set to `0`, the authentication sub-option check of the incoming binding updates and binding acknowledgements will not be performed. Also, do not insert the authentication sub-option when sending the binding updates/binding acknowledgements. Note that, `mip6control -S 1 -T 0` doesn't mean 'use only the IPsec mechanism to protect binding updates and binging acknowledgements' in ID-15 environment. This is equivalent to `mip6control -S 0 -T 0` in ID-15 environment.

4.5.4.1 Proxy Neighbour Discovery

This feature is fully supported by the implementation.

4.5.4.2 IPv6 Encapsulation & Decapsulation

IPv6 encapsulation and decapsulation are supported.

4.5.4.3 Dynamic Home Agent Address Discovery

DHAAD is implemented along with alternative static assignment of the Home Agent address.

4.5.4.4 Home Address Option

The HAO is supported in the implementation. There is also support for the Mobile Node to fallback to using its home address as the source address for outgoing packets, to be compatible with Correspondent Nodes that do not support the HOA.

4.5.4.5 Movement Detection

Movement detection is achieved by the CoA changing, which is realised via Router Advertisement messages.

4.5.4.6 Smooth Handoff

This functionality is not supported by the implementation.

4.5.4.7 IPsec

KAME/MIP6 supports both the IPsec and the authentication data sub-option mechanism described in draft 15. Kernel IPsec support ("options IPSEC" in the kernel config file) is optional, but if either the MN or HA have kernel support, then both sides must activate IPsec between the two hosts. (The Mobile IPv6 implementation makes no demands as to the IPsec protocol (AH, ESP) used between the two hosts, nor to the security protocol mode (tunnel, transport, any). However, if you define IPsec and a SPD entry of upper specification ``any" between e.g. MN and HA, together with SAs using AH, Authentication Headers will be included in the Binding Updates and Binding Acknowledgement.)

If IPsec is used, the IPsec SPD/SA database has to be set up as usual. If the authentication data sub-option is chosen, SPD has to be set up with protocol number 60 (ipv6-opts, this is subject to change), and SA with protocol AH. The IPsec protection/validation and the authentication sub-option protection/validation are enabled by default

4.5.4.8 Key exchange

KAME does not provide any key distribution mechanism. Setting SAs must be done manually.

4.5.4.9 Support for notebooks / PDAs

Poor notebook support.

4.5.4.10 MIP built-in

MIP6 is not enabled by default. One has to prepare a new kernel configuration file and rebuild the kernel that is able to speak the MIP6 protocol. Also, some user-space commands need recompilation.

4.5.4.11 Other Features

Other features supported by the KAME implementation include:

- MN functions:
  - binding update/home registration
  - neighbour advertisement when returning to home
- CN functions:
  - receiving binding update
  - insertion of a routing header

- sending binding requests

- HA functions:
  - home registration
  - encapsulate packets destined to MNs
  - dynamic Home Agent discovery
  - duplicate address detection when registering CoA.

4.5.5 KAME snap 20010604 characteristics

- Problems with HA functionality when HA is activated on more than 1 interface. An additional Ericsson patch solved this problem.
- HAs are configured statically since DHAAD is based on anycast routing and this kind of routing had not yet been fully implemented.
- It is recommended to advertise 64 bit long prefixes on shared media like Ethernet or WaveLAN (i. e. non-PPP links).
- Some random crashes (probably network interface dependent).
- HA- / MN restart after configuration changes needs rebooting.

4.6 Lancaster University's MIPv6 for Linux

4.6.1 Overview

Lancaster University's MIPv6 stack for Linux was first released to the public on 3rd April 1998. We believe this was the first publicly demonstrated implementation of Mobile IPv6, and also the first version complying with the Mobile IPv6 draft 5 specification to be released. The implementation has continued to be updated as the MIPv6 draft standard has progressed. Furthermore, in research collaborations with Microsoft and Cisco, Lancaster University's MIPv6 implementation was used as a platform to provide MIPv6 implementations for Microsoft Windows 2000 and Cisco IOS respectively. The latest version of the MIPv6 code is based on IETF draft specification version 13 and is available from http://www.cs-ipv6.lancs.ac.uk/ipv6/MobileIP/. It is provided as a kernel installable module and works with Linux kernels 2.4.16 and 2.4.17. The implementation can support Mobile Node, Home Agent (mobile-aware router), and Correspondent Node functionality.

4.6.2 Installation

There are three steps to installing Lancaster University's MIPv6 for Linux:

1. Patch the kernel.
2. Compile the kernel module.
3. Configure the system to setup Mobile IPv6 on boot/demand.

4.6.2.1 Patching the kernel

A patch for kernel versions 2.4.0 through 2.4.2 is provided in the MIPv6-1.2 directory. Copy the file to /usr/src/linux or where you unpacked the kernel source to. Apply the patch:

patch -p1 < mobile-ipv6-patch-2.4.0

Once the kernel is successfully patched it will need configuring. At a minimum the following networking options need to be set:

[M] The IPv6 protocol

[Y] Mobile IPv6.

You can then build and install the kernel and its modules as per normal.

4.6.2.2 Installing the kernel module

As root, copy module/mobile_ipv6 to:

/lib/modules/KERNEL_VERSION/kernel/net/ipv6/mobile_ipv6.o

Also, copy tools/mipv6_config to:

/usr/sbin/mipv6_config

4.6.2.3 Installing Mobile-aware Routers

Once a working IPv6 network has been realised, a compatible Linux kernel should be built for each router in the system, with IPv6 support configured as a kernel loadable module. Once built and installed, this module should be replaced by rou_ipv6.o, and the system restarted. This will enable Mobile IPv6 Home Agent routines on that router.

In order for Mobile IPv6 to function correctly, router advertisements MUST be broadcast on any network that a Mobile Node may roam to. If router advertisement is not already configured on your system, install the radvd-0.4.2-mobile.tar.gz package, by following the README found inside the archive. We recommend a router advertisement interval of around three seconds, as this results in a satisfactory handover performance, without too much overhead.

4.6.2.4 Installing Mobile Nodes

Installation of the Mobile Nodes kernel module follows the same procedure as described above, with the exception that the rou_ipv6.o module is replaced by mob_ipv6.o.

To gain the full benefit of Mobile IPv6, however, installation of the adapted PCMCIA services is also required. To do this, simply expand the pcmcia-cs-3.0.3-mobile.tar.gz archive into the /usr/src directory of the mobile client. Enter the newly created directory, and type "make all" to build the binaries, then "make install" to move them to their correct locations. Consult the PCMCIA-HOWTO contained in the archive for more details. This archive also contains a pre-patched version of the Linux WaveLAN driver, which supports true MAC layer roaming.

Finally, the Mobile IPv6 stack must be configured. This is done via the "setaddr" and "sethome" commands, are recommended be place in the /usr/bin directory. For convenience, a script, called netstart, is provided which shows examples of the use of these commands, and can be placed in /etc/rc.d/rc.local to automatically configure the stack at boot time. See the netstart script for more details.

4.6.2.5 Installing Correspondent Nodes

To configure a Linux IPv6 host as a Correspondent Node (which is necessary to use route optimisation from that host), simply replace its kernel loadable module with rou_ipv6.o - no other changes are necessary.

4.6.3 Configuration

A configuration script is provided in the scripts directory. This may be called from rc.local, or by hand. Alternatively, a custom script may be created based on the following:

On system startup

insmod ipv6 (if this hasn't already been done)

insmod mobile_ipv6 (found in /lib/modules/VERSION/kernel/net/ipv6/)

/usr/sbin/mipv6_config --mode +-CN +-MN +-HA

(either +CN or -CN, + = enable, - = disable)

For Mobile Nodes only:

echo 0 > /proc/sys/net/ipv6/conf/eth0/dad_transmits

You could leave this alone, but handoff times will be longer.

/usr/sbin/mipv6_config --homeaddr xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx 64

/usr/sbin/mipv6_config --homeagent xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx 64

In the --homeaddr case, the address is your home address

In the --homeagent case, the address is your Home Agent's address.

4.6.4 Features

The major features supported by the Lancaster University MIPv6 implementation for Linux are as follows.

4.6.4.1 Proxy Neighbour Discovery

Proxy Neighbour Discovery is implemented in the Lancaster MIPv6 implementation so that the Home Agent can intercept packets on behalf of the Mobile Node.

4.6.4.2 IPv6 Encapsulation & Decapsulation

The generation of IPv6-in-IPv6 tunnelling (as required by Home Agents) and the respective handling by the mobile is supported.

4.6.4.3 Dynamic Home Agent Address Discovery

Dynamic Home Agent Discovery is not yet available in the Lancaster MIPv6 implementation but is planned to be available soon.

4.6.4.4 Binding Management

The Lancaster MIPv6 stack maintains a bindings cache and bindings update list for the purpose of bindings management. The generation and reception/handling of binding requests, binding updates and binding (n)acks are all implemented.

4.6.4.5 Home Address Option

The generation of the home address option by the Mobile Node and subsequent handling by the Correspondent Node is supported in the current version of the Lancaster MIPv6 implementation.

4.6.4.6 Movement Detection and Handoff

Movement detection is achieved by monitoring unsolicited Router Advertisement messages, however the time taken for a Mobile Node to detect movement is dependent on the rate at which these messages are sent by the router in question. At best a one second granularity is achieved. In order to improve the performance of movement detection, the MIPv6 stack receives feedback from PCMCIA card services and device drivers. The IPv6 address autoconfiguration module traps system calls that 'bring up' a new interface, for example, between Ethernet and 802.11 networks. In this way, a Router Solicitation message can be broadcast by the Mobile Node to stimulate the receipt of a Router Advertisement message. This technique has been publicly tested for achieving efficient handovers using the Lancaster IPv6 video server (released at the same time as the MIPv6 implementation).

4.6.4.7 IPsec / AAA

IPsec is not yet supported.

4.6.4.8 Support for notebooks / PDAs

The implementation has been tested successfully (in Mobile Node and Correspondent Node modes) on Sony Vaio and Dell notebooks.

4.6.4.9 MIP built-in

The MIPv6 implementation is intended as an add-on networking feature for the Linux operating system. Linux distributions do not have the Lancaster University MIPv6 implementation built in.

4.6.4.10 Number of software patches necessary

Assuming that the target Linux machine already has a configured and working IPv6 stack, only one software patch is necessary.

4.6.4.11 Set-up capabilities

A very basic command line utility, mipv6_config, is provided for the configuration of Mobile Nodes.

4.6.4.12 Other Features

Other features supported by the Lancaster University implementation include:

- support of route collapsing/optimisation
- supports applications using UDP, TCP, and ICMP protocols
- new support for MAC layer roaming for Wavelan networks.

4.6.5 Known bugs and limitations

Some know bugs and limitations of the Lancaster University implementation are:

- no IPsec or AAA functionality
- Dynamic Home Agent Discovery is yet to be implemented.

4.7 6WIND

4.7.1 6WIND Edge Device Overview

The 6WIND IP Edge Device is Customer Premises Equipment (CPE) which integrates advanced traffic regulation capabilities, security mechanisms, IPv4 to v6 transition functions, and Mobile IPv6 functionality. Additionally the device provides high level management interfaces that allow rapid configuration of IP services. The IP Edge Device is designed to be situated at the boundary of the backbone network.

Internally the 6WIND device is based upon a custom version of FreeBSD, with bespoke implementations of IPv6 and MIPv6.

4.7.2 MIPv6 Overview

IPv6 mobility is included in a prototype of the IP Edge Device, though this function is not currently included in commercial version of the IP Edge Device.

In the Mobile IPv6 architecture, the 6WIND Edge Device can act as Home Agent, and/or Correspondent Node.

The MIPv6 functionality implemented in the 6WIND Edge Device is based on the IETF Internet Draft “draft-ietf-mobileip-ipv6-13.txt”. As the MIPv6 is still under discussion some aspects of the mechanisms involved in MIPv6 are not very stable. 6WIND has chosen to implement only the mechanisms that seem to be the most stable. So the major mechanisms are implemented, apart from the following restrictions:

- Duplicate Address Detection is not implemented
- Automatic HA discovery is not implemented, so it has to be configured manually
- Renumbering Home Subnet and tunnelling router advert to MN is not implemented
- As Security mechanisms proposed by Draft-13 have been removed from new Draft versions (14 and 15) 6WIND has not pursued the implementation of the MIPv6 Security aspect. IPSEC does not seem to be adapted to this functionality so a new approach needs to be defined
- A first set of commands provided by the CLI (Command Line Interface) allows management of basic operations.

For security reasons, MIPv6 is not activated by default. In fact, activating this functionality can create vulnerabilities. Within an operational network, if the 6WIND Edge Device is used for its security features, VPN or Firewall, it is not recommended that MIPv6 is used until the security specifications and the associated implementation stabilise.

4.7.3 Installation

The Mobile IPv6 implementation has been integrated as part of a prototype version of the 6WIND system. MIPv6 is not activated by default.

4.7.3.1 Configuring IPv6 Autoconfiguration Mechanisms

The use of Mobile IPv6 requires the activation of IPv6 auto-configuration mechanisms on the edge device.

4.7.3.2 Displaying Mobile IP Information

Mobile IPv6 status can be displayed using the following commands:

```
ED1{myconfig}display mip
ED1{myconfig-mip}display
```

Example:

```
ED1{myconfig-mip}display
    # MOBILITY STATEMENT
      mobility home_agent
```

4.7.3.3 Modifying the Mobile IP behaviour

To configure the Edge Device as a Home Agent, the following command is used:

```
ED1{myconfig-mip}mobility home_agent
```

By default, the Router Advertisements are sent every 30 seconds. As soon as the Edge Device is configured as a Home Agent, the advertising period is set to 1 second.

To configure the Edge Device as a Correspondent Node, the following command is used:

```
ED1{myconfig-mip}mobility correspondant
```

To disable the mobility function, the following command is used:

```
ED1{myconfig-mip}mobility none
```

4.7.4 Features

4.7.4.1 Proxy Neighbour Discovery

The HA will provide proxy neighbour discovery.

4.7.4.2 IPv6 Encapsulation & Decapsulation

IPv6 encapsulation and decapsulation are available functionality.

4.7.4.3 Dynamic Home Agent Address Discovery

This feature is not implemented in this version of the system.

4.7.4.4 Home Address Option

The home address option is implemented and provides the basic MIPv6 functionality.

4.7.4.5 Movement Detection

Movement detection is achieved through the use of reduced prefix announcement timers, thus providing a new Care of Address on entry to a new network.

4.7.4.6 Smooth Handoff

Since the edge device does not function as a Mobile Node this criterion is not really relevant.

4.7.4.7 Security mechanisms

There are no security measures currently available in this version of the implementation, due to the lack of stability in the draft standards.

4.7.4.8 Support for notebooks / PDAs

The Edge device has been successfully tested against the 0.9 MIPL MIPv6 implementation for Linux, running on an iPaq and on a laptop.

4.7.4.9 MIP built-in

MIP6 exists in the prototype distribution but is not enabled by default. It has to be enabled through the command interface.

4.7.5 Future developments

Several actions are foreseen for the development and improvement of MIPv6 in the Edge Device.

6WIND is carefully following the deliberations of the IETF. The MIPv6 standard is not yet stable and new drafts have been recently issued. The changing nature of the security provisions in the draft standards make it difficult, at this stage, to decide on security implementation. For example, there have been fundamental changes in the way the Binding Update is authenticated in recent drafts. Work is ongoing regarding implementation of the latest versions of the standards.

There are plans for enhancement of the MIPv6 Management by adding high-level commands.

Interoperability tests will be performed in order to improve the implementation.

5 Summary and Conclusions

The MIPv6 implementations investigated in the previous section are summarised in Table 5.

	MIPL	Cisco	Microsoft	KAME	ULANC	6WIND
Platform	Linux	Cisco IOS	2000/XP/CE	FreeBSD	Linux	FreeBSD
Draft version	15	13	13	15	13	13
Modes	MN/HA/CN	HA/CN	MN/HA/CN	MN/HA/CN	MN/HA/CN	HA/CN
PND	Yes	Yes	Yes	Yes	Yes	Yes
IPv6-in-IPv6 tunnelling	Yes	Yes	Yes	Yes	Yes	Yes
DHAAD	Yes	No	No	Yes	No	No
Binding Management	Yes	Yes	Yes	Yes	Yes	unknown
HAO	Yes	Yes	Yes	Yes	Yes	Yes
Movement Detection	RAs	N/A	RAs and NDIS notifications	RAs	RAs and PCMCIA traps	N/A
Smooth Handoff	Yes	Yes	Yes	No	Yes	N/A
IPsec/AAA	IPsec	No	IPsec and CAM	IPsec	No	No
Key exchange	MD5 or SHA-1	No	sagen utility	manual	No	No
Support for notebooks/ PDAs	Yes	N/A	Yes	Poor	Yes	N/A
MIPv6 built-in	No	Yes	No	Yes but not enabled by default	No	Yes but not enabled by default
No. of patches	1	0	1 (XP and CE) 2 (2000)	0	1	0
Set-up tools	mipdiag	command line tools	Auto-configuration and MIPv6Conf	command line tools	mipv6_config	command line tools

Table 5 Summary of MIPv6 Implementations

As the table shows, it is easy to see that the implementations have varying levels of support for MIPv6 features and to which draft version they are based upon. In most cases, MIPv6 functionality must be either applied as a separate patch to the target operating system or explicitly enabled by the host administrator. This is understandable, considering the experimental nature of the protocol, however it does present a difficulty to users of a non-technical nature who may be interested in experimenting with MIPv6.

It is also evident from the table that few implementations have support for both IPsec and a key distribution algorithm. However, the introduction of the mobility header in draft 16 [13], which allows for IPsec protection of MIPv6 signalling, may result in accelerated deployment of IPsec support in the near future.

One can conclude that the rather incomplete nature of the MIPv6 implementations we investigated is only to be expected at the time of writing. The MIPv6 standard is still work in progress with development of draft version 17 currently underway. Consequently, MIPv6 implementers are at different stages of MIPv6 support depending on the draft version that they began their work with.

Furthermore, as implementations are updated, vendors may choose to wait for a future draft version rather than update their code to the current version at that point in time. For example, many MIPv6 implementers may choose to wait for draft version 17 (expected to be released soon) rather than change their code to reflect the recent changes made by draft version 16.

References

[1] KAME homepage http://www.kame.net
[2] Monarch homepage http://www.monarch.cs.rice.edu
[3] NEC Mobile IPv6 Implementation Project http://www.6bone.nec.co.jp/mipv6
[4] 6WIND homepage http://www.6wind.com
[5] Mobile IPv6 for Linux homepage http://www.mipl.mediapoli.com/
[6] Mobile IP at NUS http://opensource.nus.edu.sg/projects/mobileip/mip.html
[7] Nokia homepage http://www.nokia.com
[8] Symbian homepage http://www.symbian.com
[9] Ericsson homepage http://www.ericsson.com
[10] SFC-MIP6 homepage http://neo.sfc.wide.ad.jp/~mip6/english/index.html
[11] IPv6 Forum Korea, "Mobile IPv6 – Implementation Trends of Mobile IPv6", 2001-005
[12] David B. Johnson, Charles Perkins, "Mobility Support in IPv6" draft-ietf-mobileip-ipv6-15.txt, Internet Draft, July 2001, work in progress
[13] David B. Johnson, Charles Perkins, "Mobility Support in IPv6" draft-ietf-mobileip-ipv6-16.txt, Internet Draft, March 2002, work in progress
[14] The Cisco IPv6 Statement of Direction, available for download via http://www.cisco.com/warp/public/732/Tech/ipv6/ipv6_learnabout.shtml
[15] The LandMARC homepage http://www.landmarc.net/
[16] C. Perkins, "IP Mobility Support for IPv4", IETF RFC 3220, January 2002.
[17] G. O'Shea, M. Roe, "Child-proof Authentication for MIPv6 (CAM)", Computer Communication Review, Vol. 31, No. 2 (April 2001).
[18] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", IETF RFC 2460, December 1998.
[19] C. Perkins, "IP Mobility Support for IPv4 (revised)", IETF RFC 3220, January 2002.
[20] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", IETF RFC2461, December 1998.
[21] MSR IPv6 http://research.microsoft.com/msripv6/

Appendix A Cisco IOS Configuration

Standard command line commands are used to configure MIPv6 functionality on the Cisco IOS "Technology Preview" release. Below are the relevant configuration commands, along with an example configuration.

Global commands:

```
[no] ipv6 mobile

  Enables mobile IPv6.
  Default is disabled; all binding updates are ignored.

[no] ipv6 mobile bindings filter <prefix-list>

  Configures a binding update filter using a prefix list. When a
  prefix list is configured all binding updates are filtered by
  source address.
  Default is no prefix list; all binding updates are accepted.

[no] ipv6 mobile binding lifetime

  Configures maximum binding cache entry lifetime.
  Default is to use the lifetime in the received BU.

[no] ipv6 mobile binding max

  Configures maximum number of binding cache entries.
  Default is an unlimited number of entries.
```

Interface subcommands:

```
[no] ipv6 mobile home-agent enable

  Enables a Home Agent on the interface.
  Default is disabled.
```

```
[no] ipv6 mobile home-agent access <acl-name>

  Configures an ACL to filter home registration binding updates
  on this interface.
  Default is no ACL configured; all binding updates are accepted.

[no] ipv6 mobile home-agent lifetime <secs>

  Configures the Home Agent lifetime.
  Default is the default router lifetime.

[no] ipv6 mobile home-agent preference <pref>

  Configures the Home Agent preference.
  Default is 0.

[no] ipv6 nd advertisement-interval

  Configures whether to send the Advertisement Interval option in
  Router Advertisments.
  Default is not to send option.
```

Changes to existing Interface subcommands:

```
[no] ipv6 nd ra-interval <secs> | msec <msecs>

  Configures the interval between IPv6 Router Advertisement
  transmissions out this interface. The value for this should be
  less than or equal to the IPv6 Router Lifetime if this is a
  default router. The default is 200 seconds. To prevent
  synchronization with other IPv6 nodes the actual value used may
        be randomly adjusted to within +/- 20%   of the specified value.

  msec <mecs>  The "msec" qualifier allows the interval to be
               set to a lower value to aid movement detection by
               a mobile-node.

[no] ipv6 nd prefix <prefix> | default
```

```
                    [ [<valid-lifetime> <preferred-lifetime>] |
                      [at <valid-date> <preferred-date>]
                      [off-link] [no-rtr-address] [no-autoconfig] ]
      By default all prefixes configured as addresses on the
      interface will be advertised in Router Advertisements. This
      command allows control over the individual parameters per
      prefix, including whether the prefix should be advertised or
      not. The "default" keyword can be used to set default
      parameters for all prefixes.
      A date can be set for prefix expiry. The valid and preferred
      lifetimes are counted down in real time. When the expiry date
      is reached the prefix will no longer be advertised.

        no-rtr-address  Do not send full router address in prefix
                        advert, and do not set R bit.
```

Show commands:

```
    show ipv6 mobile binding [care-of-address <addr>]
                             [home-address <addr>]
                            [interface <int>]

      Show the binding cache, with optional filtering parameters.

    show ipv6 mobile globals

      Show global Mobile IPv6 parameters.

    show ipv6 mobile home-agents [<interface>]

      Show neighbouring Home Agents, on the optional interface.

    show ipv6 mobile traffic

      Show binding updates received and binding acknowledgements
      transmitted.
```

Clear commands:

```
clear ipv6 mobile binding {care-of-address <prefix>
                           | home-address <prefix>
                          | interface <int>}

  Clears the Binding cache for the given parameter.
  The parameter can be a prefix for the CoA or HoA, so that
  entire networks can be cleared. Use /128 to clear an
  individual.
  The interface parameter clears all bindings on the given
  interface.

clear ipv6 mobile home-agents <interface>

  Clears the neighbouring Home Agents list on the given
  interface.

clear ipv6 mobile traffic

  Clears the statistics about the received binding updates, and
  transmitted binding acknowledgements.
```

Debug commands:

```
[un]debug ipv6 mobile correspondent-node

  Enables correspondent node debugging.

[un]debug ipv6 mobile forwarding

  Enables soft tunnel forwarding debugging.

[un]debug ipv6 mobile home-agent

  Enables Home Agent debugging.

[un]debug ipv6 mobile registrations
```

```
Enables registrations debugging, i.e. binding updates and
binding acknowledgements.
```

Example configuration

The following example configures a router as a Mobile IPv6 Home Agent on Ethernet 1

```
ipv6 mobile
!
interface Ethernet1
 ipv6 address 3000:1234:5678::1/64
 ipv6 mobile home-agent enable
```