www.feedface.com

A short overview of the SHISA MIPv6 stack

Folkert Saathoff, Dec 2005

This text represents the results of my research of the SHISA implementation of the Mobile IPv6 protocol. It should be considered as a byproduct of my diploma thesis, *Evaluation of QoS-Aspects of mobile IPv6 Clients in an IEEE 802.11 network*. When I started reading up on Mobile IPv6, there was not a lot of information apart from RFCs, the source and of course the helpful people on the kame mailing lists. So in order to help others who want to set up this rather cool network mobility protocol, I wrapped up the more practical parts of my thesis into one shorter document.

The first chapter gives an overview of the MIPv6 protocol as defined in the corresponding RFCs. The following two chapters describe in brief the components of the KAME IPv6 stack and the SHISA MIPv6 stack, as of September 2005. After that, the setup of a wireless test network, using the FreeBSD operating system, is explained.
Note that you will not find anything about IPsec in this document. Also, I only describe the commands which are needed to configure the nodes, without taking into account the startup config files included in the KAME stack.

If you have any questions or feedback, you may contact me at folkert@feedface.com.

enjoy.

Chapter 1

Mobile IP Version 6

1.1 Goal and Classification

Mobile IP is designed to allow a node on the internet to use the same network layer address on different locations on the network. This enables other nodes to address the mobile node always with only one address, regardless of its actual location. It also makes it possible for a mobile node to maintain existing transport layer connections across different locations. The MIPv6 protocol provides mobility for single hosts only. For the mobility of routers and entire networks, the NEMO network mobility protocol is specified separately.
MIPv6 works as a network layer protocol since it provides node-to-node connectivity to the upper layers. It depends itself on a network layer protocol, namely IPv6, to handle the actual routing of packets between the nodes involved. Thus, MIPv6 is best regarded as a specialized layer in the network stack, located between the network layer and the transport layer.

1.2 Overview

1.2.1 Definitions

The following definitions are used in the discussion of Mobile IP:

Mobile Node (MN) is the node for which mobility is required.

Home Agent (HA) is a node on the Mobile Node's Home Link that supports the MN in mobile communication when not at home.

Foreign Router (FR) is a router on a link the Mobile Node is connected to when not at home.

Correspondent Node (CN) means any node with which the Mobile Nodes communicates using its Home Address. Nodes with which a Mobile Node communicates using its Care-of Address are regarded as regular IP nodes.

Home Link means the link which is considered the home location of the Mobile Node. In MIPv6, it is not necessary for the MN ever to visit its Home Link, it may just as well always move from one Foreign Link to another.

Foreign Link means any other link on which the Mobile Node may be located at some time.

Home Address (HoA) means the address at which the Mobile Node should be reachable at all times, regardless of its actual location.

Care-of Address (CoA)] is a globally routable address at which the Mobile Node is reachable when not located on its Home Link.

1.2.2 The Problem with MIPv4

Mobility Support for IPv4 is specified in [Per02]. It defines a Home Agent and a Foreign Agent, which both assist the Mobile Node in attaining its mobility. A Foreign Agent is required on each link a Mobile Node visits. This Foreign Agent is responsible for informing a Mobile Node about its current location on the network. All packets from the Mobile Node are then tunneled between Foreign Agent and Home Agent, which pass the packet along to the Mobile Node and Correspondent Node, respectively. This approach has two major problems impeding a wide-scale deployment on the internet. First, it depends on the availability of a Foreign Agent on every potential network location of a Mobile Node. Because the provider of this network location in most cases will not gain anything by providing such a service, mobility will not be available on all network locations. This greatly diminishes the reliability of MIPv4. Second, the fact that all traffic to and from a Mobile Node is tunneled to the Home Agent results in a significant amplification of overall network traffic, as well as in increased latency of all mobile traffic. While both factors might be neglected when using MIPv4 on an internal network, they make a success of MIPv4 on a global scale improbable. MIPv4 is described in depth in [Sol98].

1.2.3 MIPv6 Concept

Consequently, MIPv6 was designed not to demand a MIPv6 aware node on a Foreign Link. A Home Agent located on the Mobile Node's home network location assists the MN in its mobility. On the foreign network location, only a standard IPv6 router is required. Because the Home Agent shares a network prefix with the Mobile Node's Home Address, all packets destined for the Mobile Node can be intercepted by the Home Agent. It then tunnels those packets to the Mobile Node. Conversely, packets from the Mobile Node are tunneled to the Home Agent, which then routes them to their final destination. MIPv6 also addresses the problem of increased latency by introducing an optimized routing procedure. With Correspondent Nodes that are aware of MIPv6, route optimization can be used instead of tunneling. This is done by sending packets directly to and from the Mobile Node's Care-of address, including the Home Address in a special IPv6 extension header. This results in the packets being routed along the most efficient path between Mobile Node and Correspondent Node, thus decreasing the latency of the connection as well as the overall network load significantly.

1.3 Types of Nodes

There are four different kinds of nodes involved in Mobile IP communications. Only the Mobile Node and the Home Agent have to be aware of the MIPv6 protocol. The Foreign Router and Correspondent Node may be regular IPv6 nodes. A person or organization wanting to use MIPv6 with their mobile equipment only has to set up a Home Agent on their network and configure the Mobile Nodes correctly. This is easy to do since the person or organization in question already has full control over these nodes. The nodes over which no single entity has full control are the routers supplying network uplink to a MN while not at home, as well as all other nodes with which the MN communicates potentially. Because those are not required to be aware of MIPv6 at all, the mobility features can be attained by anyone wanting them, without requiring a massive collaborative act of all participants on the network. In short, it is possible to phase in MIPv6 in an existing IPv6 network.

1.3.1 Mobile Node

The Mobile Node is the node for which network mobility is required. Typically, this is a portable computer or some kind of embedded device. The Mobile Node is responsible for:

- Determining its own network location by looking at the network prefix propagated on the current link.

- Notifying the Home Agent about its current network location by sending Binding Updates.
- Informing a Correspondent Node about its own mobility and determining whether the CN is mobility-aware.
- Notifying a mobility-aware Correspondent Node about its current network location by sending Binding Updates.
- Tunneling packets destined for non-mobility-aware CNs via the Home Agent.
- Receiving tunneled packets from the Home Agent.
- Sending packets to a mobility-aware CN by prepending a Destination Header.

To function properly, a Mobile Node has to implement a binding update list, in which it keeps information about all bindings it has established with Correspondent Nodes. The current binding with the Home Agent is stored in the binding update list as well. The Mobile Node also keeps track of Correspondent Nodes with which previous binding attempts have failed and which thus should be contacted only via the HA.
Since a Mobile Node implements the MIPv6 protocol, it will act as a Correspondent Node when in communication with other Mobile Nodes.

1.3.2 Home Agent

The Home Agent is responsible for supporting the Mobile Node in its mobile communications. It has to be stationary itself and located on the Home Link. While a dedicated node could be used as HA, it is practicable to assign this task to a router on the Home Link. Because the router already receives all packets destined for the MN's Home Address, it is in a perfect position to act as a Home Agent, routing packets either to the CoA or to the HoA, depending on the MN's current location. The Home Agent is responsible for the following tasks:

- Keeping track of the Mobile Node's current location by listening for Binding Updates from the MN.
- Tunneling packets which are destined for the Mobile Node, but were routed to its Home Address, to the MN's current Care-of Address.
- Routing packets to a Correspondent Node that were tunneled to the Home Agent by the Mobile Node.

To accomplish these tasks, a Home Agent has to implement a binding cache in which it stores the current Care-of Address of every Mobile Node it is responsible for.
Since a Home Agent implements the MIPv6 protocol, it will act as a Correspondent Node when communicating with another Mobile Node for which it itself is not responsible.

1.3.3 Foreign Router

Except for its obvious task of routing packets from and to a Mobile Node visiting its link, a Foreign Router has no special tasks in MIPv6. Nevertheless, it is responsible for:

- Propagating the network prefix configured on the link, to help the MN determine its current location.

1.3.4 Correspondent Node

Every Node with which a Mobile Node communicates using its Home Address is a Correspondent Node by definition. A CN may or may not be aware of the mobility of the MN. If a CN is not aware of the mobility, it is expected to behave just like every other IP node. If a CN is aware of the mobility, it is responsible for:

- Keeping track of a Mobile Node's current Care-of Address by listening for Binding Updates.
- Confirming the plausibility of a Binding Update from a Mobile Node by verifying that the MN does receive packets on both its Home Address and Care-of Address.
- Requesting further Binding Updates from a MN if further communication is desired by the CN.
- Routing packets directly to the MN's Care-of Address by prepending a Routing Header.

To accomplish these tasks, a Correspondent Node has to implement a binding cache in which it stores the current Care-of Address for every Mobile Node it is in communication with.

1.4 Communication Modes

There are different ways for a Mobile Node to exchange IP traffic with a Correspondent Node. While the Mobile Node is located at its Home Link, no special procedure is needed. In this

case, all traffic is routed normally to and from the Home Address. If the MN is away from its Home Link, two modes of communication are possible:

The first mode is *tunneling*. For tunneling, the CN does not have to be aware of the mobility of the MN. The backside is that every packet produces extra traffic on the network. Also, a higher latency is imposed on the packet because it might not take the shortest path between MN and CN. For tunneling to work, the Home Agent has to be aware of the current Care-of Address of the MN.

The second mode is *route optimization*. For this, the CN has to be aware of the current Care-of Address of the MN. All packets are sent directly between the MN and the CN, thus imposing neither extra latency nor traffic. The backside is that the Correspondent Node has to implement the MIPv6 protocol.

1.4.1 Tunneling

In tunneling mode, all packets between Mobile Node and Correspondent Node are sent via the Home Agent. The CN sends packets to the Home Address, as it would do with all regular IP packets. The Home Agent intercepts these packets and encapsulates them in a new IP packet containing the Care-of Address as destination and the HA's address as source. These packets are then routed to the MN, which disassembles them, thus getting the original packet. Conversely, the MN encapsulates packets destined for the CN in an IP packet containing the HA's address as destination and the CoA as source. The Home Agent then disassembles the packets and routes them to the Correspondent Node.

Because the CN is not aware of the mobility of the MN, all packets are routed to the Home Agent first, even if the MN's current location could be reached faster and cheaper by another route. The HA has to handle the same amount of traffic as the MN and CN, effectively doubling the traffic necessary for the communication. Because of this, the Home Agent needs to be located on a link with a high -and, just as important, symmetrical- bandwidth.

If the HA is also the router responsible for the Home Link, it will receive the packets destined for the MN automatically. If not, the HA has to employ Proxy Neighbor Discovery to receive the packets. In Proxy Neighbor Discovery, the HA advertises its own link-layer address in response to solicitations for the MN's Home Address. Thus, all packets destined for the HoA are sent to the HA. The Proxy Neighbor Discovery procedure is defined in [Nar98, section 7.2].

The encapsulating process is defined in [Con98a]. Since it is a part of IPv6 rather than MIPv6, routers and firewalls along the path between MN and HA can be expected to handle the packets correctly, even if they do not implement Mobile IP.

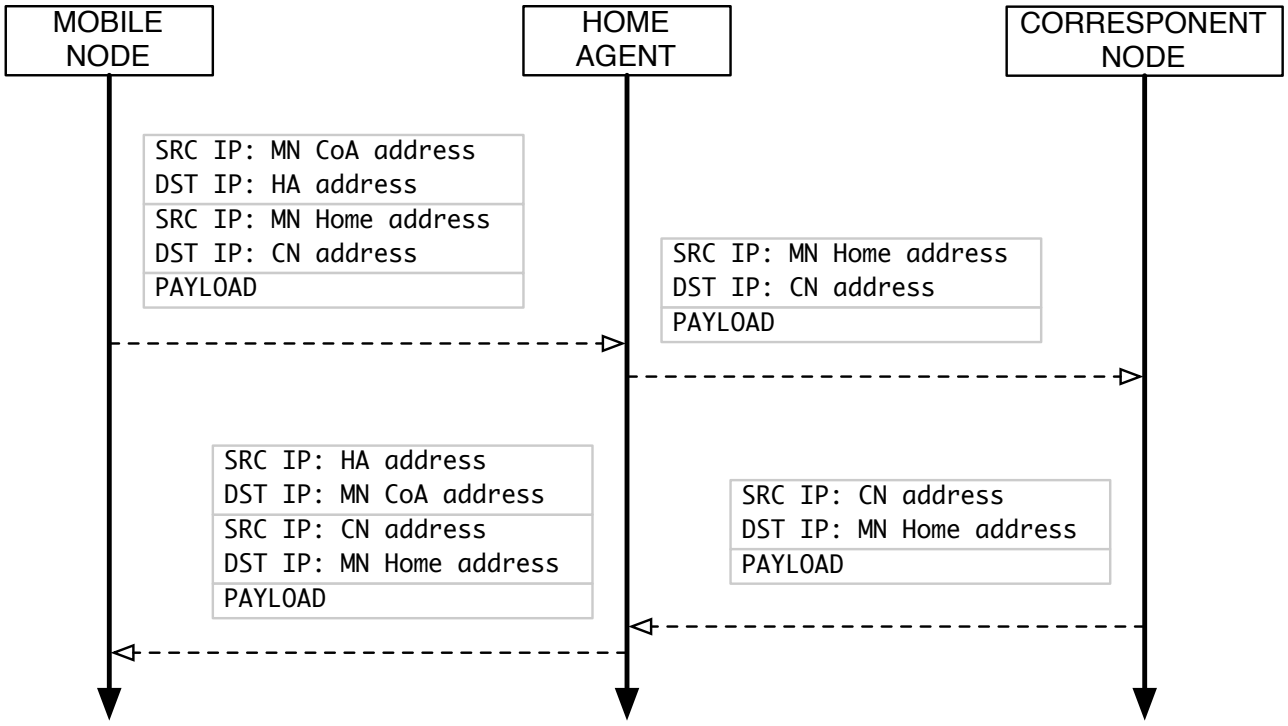

Figure 1.1: Tunneled Communication

1.4.2 Route Optimization

In route-optimized mode, the Mobile Node sends packets directly to the Correspondent Node using its Care-of Address as the IP source address. A Destination Header containing the Home Address is included in the IPv6 header chain. The Correspondent Node sends packets directly to the Care-of Address of the Mobile Node. A Routing Header containing the Home Address is included in the IPv6 header chain.

Since only the Care-of Address of the Mobile Node is used in the IP header, the packets are routed to and from the MN just as any regular IP packet would. Therefore, no extra routing latency is imposed on the packets. Also, no extra traffic is generated on the backbones or to the Home Agent.

In order for route optimization to be used, the CN has to implement the MIPv6 protocol. The Home Address received in the Destination Header allows the CN's network layer to handle the received packet as if was sent directly from the Home Address. Thus, it is possible for the CN's network layer to present all communication with the MN as regular IP communication with the MN's Home Address to the transport layer.

The Type-2 Routing Header and the Home Address Destination Option Header were defined in [Joh04, section 6]. The new type of routing header was designed in order to allow routers and

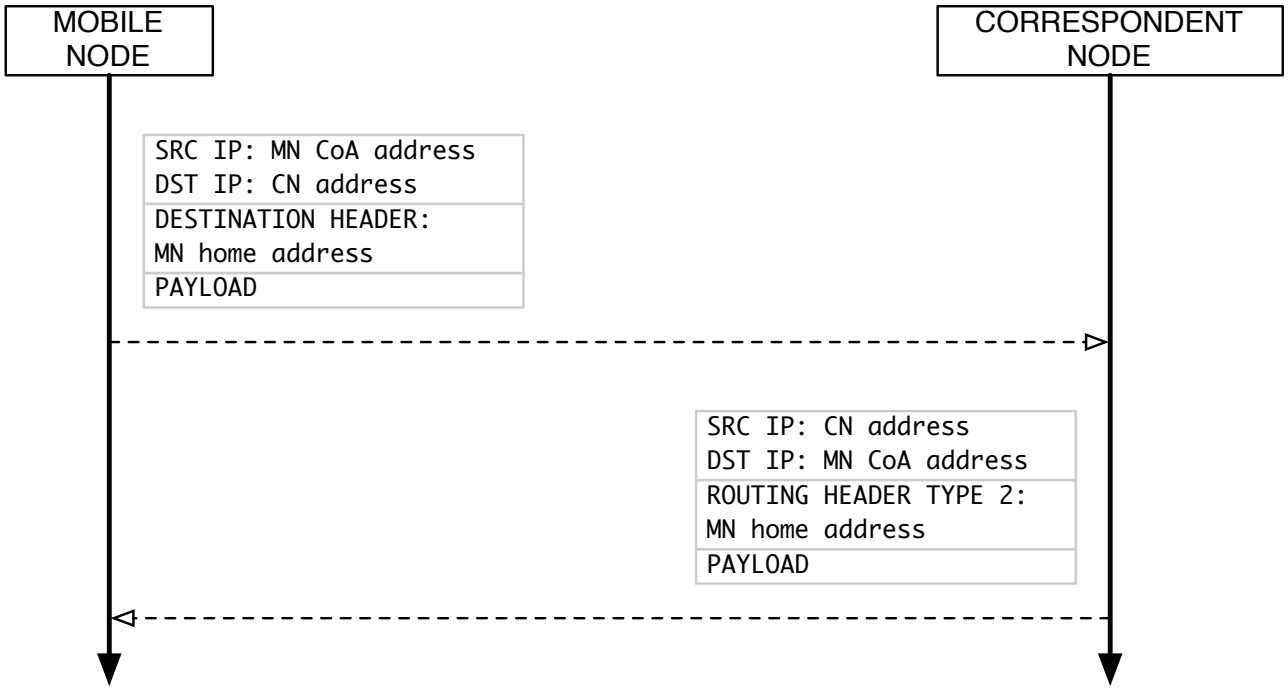

Figure 1.2: Optimized Route Communication

firewalls along the path of the packet to distinguish between source routed traffic and mobile traffic.

1.5 Determination of Location

Before any communication can take place, a Mobile Node has to determine its current location on the network. This is done with the help of the Router Discovery protocol which is part of IPv6. A Home Agent or Foreign Router sends unsolicited Router Advertisements on the Home Link in regular intervals. When a Mobile Node connects to a link, it may also issue a Router Solicitation to request the sending of a RA message. The MN inspects the Prefix Option contained in the RA. If the prefix matches the prefix of the MN's Home Address, the Mobile Node is on its Home Link and can thus communicate with all nodes on the network by regular IP routing. If the advertised prefix does not match the prefix of the Home Address, the Mobile Node is on a Foreign Link. It then has to form a globally routable Care-of Address using the IPv6 auto-configuration mechanism.

1.6 Home Registration

Once the Mobile Node on a Foreign Link has formed a Care-of Address, it has to register this CoA with its Home Agent. If the address of the HA is not known from a previous visit to the Home Link and has not been configured statically, it has to be acquired using the Dynamic Home Agent Address Discovery protocol. In DHAAD, the MN sends a DHAAD Request to the anycast address for Home Agents on its Home Link. A HA responsible for Mobile Nodes with the Home Link's prefix answers the request by sending its address in a DHAAD Reply message. After the address of the Home Agent is known to the Mobile Node, it sends a Binding Update message to the HA. The HA answers with a Binding Acknowledgement message and stores the current CoA of the MN in its binding cache. After the Home Agent has acknowledged the current CoA of the Mobile Node, all packets destined for the MN will be tunneled to the CoA by the Home Agent.

It would be easy for a malicious attacker to send a spoofed Binding Update to the Home Agent, thus convincing it to send all traffic destined for the Mobile Node to the attacker. Because of this, all Binding Updates and Binding Acknowledgements exchanged between MN and HA have to use IPsec. This way, the integrity and authenticity of the registration is guaranteed. Because the Mobile Node and Home Agent are administered by the same person or organization, it is possible to use pre-shared secrets or certificates for IPsec. Therefore, no special authentication procedure is needed.

1.7 Return Routability Procedure

When a Mobile Node wants to send packets to a Correspondent Node for the first time, it has to determine whether route optimization can be used with this CN. Because of the lower impact on latency and overall network traffic, route optimization is always preferred to tunneling. In order to use route optimization, the CN has to implement the MIPv6 protocol itself. Therefore, the Mobile Node has to ascertain the CN's mobility awareness before routing packets directly to it. Because it would be easy for a malicious attacker to convince the CN to send traffic to the attacker instead of the MN, the Correspondent Node has to verify that the given Care-Of Address is indeed the current location of the Mobile Node with the given Home Address. To accomplish this, the Return Routability procedure is employed.

The Mobile Node initiates the Return Routability procedure by sending a Home Test Init message and a Care-of Test Init message. The Home Test Init message is tunneled via the Home Agent and thus received by the Correspondent Node with the Home Address as source.

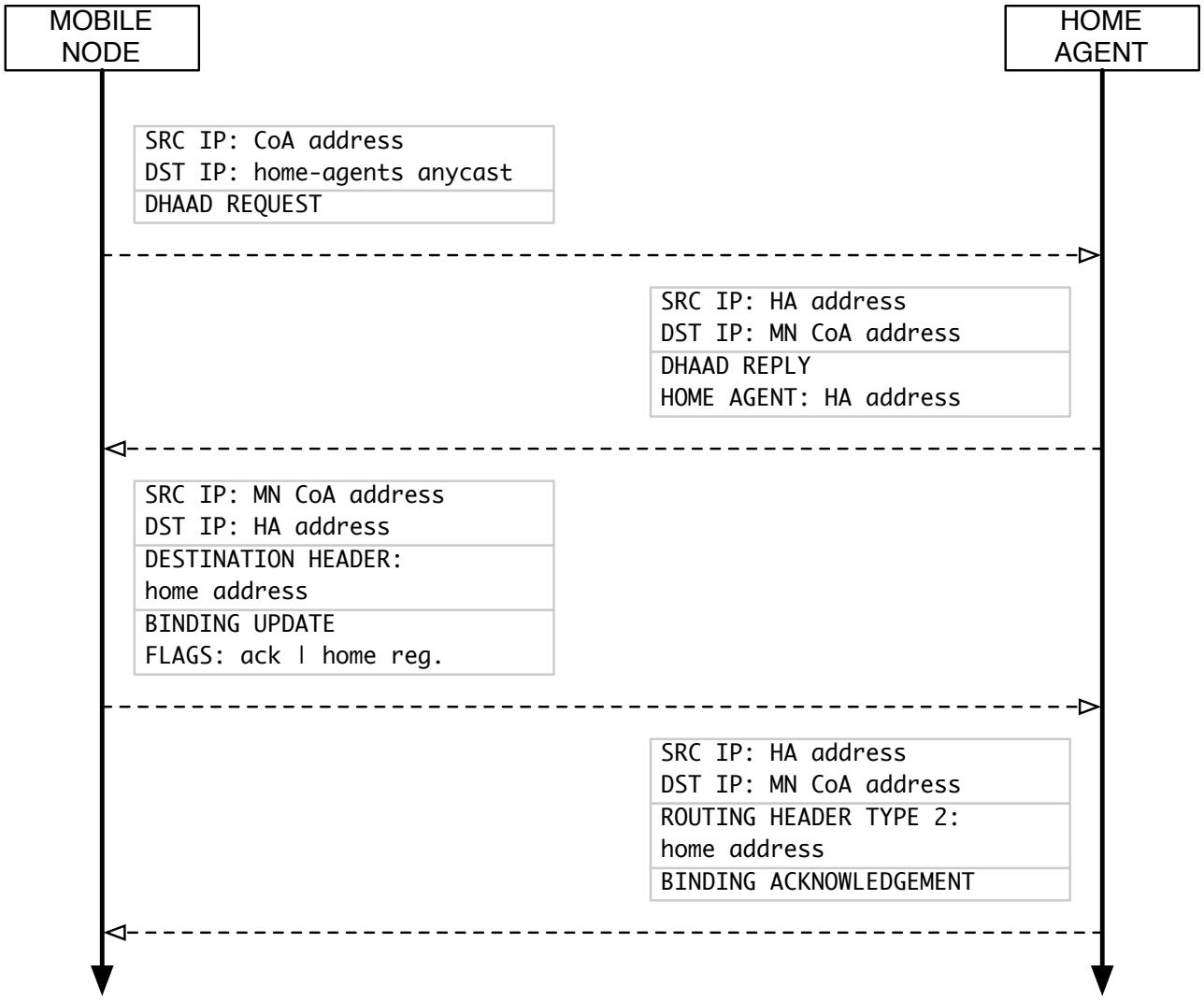

Figure 1.3: Home Agent Registration

The Care-of Test Init message is sent directly to the CN with the Care-of Address as source.

If the Correspondent Node does not implement the MIPv6 protocol, it will respond to the Test Init messages with a *Parameter Problem* ICMPv6 message. In this case, all communication is done via the Home Agent. If the CN does implement the MIPv6 protocol,it sends a Home Test message to the Home Address and a Care-of Test message to the Care-of Address. The Home Test message is tunneled to the MN by the Home Agent. The Correspondent Node keeps an indexed list of generated random numbers, called *nonces*. It forms two cryptographic tokens, *home token* and *care-of token*. The home token is formed by hashing the Home Address concatenated with a random nonce. It is sent to the Mobile Node in the Home Test message, along with the index of the nonce used in the generation. The care-of token is formed by hashing the Care-of Address concatenated with another random nonce. It is sent to the Mobile

Node in the Care-of Test message, along with the index of the nonce.

The Mobile Node receives both tokens, one via its Home Address, the other via its Care-of Address. It forms a cryptographic token called *binding key* by hashing over the home token concatenated with the care-of token. This binding key is sent back to the Correspondent Node in a Binding Update message, together with the nonce indices from the Test messages. Since these indices only have meaning in the internal structures of the Correspondent Node, an attacker gains no knowledge by intercepting them. They are provided so that a CN does not have to maintain state for a Mobile Node with which the binding procedure still has to be completed.

When the CN receives the Binding Update, it recomputes the home token and care-of token using the nonce indices supplied. It then proceeds to compute the binding key itself. If the computed binding key is the same as the binding key received from the Mobile Node, the Correspondent Node can be certain that the Care-of Address is indeed the current location supplied in the Binding Update. This is because one half of the input parameters to the binding key was sent to the Home Address, while the other half was sent to the Care-of Address. Only a node receiving both halves can form the correct binding key. By definition, this is the Mobile Node.

The advantage of this procedure lies in the fact that no pre-shared secret between CN and MN is necessary, thus allowing for unscheduled communication. It should be noted that an attacker located on the same link as the MN could also intercept both tokens as long as the communication between HA and CN is not encrypted. However, such an attacker would gain nothing by doing so, since he can intercept all traffic between MN and CN already. On the other hand, an attacker located on the same link as the CN could easily divert traffic away from the MN, either to itself or to some other network location. This is acceptable because such an attacker could achieve the same results with other techniques as ND or RD spoofing.

1.8 Binding Lifetime

A Home Agent or Correspondent Node cannot depend on the Mobile Node to inform it prior to changing location. For example, the MN might not be connected to the network while traveling between locations. When roaming between wireless links, the Mobile Node might not know that it will change its network location until after this has happened. Because of this, every binding entry in the binding cache of a HA or CN is only valid for a short amount of time. The minimum time a binding may be valid is four seconds, since this is the smallest value that can be expressed in a Binding Update message. The largest value that can be expressed is $4 * 2^{16}$

seconds, which amounts to about three days. When the timeout is reached, the binding will be deleted from the cache. It is the responsibility of the Mobile Node to refresh its bindings before they expire. This is done by sending a Binding Update message to the Correspondent Node or Home Agent, thus resetting the timeout counter associated with the binding entry. Because the Mobile Node may not know that further communication with a certain CN should take place, the CN can request the MN to update the binding. This is done by sending a Binding Refresh Request to the MN's Home Address. The Home Agent tunnels the request to the MN, which sends a Binding Update to the CN, thus resetting the timeout counter.

Because Binding Updates are designed as an IPv6 Mobility Extension header as opposed to a ICMPv6 message, it is possible to piggyback a Binding Update onto a packet carrying payload. Therefore, the amount of traffic overhead needed by Mobile IP is reduced to only 12 octets (the size of the Binding Update Mobility Header) every four seconds. (This is of course assuming that such a fine granularity is needed, a binding's validity might as well be in the range of minutes, making the overhead negligible.)

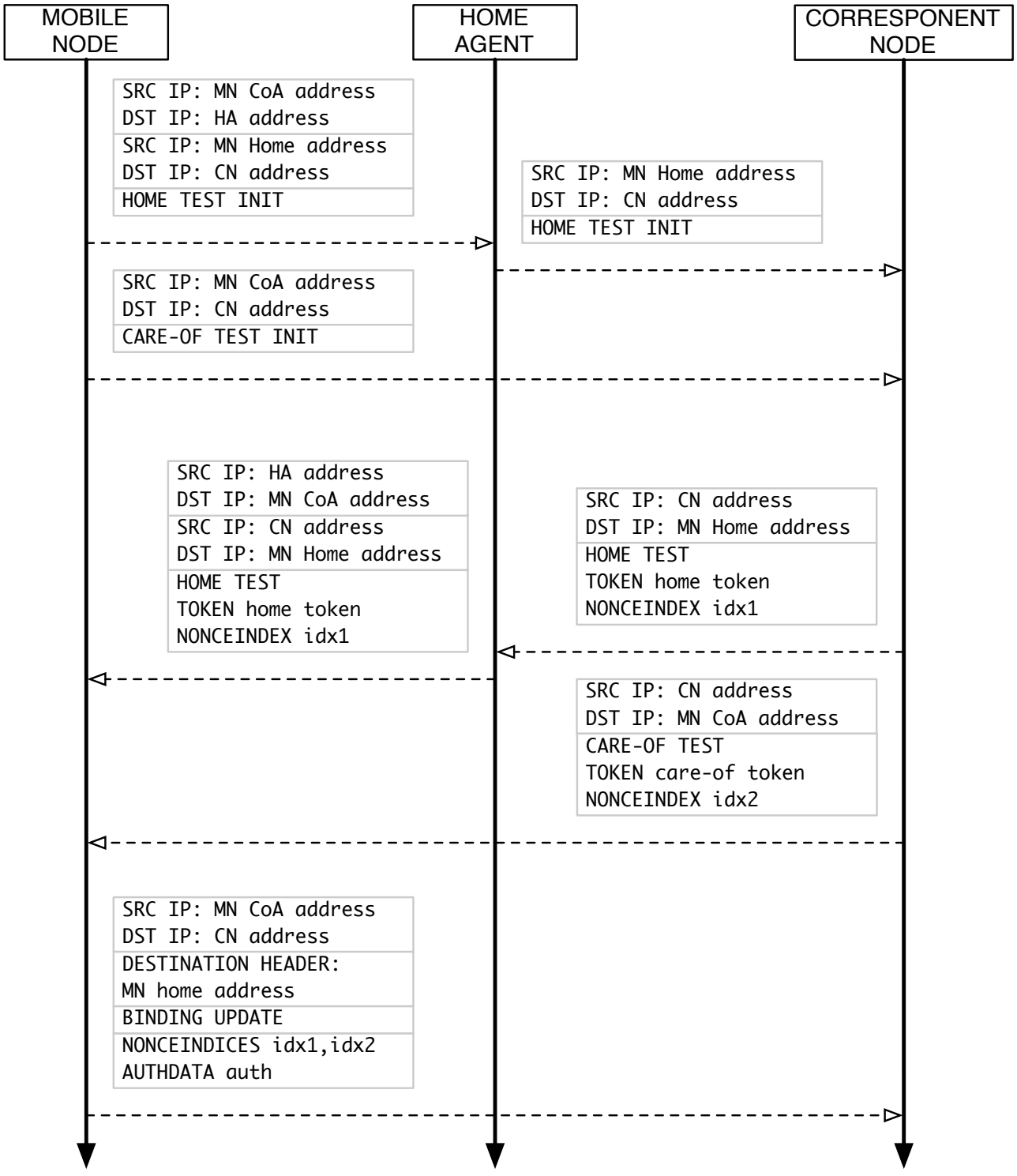

Figure 1.4: Return Routability Procedure

Chapter 2

KAME IPv6 Stack

Kame is the Japanese word for turtle.

The KAME IPv6 stack is an implementation of the IPv6 protocol for the BSD operating system family. It was first created in the year 2000 by the KAME project[1]. The KAME project is a joint effort of the Japanese Keio University and the companies Fujitsu Limited, Hitachi Ltd, Internet Initiative Japan Incorporated, NEC Corporation, Toshiba Corporation and Yokogawa Electric Corporation. The goal of the KAME project is to provide a single IPv6 software set common to BSD operating system variants. The KAME IPv6 stack is available for -amongst others- the OpenBSD, FreeBSD and NetBSD open source operating systems. It implements IPv6 as specified in [Dee98], [Nar98], [Tho98] and [Con98b].

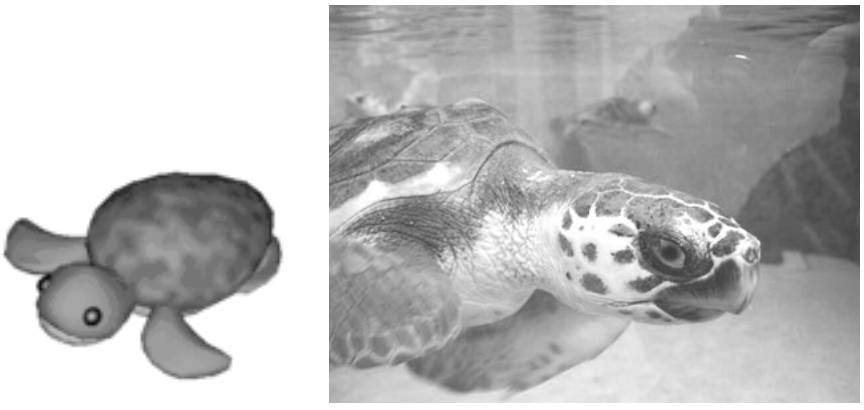

Figure 2.1: KAME logo, Kame

Most of the KAME IPv6 software is part of the operating system's kernel. It is interfaced by

[1] http://www.kame.net

userland software mainly by means of *socket* and related syscalls. Because of this, no libraries are needed for supplying IPv6 services to application software, as all IPv6 specific settings are passed to the kernel as parameters to syscalls. Thus, the software interface already available for UNIX network programming is used. The major exception to this is the introduction of the *getaddrinfo* library function, which extends and replaces the DNS resolver library of the IPv4 network stack.

Specific parts of the kernel's IPv6 functionality can be configured with the help of the *sysctl* command. The most important parameters for KAME IPv6 are:

- `net.inet6.ip6.forwarding`
  When set to 1, this node will act as a router by forwarding packets not destined for itself. When set to 0, this node will act as a host.

- `net.inet6.ip6.accept_rtadv`
  When set to 1, this node will accept Router Advertisements from other nodes. These RAs will be used in building dynamic routing table entries. When set to 0, only static routing entries are used.

- `net.inet6.ip6.auto_linklocal`
  When set to 1, link-local addresses will be assigned to every interface automatically. When set to 0, this has to be done manually.

The KAME IPv6 stack includes two daemons responsible for Router Discovery, as well as a set of command line tools for altering and displaying network preferences.

2.1 rtadvd

The Router Advertisement Daemon *rtadvd* is responsible for sending NDP Router Advertisement messages on all configured interfaces, periodically as well as in response to a Router Solicitation message. To accomplish the latter task, it joins the all-routers link-local multicast address `FF02::2`. The *rtadvd* daemon is called with the list of interfaces on which it is to operate. If the Mobile IP extensions to the Router Discovery protocol are to be used, the *-m* parameter is added:

```
rtadvd -m wi0
```

The daemon can be configured through the file `/usr/local/v6/etc/rtadvd.conf`. This configuration file is in termcap format. Each interface is configured in one line, the different options are

separated by colons[1]. All options allowed in a Router Advertisement message can be given specific values. If no option file is present or an option is not defined in it, an intelligent value constructed from the routing table and the addresses configured on the interface is advertised.

```
wi0: \
        :maxinterval#4 \        #max seconds between unsolicited RAs
        :mininterval#3 \        #min seconds between unsolicited RAs
        :chlim#23 \             #current hop limit
:
```

2.2 rtsold

The Router Solicitation Daemon *rtsold* is the counterpart to *rtadvd*. It is responsible for sending Router Solicitation messages on the configured interfaces. It watches the state of all those interfaces. If an interface is connected or reconnected to a link, *rtsold* will send Router Solicitation messages to the all-routers link-local multicast address. The *rtsold* is called with the list of interfaces on which it is to operate. If the Mobile IP extensions to the Router Discovery protocol are to be used, the *-m* parameter is added:

```
rtsold -m wi0
```

It is also possible to solicit a Router Advertisement message manually by invoking *rtsold* in the form

```
rtsol wi0
```

This causes *rtsold* to only probe once. This means that *rtsol* sends a RS message until a RA response is received, up to six times. If the RA response was received or the sixth RS was sent without a response, *rtsol* returns.

2.3 ifconfig

The command *ifconfig* is used to configure parameters of a network interface. For setting IPv6 addresses, the keyword *inet6* has to be specified. If the length of the prefix of the address is not 64, the keyword *prefixlen* has to specified before the custom length. The syntax to add an IPv6 address is

```
ifconfig rl1 inet6 2005:ffff:feed:face::1 prefixlen 64
```

The syntax to delete an IPv6 address is

[1]It is not clear why the termcap format was chosen, especially since IPv6 addresses contain a lot of colons themselves.

```
ifconfig rl1 inet6 delete 2005:ffff:feed:face::1
```

To configure an anycast address, the keyword *anycast* is used:

```
ifconfig rl1 inet6 2005:ffff:feed:face:fdff:ffff:ffff:fffe anycast
```

To specify that an address should be treated as the home address of a Mobile Node, the keyword *home* is used:

```
ifconfig mip0 inet6 2005:ffff:cafe:babe::b home
```

2.4 route

The *route* command is used to alter the (static) routing tables of a node. The keyword *-inet6* has to be specified to alter the IPv6 routing table. The network prefix is specified by appending a slash followed by the prefix length to the network address. To add a route to an IPv6 network, the syntax is

```
route add -net -inet6 2005:ffff:cafe:babe::/64 2005:ffff:feed:face:1
```

The syntax to delete a route to an IPv6 network is

```
route delete -net -inet6 2005:ffff:cafe:babe::/64
```

To clear the IPv6 routing table, the command is

```
route flush -inet6
```

2.5 ndp

The command *ndp* is used to inspect and alter the address mapping table used by the Neighbor Discovery Protocol (NDP). It thus replaces the *arp* command used with IPv4. To print a list of all NDP entries currently in the table, the parameter used is

```
ndp -a
```

To print the list of default routers acquired via NDP, the parameter used is

```
ndp -r
```

To print a list of all network prefixes acquired via NDP, the parameter used is

```
ndp -p
```

To erase all entries in the address mapping table, the parameter

```
ndp -c
```

is used. The *ndp* command also allows for a more specific manipulation of entries in the NDP table.

2.6 ping6

The *ping6* tool is used for testing basic network layer connectivity between nodes. It has a parameter set similar to the *ping* command for IPv4. To test whether bidirectional IP connectivity exists to another IPv6 node, the command

```
ping6 2005:ffff:feed:face::1
```

is used. A special syntax is necessary when pinging a link-local address on the other node. Since the link-local address prefix is configured on all IPv6 interfaces, the interface to be used has to be specified. This is done by appending a percent sign and the name of the interface:

```
ping6 fe80::202:2eff:fe39:f9%rl1
```

2.7 traceroute6

The *traceroute6* tool is used to examine the path a packet will take to another node. The syntax is straightforward:

```
traceroute6 2005:ffff:feed:face::4
```

2.8 netstat

The *netstat* command is used to get information about the current state of the network stack. The parameter *-finet6* is used to obtain information about the IPv6 protocol family. To get a list of active UDP6 and TCP6 sockets, the command

```
netstat -tau -finet6
```

is used. To print the entries in the IPv6 routing table, the syntax is

```
netstat -r -finet6
```

2.9 sockstat

Similar to *netstat -tau*, the *sockstat* command lists open sockets. It offers additional information about the ID and owner of the process that opened the socket. The sockstat command is a part of the FreeBSD operating system, not of the KAME stack. To get a list of open IPv6

sockets with their corresponding processes, the syntax is

```
sockstat -6
```

2.10 dig

While part of the BIND tools, not of the KAME stack, the *dig* command allows to lookup IPv6 addresses corresponding to hostnames manually by querying a recursive nameserver listed in `/etc/resolv.conf`. The record type for IPv6 addresses is defined as `AAAA` in [Tho95]. To resolve a hostname to its IPv6 address, the syntax to dig is

```
dig AAAA www.kame.net
```

Chapter 3

SHISA MIPv6 Stack

Shisa is a traditional Ryukyuan decoration, often found in pairs, resembling a cross between a lion and a dog. Many people put a pair of Shisa on their rooftops or flanking the gates to their houses. Shisa are believed to protect from various evils.[1]

The SHISA MIPv6 stack is an implementation of the MIPv6 protocol and is included in the KAME stack. It was first created in 2004. At the time of this writing, development of the stack was still in progress, especially with regard to the NEMO Network Mobility functions. The SHISA stack implements MIPv6 as specified in [Joh04] and [Ark04], as well as Network Mobility as specified in [Dev05].

Figure 3.1: SHISA Logo, Shisa

Because Mobile IP is by definition transparent to the transport layer, no application software needs to interface with the stack directly. On a Mobile Node, the SHISA stack offers a new

[1] http://en.wikipedia.org/wiki/Shisa

kind of virtual network interface. The mobile interface *mip0* can be used just like any physical network interface. Networking software that requires mobility support can bind datagram or stream sockets to the home address assigned to the mip0 interface.

With the help of the *sysctl* command, the SHISA part of the kernel can be configured:

- `net.inet6.mip6.use_ipsec`
  When set to 1, IPsec will be used to protect Binding Updates, as described in [Ark04]. When set to 0, no IPsec will be used.

The SHISA stack includes four daemons used to configure the MIPv6 specific settings of the kernel.

3.1 had

The Home Agent Daemon *had* is used to configure the kernel to function as a home agent. It determines the home link by looking at the address of the interface it was called with and passes this information to the kernel. By connecting to port 7778 on localhost, it is possible to view statistics and the binding cache as well as to clear the cache. The Home Agent Daemon is invoked with

```
had wi0
```

3.2 mnd

The Mobile Node Daemon is used to configure the kernel to function as a mobile node. It determines the Home Address of the MN by looking at the home address assigned to the interface it was called with. This information is passed to the kernel. By connecting to port 7778 on localhost, it is possible to view the binding cache, a list of home agents and a list of hosts for which no route optimization should be done. It is also possible to view statistics, as well as to clear the lists and cache. The Mobile Node Daemon is invoked with

```
mnd mip0
```

3.3 cnd

The Correspondent Node Daemon is used to configure the kernel to function as a correspondent node. By connecting to port 7777 on localhost, it is possible to view statistics and to clear the binding cache of the CN. The Correspondent Node Daemon is invoked with

```
cnd
```

3.4 babymdd

The baby Movement Detection Daemon runs on a mobile node. It is responsible for determining the current network location of the node, as well as notifying the kernel about a change of location. This is done by polling the status of the given interface in regular intervals, as well as by watching a routing socket for changes in routing information. The Movement Detection Daemon is invoked with

```
babymdd -h mip0 -p wi0
```

The *baby* prefix of the daemon's name reflects the fact that the current implementation only includes the most basic requirements for a usable movement detection daemon.

Chapter 4

Test Setup

The FreeBSD 5.4 operating system was chosen as a platform for the KAME and SHISA stacks. The decision for FreeBSD was based on the stability and flexibility of this platform, as well as on the fact that a lot of documentation regarding FreeBSD networking is freely available.

Figure 4.1: FreeBSD Logo

4.1 Installing KAME

The 20050919 snapshot of the KAME source code was obtained from the KAME public file server[1]. Prior to compilation, a kernel configuration file `FF.MIP6.KAME` was created based on the `GENERIC.KAME` config file supplied with KAME. All support for hardware not available on the machines was removed. The line `options MIP6` was added to specify Mobile IP support to be built.

[1] ftp://ftp.kame.net

A line `device mip 1` was added to specify support for the mobile interface pseudo device. The KAME stack was then built and installed using the following steps:

First, the KAME source tree was prepared by setting symlinks to the freebsd5 source tree:

```
cd ~/src/kame && make TARGET=freebsd prepare
```

Next, the FreeBSD kernel source was configured from the prepared config file:

```
cd ~/src/kame/freebsd5/sys/i386/conf && /usr/sbin/config FF.MIP6.KAME
```

The kernel was compiled and installed:

```
cd ~/src/kame/freebsd5/sys/i386/compile.FF.MIP6.KAME && (make depend && make && sudo make install)
```

After that, it was necessary to patch the system include files to reflect the changes made in the kernel:

```
cd ~/src/kame/freebsd5/ && make includes && sudo make install-includes
```

Last, the userland daemons and tools, as well as the libraries included in KAME, were built and installed:

```
cd ~/src/kame/freebsd6 && make && sudo make install
```

To load the new kernel, a reboot was necessary:

```
sudo reboot
```

After this procedure, the system was running the KAME stack with the SHISA mobile IP extensions.

4.2 Network Setup

The test internetwork was configured to use IPv6 exclusively. It consisted of three networks. Because the MIPv6 specifications demand that no site-local addresses are used for Home Address and Care-of Address, the network prefix `2005:ffff::/32` was chosen as a prefix common to all three networks[1]. The wireless home network was hosted by the Home Agent. It was assigned the network prefix `2005:ffff:cafe:babe::/64`. This was considered the Home Link of the Mobile Node. The wireless foreign network was hosted by the Foreign Router. It was assigned the network prefix `2005:ffff:c0ca:c01a::/64`. This was considered a Foreign Link for the Mobile Node. The backbone network was formed by connecting the Home Agent, Foreign Router and Correspondent Node by means of a network hub. This was considered the 'arbitrary topology of routers and links' by which Home Link, Foreign Link and Correspondent Node are connected.

[1] This was possible because the test network was not connected to the Internet, so no address ambiguities were introduced.

The IPv6 prefixes and addresses were chosen specifically to make them easily recognizable in network dumps. To avoid dealing with IPv6 addresses on the command line directly, static hostnames were configured in `/etc/hosts`.

```
2005:FFFF:FEED:FACE::1          ichi
2005:FFFF:FEED:FACE::3          san
2005:FFFF:FEED:FACE::4          chi

2005:FFFF:CAFE:BABE::A          heimat
2005:FFFF:CAFE:BABE::B          mobil
2005:FFFF:C0CA:C01A::A          ferne
```

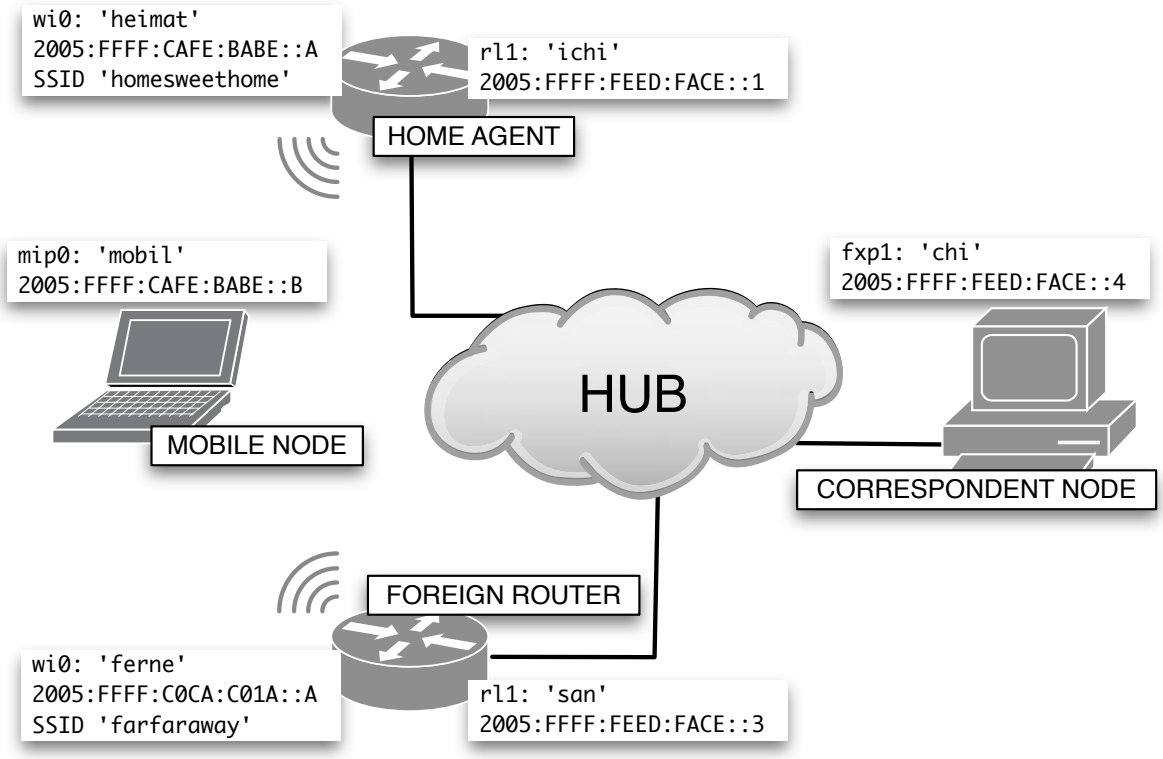

Figure 4.2: Test Network Topology

4.3 Host Configuration

4.3.1 Home Agent

The address 2005:ffff:feed:face::1 was assigned to the interface connecting the Home Agent to the backbone network. A static route to the foreign network was added. The Home Agent was configured to host the wireless home network and to advertise its routes on this link. The unicast address 2005:ffff:cafe:babe::a, as well as the anycast addresses specifying the set of Home Agents and Routers on the subnet, were assigned to the Home Link's interface. The Home Agent was configured to route packets between the two networks it was connected to. The Home Agent Daemon was started to enable the home agent features of MIPv6.

```
sysctl net.inet6.ip6.forwarding=1
sysctl net.inet6.ip6.accept_rtadv=0
sysctl net.inet6.mip6.use_ipsec=0

ifconfig rl1 inet6 ichi
route add -net -inet6 2005:ffff:c0ca:c01a::/64 2005:ffff:feed:face::3

ifconfig wi0 inet6 heimat
ifconfig wi0 media DS/11Mbps mediaopt hostap
ifconfig wi0 channel 1
ifconfig wi0 ssid homesweethome

ifconfig wi0 inet6 2005:ffff:cafe:babe:: prefixlen 64 anycast
ifconfig wi0 inet6 2005:ffff:cafe:babe:fdff:ffff:ffff:fffe prefixlen 64 anycast

rtadvd -m wi0
had -n wi0
```

4.3.2 Foreign Router

The address 2005:ffff:feed:face::3 was assigned to the interface connecting the Foreign Router to the backbone network. A static route to the home network was added. The Foreign Router was configured to host the wireless foreign network and to advertise its routes on this link. The unicast address 2005:ffff:c0ca:c01a::a was assigned to the Foreign Link's interface. The Foreign Router was configured to route packets between the two networks it was connected to.

```
sysctl net.inet6.ip6.forwarding=1
sysctl net.inet6.ip6.accept_rtadv=0
sysctl net.inet6.mip6.use_ipsec=0
```

```
ifconfig rl1 inet6 san
route add -net -inet6 2005:ffff:cafe:babe::/64 2005:ffff:feed:face::1

ifconfig wi0 inet6 ferne
ifconfig wi0 media DS/11Mbps mediaopt hostap
ifconfig wi0 channel 11
ifconfig wi0 ssid farfaraway

rtadvd wi0
```

4.3.3 Correspondent Node

The address 2005:ffff:feed:face::4 was assigned to the interface connecting the Correspondent Node to the backbone network. Static routes to the home and foreign network were configured. Depending on the test parameters, the CN was given mobility awareness by starting the Correspondent Node Daemon.

```
sysctl net.inet6.ip6.forwarding=0
sysctl net.inet6.mip6.use_ipsec=0

ifconfig fxp1 inet6 chi
route add -net -inet6 2005:ffff:cafe:babe::/64 2005:ffff:feed:face::1
route add -net -inet6 2005:ffff:c0ca:c01a::/64 2005:ffff:feed:face::3

cnd
```

4.3.4 Mobile Node

The address 2005:ffff:cafe:babe::b was assigned to the mobile interface as the Mobile Node's Home Address. No further routes or addresses were configured. The MN was configured to solicit Router Advertisements on its wireless interface. The Mobile Node Daemon and Movement Detection Daemon were started to enable the mobility features.

```
sysctl net.inet6.ip6.forwarding=0
sysctl net.inet6.ip6.accept_rtadv=1

ifconfig mip0 inet6 mobil home
ifconfig wi0 up

rtsold -m wi0
mnd mip0
babymdd -h mip0 -p wi0
```

To join the home network, the command

```
ifconfig wi0 ssid homesweethome
```

was used. To join the foreign network, the command used was

```
ifconfig wi0 ssid farfaraway
```

Appendix A

References

[Ark04] Arkko, Devarapalli, Dupont. *Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents*. RFC 3776. 2004

[Con98a] Conta, Deering. *Generic Packet Tunneling in IPv6*. RFC 2473. 1998

[Con98b] Conta, Deering. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)*. RFC 2463. 1998

[Dee98] Deering, Hinden. *Internet Protocol, Version 6 (IPv6) - Specification*. RFC 2460. 1998

[Dev05] Devarapalli, Wakikawa, Petrescu, Thubert. *Network Mobility (NEMO) Basic Support Protocol*. RFC 3963. 2005

[Joh04] Johnson, Perkins, Arkko. *Mobility Support in IPv6*. RFC 3775. 2004

[Nar98] Narten, Nordmark, Simpson. *Neighbor Discovery for IP Version 6*. RFC 2461. 1998

[Per02] Perkins. *Mobility Support for IPv4*. RFC 3344. 2002

[Sol98] Solomon: *Mobile IP - The Internet Unplugged*. Prentice Hall, 1998

[Tho95] Thomson, Huitema. *DNS Extensions to support IP version 6*. RFC 1886. 1995

[Tho98] Thomson, Narten. *IPv6 Stateless Address Autoconfiguration*. RFC 2462. 1998

Appendix B

Glossary

B

BA Binding Acknowledgement.

BU Binding Update.

C

CN Correspondent Node.

CoA Care-of Address.

D

DHAAD Dynamic Home Agent Address Discovery.

F

FR Foreign Router.

H

HA Home Agent.

HoA Home Address.

host A node that does not forward IP packets.

I

ICMP Internet Control Message Protocol, here always referring to ICMPv6.

ICMPv6 Internet Control Message Protocol Version 6.

interface a device by which a node is connected to a link.

IP Internet Protocol, here always referring to IPv6.

IPv4 Internet Protocol Version 4.

IPv6 Internet Protocol Version 6.

L

link a medium shared between two or more nodes over which messages can be transported.

M

MIP Mobile IP, here always referring to MIPv6.

MIPv4 Mobile IP Version 4.

MIPv6 Mobile IP Version 6.

MN Mobile Node.

N

NA Neighbor Advertisment.

ND Neighbor Discovery.

NDP Neighbor Discovery Protocol.

node A device that implements the IP specifications.

NS Neighbor Solicitation.

R

RA Router Advertisment.

RD Router Discovery.

router A node that forwards IP packets which are not addressed to itself.

RRP Return Routability Procedure.

RS Router Solicitation.

V

vio1a a very cute and very nice girl.