Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/ULANC/DS/4.1.2/A1**
Contractual Date of Delivery to the CEC:	December 31st 2002
Actual Date of Delivery to the CEC:	February 11th 2003
Title of Deliverable:	Initial MIPv6 Support Guide
Work package contributing to Deliverable:	WP4
Type of Deliverable*:	R
Deliverable Security Class**:	PU
Editors:	Martin Dunmore, Christopher Edwards
Contributors:	Martin Dunmore, Alun Evans, Patrick Grossetete, Oliver Krämer, Kevin Miles, Reinhard Ruppelt, Gunter Van De Velde

* 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 an initial guide for people wishing to deploy a Mobile IPv6 (MIPv6) testbed at their site. We describe what infrastructure is required to support MIPv6 and detail the installation, configuration and operation of the most suitable implementations.

Keywords:

mobile, IPv6, MIPv6, guide, site, implementation, host, router

Executive Summary

Following on from deliverable D4.1.1 'Survey and Evaluation of Mobile IPv6 Implementations', this document provides an initial guide for people wishing to deploy a Mobile IPv6 (MIPv6) testbed at their site. Although the MIPv6 protocol is still work in progress[1], implementations have been available since 1998. Yet, precisely because the MIPv6 specification is still work in progress, these implementations differ greatly in their supported features and are not likely to be 100% interoperable in most cases. Not until the MIPv6 protocol reaches RFC status in the IETF can we realistically expect fully compliant and interoperable implementations to become available.

This deliverable describes the steps necessary to install, configure and operate the MIPv6 implementations of Cisco, MIPL (Mobile IPv6 for Linux), KAME (FreeBSD), Microsoft and Lancaster University (Linux).

In addition, three case studies of MIPv6 testbed deployment at Sony, FhG and Lancaster University are described.

[1] The current draft is version 20 and has been submitted for IETF last call.

Table of Contents

1 Introduction

This document provides an initial guide for people wishing to deploy a Mobile IPv6 (MIPv6) testbed at their site. Although the MIPv6 protocol is still work in progress, implementations have began to appear since 1998. These implementations differ greatly in their supported features and are not likely to be 100% interoperable in most cases. This is only to be expected since they are at different stages of development and are based on different versions of the draft MIPv6 specification. Not until the MIPv6 protocol reaches RFC status in the IETF can we realistically expect fully compliant and interoperable implementations to become available.

Nevertheless, there are many people within the Internet community who wish to deploy and gain experience of MIPv6 in order to help realise the new mobile applications and services of the next generation Internet. This document describes what infrastructure is required to deploy a general MIPv6 testbed and details the installation, configuration and operation of the most suitable implementations currently available.

Whilst reading this document, it is important to understand that this is a snapshot of available MIPv6 implementations. Information herein may soon become outdated as the MIPv6 protocol develops and new, improved implementations become available.

The rest of this document is structured as follows. The next section provides an overview of the general requirements for deploying a MIPv6 testbed. Sections 3, 4, 5, 6, and 7 detail the installation, configuration and usage of MIPv6 implementations from Cisco, MIPL, KAME, Microsoft, and Lancaster University respectively. Finally, section 8 describes three examples of MIPv6 testbed deployment at Sony, FhG and Lancaster University.

2 General Requirements for a MIPv6 Testbed

A general MIPv6 testbed would consist of:

- at least one Home Agent
- at least one Mobile Node
- at least one Correspondent Node
- several IPv6 networks

At least one of the IPv6 networks must contain a Home Agent. Ideally, another two IPv6 networks should allow for the location of a Mobile Node (when away from home) and a Correspondent Node respectively. A very simple MIPv6 testbed like this is illustrated in Figure 1.

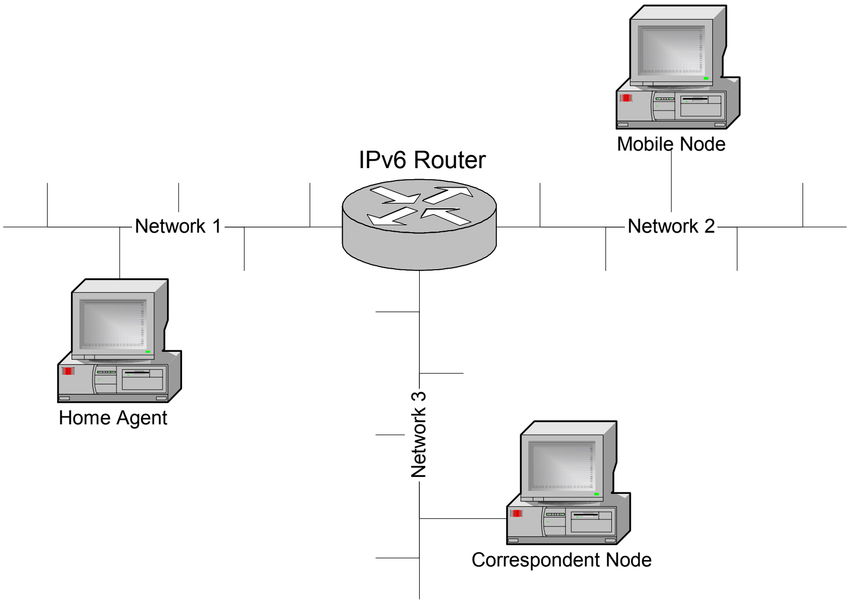


Figure 1 Simple Mobile IPv6 Testbed

The example in the figure shows three networks although, in theory, two networks would suffice (the Correspondent Node could be located in Network 1 or 2).

2.1 Hardware Requirements

In addition to the required network infrastructure (routers, cables, hubs etc.) a MIPv6 testbed will require at least 3 separate machines: the Home Agent, the Mobile Node and the Correspondent Node. One could also use PC-based IPv6 routers (using open source systems like Linux and FreeBSD) instead of commercial IPv6 routers. This will certainly give more flexibility with regard to the addition of new IPv6 features and fine-tuning of network parameters (e.g. router advertisement intervals). Another possibility is to use a PC-based router that also offers Home Agent functionality.

The simplest way to simulate handoffs between networks is to unplug the Ethernet (or other media type) cable to which the MN is currently attached and replace it with a cable from the network you wish to move into. Of course, this means that the Ethernet cables pertaining to the different networks should be within easy reach (e.g. within a network room) if quick handoff latencies are desired.

Alternatively, you may wish to employ Wireless LANs to better facilitate roaming between networks (see case studies in section 8). A simple configuration would be to have each network served by its own wireless access point (AP). In this way (assuming the APs are Wi-Fi certified), one can perform handoffs simply by moving between APs without the need to unplug the WLAN network adapter from the Mobile Node.

2.2 Software Requirements

Before configuring the MIPv6 testbed, you will need to make some policy decisions:

- What kind of global network prefixes you are going to use in your network? 6BONE (3ffe::/16) or production network (2001::/16). Obviously, you will need to obtain suitable IPv6 address space first!
- How are you going to implement the IP routing? In a small network `ripng` would be a good choice. In both, Linux and in FreeBSD, you can use `zebra` (http://www.zebra.org) for accomplishing this.

Since MIPv6 implementations tend to be staggered in terms of supporting draft versions of the MIPv6 specification, interoperability between implementations is somewhat of a hit and miss affair. Therefore, it is wise to first deploy the same MIPv6 implementation on all the machines in the testbed (so make sure that the implementation has HA, MN and CN functionality). Once confidence in one implementation is achieved, you may decide to deploy other implementations and see if they are interoperable.

Note that movement detection algorithms rely on receiving router advertisements from the default IPv6 router when first entering a network. Intelligent movement detection algorithms will also make use of media disconnect/connect notifications and issue router solicitation messages to speed up the reception of these router advertisement messages. Therefore, it is essential that the routers pertaining to the networks in the MIPv6 testbed issue periodic router advertisements so that a MN can configure itself with a new Care of Address when it connects to the network.

The following sections will try to throw some light on how a MIPv6 testbed can be accomplished by using open source software like FreeBSD and Linux in addition to commercial software such as Microsoft Windows 2000/XP and Cisco IOS.

3 Cisco Mobile IPv6

The Cisco MIPv6 implementation began with a research collaboration with Lancaster University and the porting of their MIPv6 implementation for Linux. Since this date, Cisco integrates the MIPv6 Home Agent support in a "Technology Preview" release that is available for experiment.

The feature is not offered yet in a commercial Cisco IOS release as the IETF Mobile IPv6 draft is still evolving. A stable draft version must become available before full inter-operability could be guarantee between the MIPv6 HA and Clients.

The Cisco IOS release of the protocol has been demonstrated at several events, eg. at the Madrid IPv6 Summit, March 2002 and N+I Tokyo, July 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 18 of the IETF draft specification, previous implementation was compliant with ID-13. Based on Cisco IOS 12.2T release train, the Technology Preview works on Cisco 2600, 3600, 3700 and 7200 routers series.

3.1 Available feature set

Home Agent Home agent functionality will allow a suitably configured IPv6 router to act as a home agent for one or more mobile nodes when they are away from home.

Timer changes MIPv6 requires that the range of some timers be changed, for example, the minimum interval between unsolicited router advertisements is reduced from the 3 seconds specified in RFC2461. These relaxations are supported.

Advertisement Interval option MIPv6 requires that routers be configurable to allow them to send an Advertisement Interval option in their Router Advertisements to help mobile nodes perform movement detection. This feature is supported.

Duplicate Address Detection A mobile node may request that a home agent perform Duplicate Address Detection (DAD) when processing its registration and, whilst acting as the mobile node's home agent, defend its address. This feature is supported.

Dynamic Home Agent Address Discovery Home Agents in a subnet learn of each others presence and capabilities by listening to Router Advertisements. A mobile node may obtain the list of home agents on its home subnet by sending a request to the anycast address for all MIPv6-home-agents. This feature is supported.

Access Control List Support for an ACL to control the subnets from which the router will accept Binding Updates and DHAAD requests. This may be used to ‘black list’ certain sub-networks, preventing mobile nodes roaming to them, and refusing to perform correspondent node route optimization with mobile nodes that are currently visiting such sub-networks.

3.2 How to get it

At this date, the software is only available for experiment under a Beta Agreement with Cisco. Request must be sent to Patrick Grossetete, pgrosset@cisco.com.

3.3 Installation

Appropriate Cisco IOS image must be downloaded on the router acting as Mobile IPv6 Home Agent. Minimum memory size must be compliant with the latest Cisco IOS 12.2T release, check CCO software centre to determine the appropriate memory size.

3.4 Configuration

3.4.1 Example Configuration

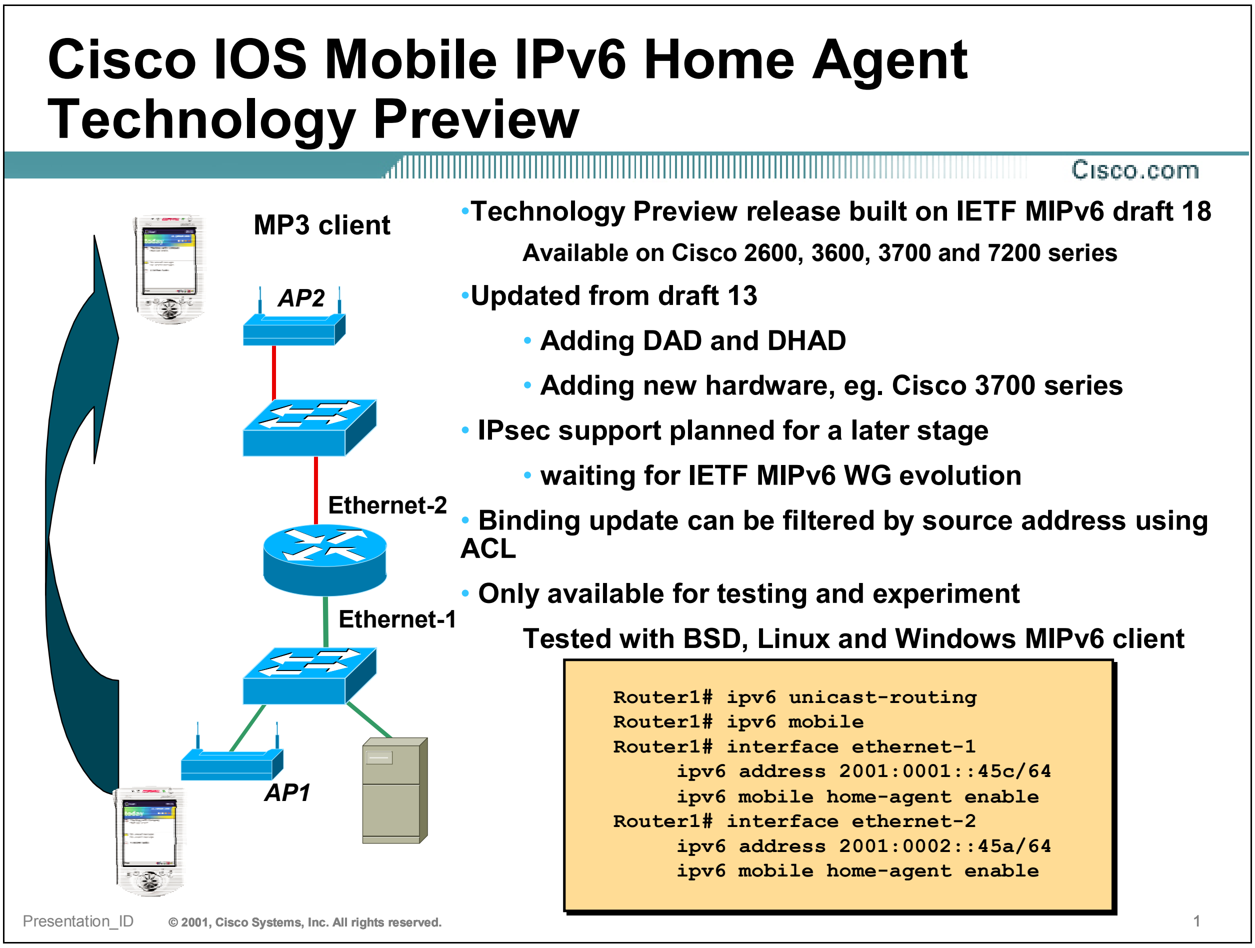


Figure 2 Example Configuration on a Cisco Router

3.4.2 Configuration Commands

Global commands:

```
[no] ipv6 mobile mh-number
```

Changes the number used in the MIPv6 MH.

`[no] ipv6 mobile binding maximum`

Specifies the maximum number of registration bindings which may be maintained concurrently. By default, binding maximum is unset indicating unlimited. If the configured number of home agent registrations is reached or exceeded, subsequent registrations will be refused with the error "Insufficient resources". No existing bindings will discarded until their lifetime has expired, even if binding maximum is set to a value below the current number of such bindings.

`[no] ipv6 mobile binding refresh`

Default is 5 minutes (300 seconds).

Interface commands:

`[no] ipv6 mobile home-agent { create | run }`

Enables home agent operation on the interface. By default, home agent operation is disabled.

`create` is used to initialize the home agent feature on the interface, but does not start operation. Interface level parameters may be configured before operation is commenced.

`run` causes home agent operation to commence on the interface. Interface level parameters may be configured whilst the home agent is in operation.

`no ? run` stops home agent operation on the interface for new registrations. Existing bindings will be allowed to expire (until a `no ? create` command is executed).

`no ? create` immediately discards bindings and any parameter settings, and terminates home agent operation on the interface.

`[no] ipv6 mobile home-agent access`

Configures a binding update filter using an ACL. When an ACL is configured, all received binding updates are filtered. This feature may be used to deny home agent services to mobile nodes that have roamed to particular sub-networks. When the filter blocks a binding update, a binding acknowledgement is returned with error status "Administratively prohibited". Default is no filter; all binding updates are accepted. Note that the filter is also applied to Home Agent Address Discovery messages. When blocked, these are silently discarded.

In configuration of the ACL, the src is the CoA and the dst is the HoA.

`[no] ipv6 mobile home-agent preference`

Specifies the value to be use for Preference in the Home Agent Information Option transmitted on the interface. A value in the range -32768 to +32767 may be specified. By default, a value for Preference of zero is assumed for home agent operation on this interface.

`[no] ipv6 nd ra-interval  [msec]`

Specifies the interval between sending unsolicited multicast Router Advertisements on this interface. This command already exists, but the optional sufffix has been introduced to indicate that the interval has been specified in milliseconds, rather than the default of seconds. This allows specification of the new minimum value of 0.05 seconds. The interval should be set to a low value on interfaces providing service to visiting mobile nodes.

`[no] ipv6 nd advertise-interval`

Specifies whether an Advertisement Interval option should be transmitted in Router Advertisements. This option may be used to indicate to a visiting mobile node how frequently it may expect to receive RAs. It may use this information in its movement detection algorithm.

```
[no] ipv6 nd prefix
      | default [ [ ]
            | [at  ] [off-link]
                  [no-rtr-address] [no-autoconfig] ]
```

This command already exists and is modified to support the `no-rtr-address` option. 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; do not set R bit.

Show commands:

```
show ipv6 interface
```

Existing command; output extended to include home agent data where and when applicable.

```
show ipv6 mobile binding [home-address ]
                         [care-of-address ]
                         [interface ]
```

Displays details of all bindings which match all the search criteria. If no parameters are specified, all bindings are listed. The output is in a form where parts may be cut/pasted into subsequent commands, e.g., `clear ipv6 mobile binding`.

```
show ipv6 mobile globals
```

Displays the values of all global configuration parameters associated with MIPv6, and lists the interfaces on which home agent functionality is currently operating.

```
show ipv6 mobile traffic
```

Displays counters and other information associated with MIPv6.

```
show ipv6 mobile home-agents [ []]
```

Displays the Home Agents List for the specified interface or, if none is specified, displays the Home Agents List for each interface on which the router is acting as a home agent.

Clear commands:

```
clear ipv6 mobile binding [home-address ]
                          [care-of-address ]
                          [interface ]
```

Clears all bindings with the mobile nodes which match the search criteria. Before the bindings are actually cleared, the user will be prompted with the approximate number of bindings that will be deleted and requested to confirm his action. E.g.,

```
router# clear ipv6 mobile binding
Clear 27 bindings [confirm]
```

Note that when this command is used to delete bindings, the mobile node will not be informed that its home agent is no longer acting on its behalf.

`clear ipv6 mobile home-agent`

Clears the Home Agents List on the specified interface. It will be subsequently reconstructed from received Router Advertisements.

`clear ipv6 mobile traffic`

Zeros counters associated with MIPv6.

Debug commands:

`[un] debug ipv6 nd`

Existing command; output modified to include relevant home agent data.

```
[un] debug ipv6 mobile {home-agent | registration
                      | correspondent-node | forwarding}
```

Best to turn all on currently. Open to suggestions.

3.5 Operation

Mobile IPv6 clients must be compliant with ID-18. Previous release offers inter-operability with ID-13.

4 Mobile IPv6 for Linux

Mobile IPv6 for Linux (MIPL) is an implementation 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, has been released under GNU GPL and is freely available to anyone.

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)

4.1 How to get it

In Linux, the standard IPv6 stack is included in the mainstream kernel distribution. However, you have to patch the kernel if you want Mobile IPv6 features. Therefore, a MIPL release is usually compatible only with a specific kernel revision (unless between Linux revisions the kernel code that has to be patched has not changed). MIPL v.0.9.4, which is described in this document, requires Linux kernel version 2.4.19. A short test showed that trying to apply the patch to kernel 2.4.20 did not succeed (at least not with the standard settings of the "patch" command). MIPL 0.9.4 is based on draft version 15 of the MIPv6 specification [9]. The most recent MIPL release, MIPL 0.9.5 is based on draft version 19 [14]. This has not yet been tested within the 6NET consortium, although future updates of this document will address this.

The kernel 2.4.19 can be obtained from your favourite kernel mirror site, e.g. http://www.kernel.org

The MIPL patch mipv6-0.9.4-v2.4.19.tar.gz can be obtained from http://www.mipl.mediapoli.com

4.2 Installation

The following steps describe the MIPL installation (based on the README and INSTALL files of the package).

1. Unpack the MIPL tar file, for example in the `/usr/src` directory. This will create the directory `/usr/src/mipv6-0.9.4-v2.4.19`
2. Go to the Linux source directory (e.g. `/usr/src/linux`)
3. Apply the MIPL kernel patch:
   "patch -p1 < /usr/src/mipv6-0.9.4-v2.4.19/mipv6-0.9.4-v2.4.19.patch"
4. "`make xconfig`" (or config or whatever you prefer) to configure the kernel: under the network settings,

   ```
   CONFIG_IPV6=m
   CONFIG_IPV6_IPV6_TUNNEL=m
   CONFIG_IPV6_MOBILITY=m
   CONFIG_IPV6_MOBILITY_AH=y
   CONFIG_IPV6_MOBILITY_DEBUG=y
   ```

   The last two settings are optional. `CONFIG_IPV6_MOBILITY_AH` enables support for the Authentication Header and `CONFIG_IPV6_MOBILITY_DEBUG` turns on debugging messages.
5. Compile the kernel and modules ("`make bzImage modules modules_install`")
6. Build the user space tools of MIPL

- go to the "`userspace`" sub-directory of the MIPL directory
- "`./configure`"
- "`make install`"

  This might result in an compiler error message that the ipv6_tunnel.h file cannot be found, which seems to be a bug in the MIPL installation. To fix the problem, create a symbolic link in the `/usr/include/linux` directory to the header file in the kernel include directory:
  "`ln -s /usr/src/linux/include/linux/ipv6_tunnel.h .`" (when in `/usr/include/linux`)

- the "`mipdiag`" tool should be available now, which allows configuration and status query.

7. If your distribution does not provide a router advertisement daemon, get one from http://v6web.litech.org/radvd/ and compile it.
8. Start MIPL with:
   `/etc/init.d/mobile-ip6 start`

4.3 Configuration

The configuration files for MIPL are as follows:

`/etc/network-mip6.conf`	- Mobile IPv6 features
`/etc/radvd.conf`	- Configuration file for the router advertisement daemon
`/etc/modules`	- Configuration file for modules loading

The main configuration file is `/etc/sysconfig/network-mip6.conf`. The following table lists the configuration variables that can be set in this file and their meaning (taken mostly from the descriptions in the config file).

Variable name	Description
FUNCTIONALITY	Defines the mode of the node: "ha" – Home Agent "cn" – Correspondent Node "mn" – Mobile Node
DEBUGLEVEL	If debugging was enabled when configuring the MIPL module, this variables controls the verbosity of the output (default: 0)
TUNNEL_SITELOCAL	Should unicasts to node's site-local address be tunneled when mobile node is not in its home network (default: yes)
MIN_TUNNEL_NR	Minimum number of free tunnel devices kept in cache on MN or HA. Must be set to at least 1 for MN and HA. To ensure successful bindings even during high work loads it could be set to a bigger value on the HA.
MAX_TUNNEL_NR	Maximum number of free tunnel devices kept in cache on MN or HA. Must be set to at least 1 for MN and HA. To improve performance set it higher than MIN_TUNNEL_NR.
USE_REVERSE_TUNNEL	Should MN use reverse tunnel for communication with CN (default: no)

HOMEADDRESS	The MN's home address (including prefix length). Only valid in mn mode.
HOMEAGENT	The home agent address (including prefix length). Only valid in mn mode.
MOBILENODEFILE=/etc/mipv6_acl.conf	In HA mode, this file contains the addresses of MNs which are accepted by the HA.
AUTHENTICATION=draft15	Use the new authentication mechanism from draft 15. Otherwise the signalling is authenticated only if you compiled AH support
SAFILE=/etc/mipv6_sas.conf	Configuration file for security associations used for authentication of MIPv6 signalling.
RTR_SOLICITATION_INTERVAL	Mobile nodes are allowed to send Router Solicitation messages at a shorter minimum interval than normal IPv6 nodes (default: 4 seconds). This value controls the mobile nodes minimum Router Solicitation interval in seconds (default: 1).
RTR_SOLICITATION_MAX_SENDTIME=5	After sending MAX_RTR_SOLICITATIONS (defined by IPv6, default: 3) mobile node reduces Router Solicitation send rate with binary exponential back-off mechanism until the maximum send time limit is reached. This option sets the maximum send time in seconds.

For a Home Agent, edit this file and make sure it contains:

`FUNCTIONALITY=ha`

(Other possible modes are "`cn`" for Correspondent Node and "`mn`" for Mobile Node.)

For a Mobile Node, make sure you have the following entries:

`FUNCTIONALITY=mn`
`HOMEADDRESS=`*<mn's home address>*
`HOMEAGENT=`*<ha's address>*

You have to set the MN's home address on the interface used by the MN for roaming.

It is recommended to increase the verbosity level of the kernel module by setting (default=0):

`DEBUGLEVEL=4`

Make sure that the IPv6 kernel module and the mobile_ip6 are loaded in this order at boot time. You can do this for example by adding the following two lines to the `/etc/modules` file:

```
ipv6
mobile_ip6
```

4.4 Installation and Usage Notes/Problems

The ipv6_tunnel.h file is missing from `/usr/src/include linux` (see step 6 of installation). The main issues here are that (1) `usr/include/linux` should have the headers with which the `glibc` used in that distribution was compiled, and (2) there is no standard directory where the includes of the current kernel are installed.

On the Mobile Node, we experienced that after some time (some hours) of operation, the error message "`kernel: route_add failed`" appeared in the kernel log file (`/var/log/messages`). When this happens, the IPv6 routing seems to be in a bad state, and since the IPv6 module cannot be removed while the system is running, the PC needs to be rebooted.

Sometimes the HA sends Binding Nacks on updates from the MN without apparent reason. After restarting MIPL, the Binding Updates are accepted. It is unclear whether this behaviour is related to the fact that the MN registers not only the explicitly stated Home Address (given in the config file), but also additional addresses that are autoconfigured and which have the same network prefix as the home address, or whether a setup problem in our testbed causes this behaviour.

In our testbed, we have not used authentication so far, so we cannot comment on this feature.

Multiple interface support works (we tested for example handovers between Bluetooth and Wavelan access networks with a client switching the MIPL priority of the interfaces), but sometimes when both interfaces receive router advertisements at short intervals (around 1 s), there are spurious switches between the interfaces, though the interface preference is not changed. This might be a problem with how the router advertisement lifetime is handled in the MIPL implementation. If the router advertisements are received at a slightly lower rate (2 s), these interface changes do not occur.

5 KAME / FreeBSD Mobile IPv6

The KAME project developed a Mobile IPv6 implementation for the FreeBSD platform. The code is implemented as part of the kernel. In addition to this two applications for configuration of Mobile IPv6 and extracting statistics have been developed (mip6config and mip6stat). These applications reside in user space.

The implementation follows draft-ietf-mobileip-ipv6-13 [11] with some minor exceptions. The implementation includes the Correspondent Node part (mandatory for an IPv6 implementation that claims to be IPv6 compliant), the Home Agent node part and the Mobile Node part.

5.1 How to get it

Kame is offering an IPv6 stack for BSD-like systems. Among other features, Mobile IPv6 is offered. For having a running Kame IPv6 stack proceed with the following steps:

Get and install *FreeBSD 4.7-RELEASE* (more information in http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/)

Get the latest Kame SNAP from ftp://ftp.kame.net/pub/kame/snap/ or favoured mirror servers.

5.2 Installation

Fine tune the kernel configuration to meet your hardware and be sure to have enabled the mobile-ipv6 features:

```
# mobile-ip6
options          MIP6
options          MIP6_DEBUG
(pseudo-device   hif   1     #MN specific)
```

Compile and install the kernel. This will give you a kernel supporting two new features: Mobile IPv6 Mobile Node and Mobile IPv6 Home Agent. To run a Correspondent Node nothing else has to be prepared.

Compile the user space kame applications. This will also provide you with a Mobile IPv6 suitable (i.e. implementing the extensions mentioned in section 7. Modifications to IPv6 Neighbor Discovery of the draft-ietf-mobileip-ipv6-17.txt) routing advertisement daemon - `rtadvd`. All the new compiled user space kame IPv6 applications are located under `/usr/local/v6`. The two supporting binaries “`mip6config`” and “`mip6stat`” should have been compiled by applying the ordinary kame tree “`make`” procedure.

5.3 Configuration

Configuration files you have to edit:

`/etc/rc.conf` - system wide configuration file
`/etc/rtadvd.conf` - configuration file for the router advertisement daemon.
`/etc/mip6.conf` - configuration file for Mobile IPv6 features

5.3.1 Home Agent

For having the Home Agent functionality on the FreeBSD system you will need to do the following (note that a node cannot act as a Mobile Node and a Home Agent at the same time). Add the following entries in

`/etc/rc.config`:
```
ipv6_ha="YES"
mip6_options="-g -D 1 -T 0"
```

Also make sure you have the following two entries in `/etc/mip6.conf`:
```
debug
enable_ha
```

The Access Router functionality is started at boot time if you add the following entries in the `/etc/rc.config`:
```
rtadvd_enable="YES"
rtadvd_daemon=/usr/local/v6/sbin/rtadvd
rtadvd_flags="-ms -c /etc/rtadvd.conf"
```

Then you will have to adjust the file `/etc/rtadvd.conf` according to your needs.

```
#
# $Id: rtadvd.conf,v 1.2 2001/04/13 21:42:09 cco Exp $
# $Name:  $
#

#	$KAME$
#
# Note: All of the following parameters have default values defined
#	in specifications, and hence you usually do not have to set them
#	by hand unless you need special non-default values.
#
#	You even do not need to create the configuration file. rtadvd
#	would usually work well without a configuration file.
#	See also: rtadvd(8)

# that is advertised on each and every interface
default:\
		:chlim#64:rltime#10:rtime#10:retrans#0:\
		:vltime#30:pltime#10:mtu#1500:

# sent only by simple routers
router:\
	:raflags#0:pinfoflags#192:mininterval#3:maxinterval#5:\
	:tc=default:

# sent only by the Home Agents
```

```
ha:\
        :hatime#100:hapref#10:raflags#32:pinfoflags#224:\
        :mininterval#1:maxinterval#5:tc=default:

# per-interface definitions.
#   Mainly IPv6 prefixes are configured in this part.  However, rtadvd
#   automatically learns appropriate prefixes from the kernel's routing
#   table, and advertises the prefixes, so you don't have to configure
#   this part, either.
#   If you don't want the automatic advertisement, (uncomment and) configure
#   this part by hand, and then invoke rtadvd with the -s option.

# this is a router
xl0:\
        :addrs#1:addr="3ffe:0400:0190:0046:250:4ff:fe64:eb78":prefixlen#64:tc=router:

# this is a HA interface
xl1:\
        :addrs#1:addr="3ffe:0400:0190:0047:210:4bff:feb4:d3a1":prefixlen#64:tc=ha:
```

Figure 3: *Rtadvd sample configuration parameter file*

After rebooting, the FreeBSD box will act as a Home Agent and Access Router. For checking the status of your Home Agent you should use:

`/usr/local/v6/sbin/mip6control`

5.3.2 Mobile Node

Both the Home Address and the Home Agent address have to be explicitly specified. Add the following line to the `mip6.conf` file:

`homeaddr` *<Home addr>*`/`*<plen>*`@`*<interface>*`%`*<Home Agent addr>*

5.3.3 Correspondent Node

To run a CN, no additional measures are required.

5.4 Remarks

Latest snapshot at that time is kame-20021209-freebsd47-snap.tgz of December 9th, 2002 (ftp://ftp.kame.net/pub/kame/snap/).

6 Microsoft Research Mobile IPv6

The Microsoft Research (MSR) Mobile IPv6 implementation was produced in collaboration with Lancaster University as part of the LandMARC project [5]. During the project, Lancaster University's MIPv6 implementation for Linux (see section X) was ported to the Windows 2000 operating system. The implementation was available in executable and source code format as a free download for research purposes from http://research.microsoft.com/mobileipv6/ . This MIPv6 implementation was a modified version of the MSR IPv6 stack, version 1.4 [4].

The public release of the MIPv6 implementation supported version 12 of the IETF MIPv6 draft [10] and provided Mobile Node, Correspondent Node and Home Agent functionality. There was no public release of a MIPv6 implementation for other MS Windows operating systems. However, MIPv6 functionality in Windows XP and CE could be obtained from the Windows 2000 MIPv6 source code by re-compiling with appropriate changes made.

6.1 How to get it

Until recently, the MSR MIPv6 implementation was available for download from the MSR MIPv6 homepage [3]. Unfortunately, Microsoft is no longer making the implementation available for public download. The statement on the homepage reads:

"*Microsoft is keen to provide a supported implementation of Mobile IPv6 in its Windows products. We are working with our partners and customers, and with the IETF to provide this support. At the present time [December 2002], the IETF has not completed the Mobile IPv6 RFC. Microsoft does not therefore include Mobile IPv6 in its Windows products.*

A preview edition of Mobile IPv6 for early versions of Windows 2000, confirming to version 12 of the Mobile IPv6 Internet draft, has been available for research purposes from the Microsoft Research website. This has not been actively maintained against recent releases of the Windows operating system. We do not plan to upgrade this particular implementation because we do not plan to release further releases of IPv6 for Windows 2000. For the best experience with IPv6 from Microsoft, please use Windows XP and the forthcoming Windows .NET Server Family."

However, Windows XP and Windows .NET Server do not yet include MIPv6 functionality although they do have IPv6 functionality built in. Suffice to say, at the time of writing there is no available MIPv6 functionality on a Windows operating system (unless one can obtain a copy of the Windows 2000 preview edition). The following sections are intended as a guide for anyone that has a copy, or is able to obtain a copy, of the Windows 2000 MSR MIPv6 implementation.

6.2 Installation

Before installing the MSR MIPv6 implementation, a working IPv6 stack must already exist on the system. If no IPv6 stack is present, one can be downloaded from [4] and then installed using the following steps:

- Select *Control Panel → Network and Dialup Connections → Local Area Connection*
- Choose *Properties,* go to the *General* tab.
- Click on *Add*.
- Choose *Protocol* and click on *Add*.

- Click on *Have Disk...*. When it asks you for a disk, give it the full pathname of the location where you downloaded the binary distribution.
- IPv6 should now install itself. There is no need to reboot, IPv6 starts immediately once installed.

This install procedure will copy the files from the install kit into the appropriate places and add entries to the system registry for IPv6 configuration. The protocol stack itself (`tcpip6.sys`) is installed in the `%SystemRoot%\system32\drivers` directory. The Winsock helper for the INET6 address family (`wship6.dll`) and all the user applications and utilities (`ipv6.exe`, `ping6.exe`, `tracert6.exe`, `ttcp.exe`, etc) live in the `%SystemRoot%\system32` directory.

Once installed, the MSR MIPv6 binary distribution needs to be added to the relevant network connections in *Control Panel → Network and Dialup Connections → Local Area Connection*.

6.3 Configuration

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 Auto-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 Auto-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 operation. Additional arguments (to those supplied in MSR IPv6 1.4) 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).

The stack will remember mobility parameters, in particular home addresses, in the registry under key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip6\Mobility`. These settings will 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.

6.3.1 MIPv6 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 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.

6.3.2 The MSR MIPv6 Configuration Tool

The MIPv6 Auto-Configuration Service can be controlled using the `MIPv6Conf.exe` utility. This provides a GUI interface for inspecting and manually configuring home addresses. It also allows one to specify parameters dictating the automatic configuration of home addresses by the MIPv6 Auto-Configuration Service, such as the maximum number of home addresses that should be configured.

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).

[1] Home addresses for which the binding updates have been positively acknowledged by the Home Agent within a certain time frame.

Note that enabling or disabling auto-configuration support actually starts or stops the auto-configuration service respectively.

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.

6.4 Operation

Once installed and configured, the MIPv6 functionality appears as an additional virtual interface via the `ipv6.exe` utility. For example:

```
C:\>ipv6 if
Interface 5: Ethernet: Local Area Connection
  Guid {BBAAAD13-A373-482C-BCED-0132170B3D72}
  uses Neighbor Discovery
  uses Router Discovery
  sends Router Advertisements
  forwards packets
  media reconnect flushes stale auto-configured state after 1500ms
  does not heuristically flush stale auto-configured state
  link-layer address: 00-01-02-b6-e8-e2
    preferred global 2001:630:80:7030::2, life infinite (manual)
    preferred link-local fe80::201:2ff:feb6:e8e2, life infinite
    multicast interface-local ff01::1, 1 refs, not reportable
    multicast link-local ff02::1, 1 refs, not reportable
    multicast link-local ff02::1:ffb6:e8e2, 1 refs, last reporter
    multicast link-local ff02::1:ff00:2, 1 refs, last reporter
    multicast interface-local ff01::2, 1 refs, not reportable
    multicast link-local ff02::2, 1 refs, last reporter
    multicast site-local ff05::2, 1 refs, last reporter
  link MTU 1500 (true link MTU 1500)
  current hop limit 128
  reachable time 39000ms (base 30000ms)
  retransmission interval 1000ms
  MaxRtrAdvInterval 600000ms
  MinRtrAdvInterval 200000ms
  DAD transmits 1
```

```
Interface 4: MIPv6 Pseudo-Interface
  Guid {BADE68B3-9FC9-5E9E-6285-D4F8E3E476DD}
  does not use Neighbor Discovery
  does not use Router Discovery
  media reconnect flushes stale auto-configured state after 1500ms
  does not heuristically flush stale auto-configured state
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
  reachable time 23000ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
Interface 3: 6to4 Tunneling Pseudo-Interface
  Guid {A995346E-9F3E-2EDB-47D1-9CC7BA01CD73}
  does not use Neighbor Discovery
  does not use Router Discovery
  media reconnect flushes stale auto-configured state after 1500ms
  does not heuristically flush stale auto-configured state
  routing preference 1
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
  reachable time 26500ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
Interface 2: Automatic Tunneling Pseudo-Interface
  Guid {48FCE3FC-EC30-E50E-F1A7-71172AEEE3AE}
  does not use Neighbor Discovery
  does not use Router Discovery
  media reconnect flushes stale auto-configured state after 1500ms
  does not heuristically flush stale auto-configured state
  routing preference 1
  EUI-64 embedded IPv4 address: 0.0.0.0
  router link-layer address: 0.0.0.0
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
  reachable time 42000ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
Interface 1: Loopback Pseudo-Interface
  Guid {6BD113CC-5EC2-7638-B953-0B889DA72014}
```

```
does not use Neighbor Discovery
does not use Router Discovery
media reconnect flushes stale auto-configured state after 1500ms
does not heuristically flush stale auto-configured state
link-layer address:
  preferred link-local ::1, life infinite
  preferred link-local fe80::1, life infinite
link MTU 1500 (true link MTU 4294967295)
current hop limit 128
reachable time 15000ms (base 30000ms)
retransmission interval 1000ms
DAD transmits 0
```

Experience at Lancaster demonstrated that the implementation works well in accordance with draft version 12 and exhibits relatively stable behaviour. However, experience with handoff testing showed that the implementation has relatively poor handoff latencies in the range of 2 to 30 seconds depending largely on the behaviour of network adapters and frequency of router advertisements.

Limitations of the MSR MIPv6 implementation can be summarised as:

- 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
- erratic behaviour when two hosts move simultaneously.

7 Lancaster University Mobile IPv6 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.

7.1 How to get it

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.

7.2 Installation

Before installing Lancaster University's MIPv6 implementation for Linux, it is necessary to have a functional IPv6 network using Linux routers. A good HOWTO on configuring an IPv6 network using Linux routers is available at

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.

7.2.1 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.

7.2.2 Installing the kernel module

As root, copy `module/mobile_ipv6.o` to:

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

Also, copy `tools/mipv6_config` to:

```
/usr/sbin/mipv6_config
```

7.2.3 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.

7.2.4 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, and are recommended be placed 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.

7.2.5 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.

7.3 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) e.g. for MN and CN functionality type:
>
> `/usr/bin/mipv6_config –mode +CN +MN -HA`

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.

7.4 Notes

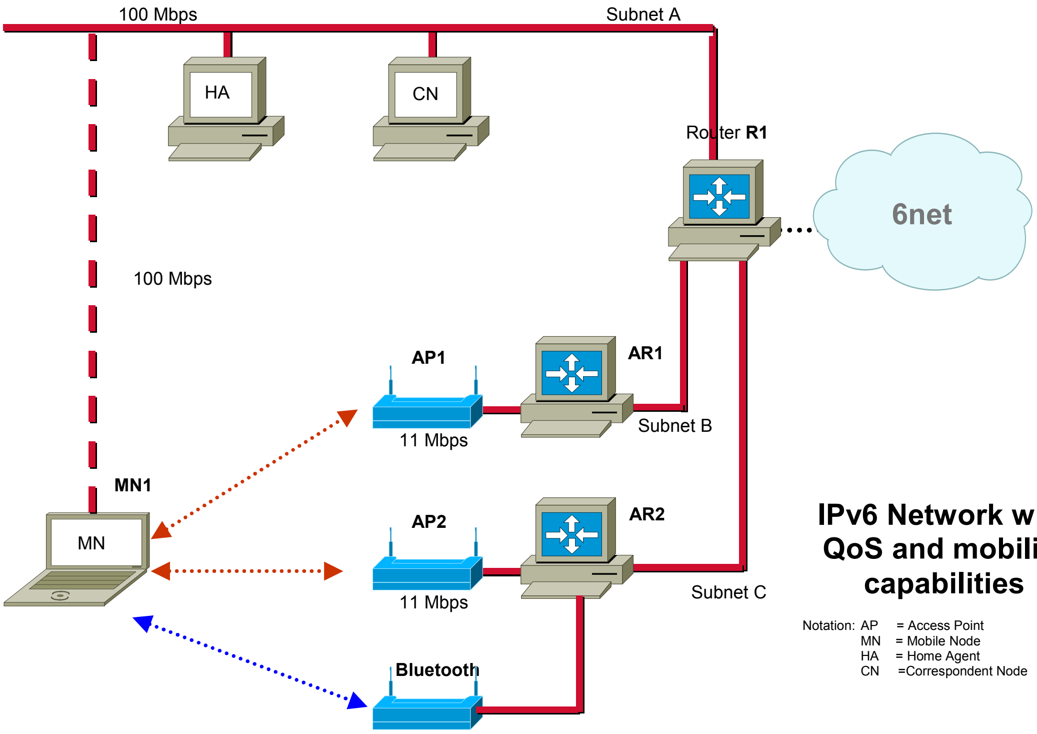
The implementation has been tested in HA, MN and CN modes on Linux PCs running kernel versions 2.4.16 and 2.4.17 and on Sony Vaio and Dell notebooks (running the same Linux kernel versions). The implementation proved to be relatively stable and was tested with UDP, TCP and ICMP. During testing, average handoff latencies observed were between 1 and 3 seconds depending on how rapidly a router would respond to a router solicitation message.

The implementation does not have any support for IPSec nor any AAA functionality. Dynamic Home Agent Discovery is also not available, so the MNs home addresses have to be configured manually.

8 Case Studies

8.1 Sony

All PCs in the Sony testbed run Linux kernel 2.4.19 based on the SuSE 8.0 distribution. One router connects 3 subnets, the "backbone" and 2 subnets which connect access routers providing WaveLAN access to the main router (see figure below).



The MIPv6 movement detection is based on router advertisements, thus in each subnet a RADVD (Router ADVertisment Daemon) needs to be set up to send router advertisements. These contain the IPv6 prefix of the subnet, which allows MIPv6 to detect a network change. A shorter router advertisement interval allows a faster handover when the subnet is switched. A sample RADVD configuration file is shown below (please refer to the man page of radvd for more detailed information):

```
interface eth1
{
        AdvSendAdvert on;
        MaxRtrAdvInterval 3;
        MinRtrAdvInterval 2;
        AdvIntervalOpt on;

        prefix 3ffe:aaaa:bbbb:3::4/64
        {
                AdvOnLink on;
                AdvAutonomous on;
```

```
                AdvRouterAddr on;
        };
};

interface eth2
{
        AdvSendAdvert on;
        MaxRtrAdvInterval 1.5;
        MinRtrAdvInterval 0.5;
        AdvIntervalOpt on;

        prefix 3ffe:aaaa:bbbb:10::4/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };
};
```

The above RADVD configuration file is used for AR2 (Bluetooth and WaveLAN access router in our testbed). The daemon sends out router advertisements on the interfaces via which the Bluetooth and WaveLAN access points are connected at every 0.5 to 1.5 seconds.

8.2 FhG

The Fokus testbed is divided into two sections. A local testbed (**Figure 4**) not constantly connected to the 6NET network serves as an internal development- and test network mainly for MIPv6 test purposes whereas the other part (**Figure 3**) consists of a testbed with additional provision of native IPv6 connectivity to the outer 6NET world.

The latter environment is intended to serve as a platform enabling functionality-, interoperability- and performance testing among 6NET participants.

All machines in the local Fokus testbed run FreeBSD or Linux kernel 2.4.19 based on the Debian distribution [9]. One router connects three subnets providing WaveLAN access (see figure below).

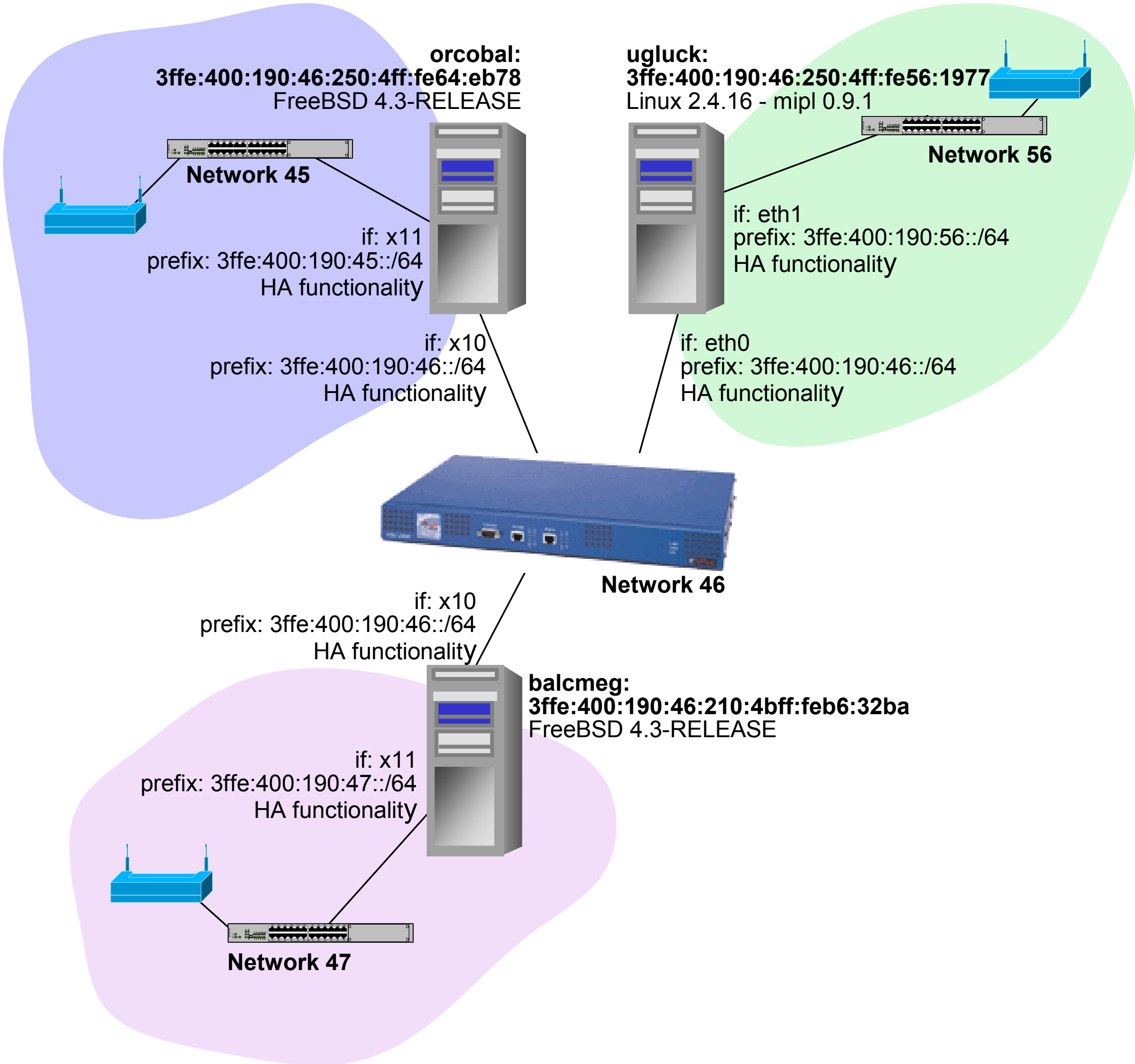


Figure 4*: FhG Fokus Mobile IPv6 testbed*

The MIPv6 movement detection is based on router advertisements, thus in each subnet a `radvd` (router advertisement daemon) needs to be set up to send router advertisements. These contain the IPv6 prefix of the subnet, which allows MIPv6 to detect a network change. A shorter router advertisement interval allows a faster handover when the subnet is switched. A sample `radvd` FreeBSD configuration file is shown below:

The daemon sends out router advertisements at every 1 to 5 seconds.

8.2.1 FOKUS 6NET Test/Demo Network Infrastructure

This section collects information about the Fokus 6NET / IPv6 infrastructure.

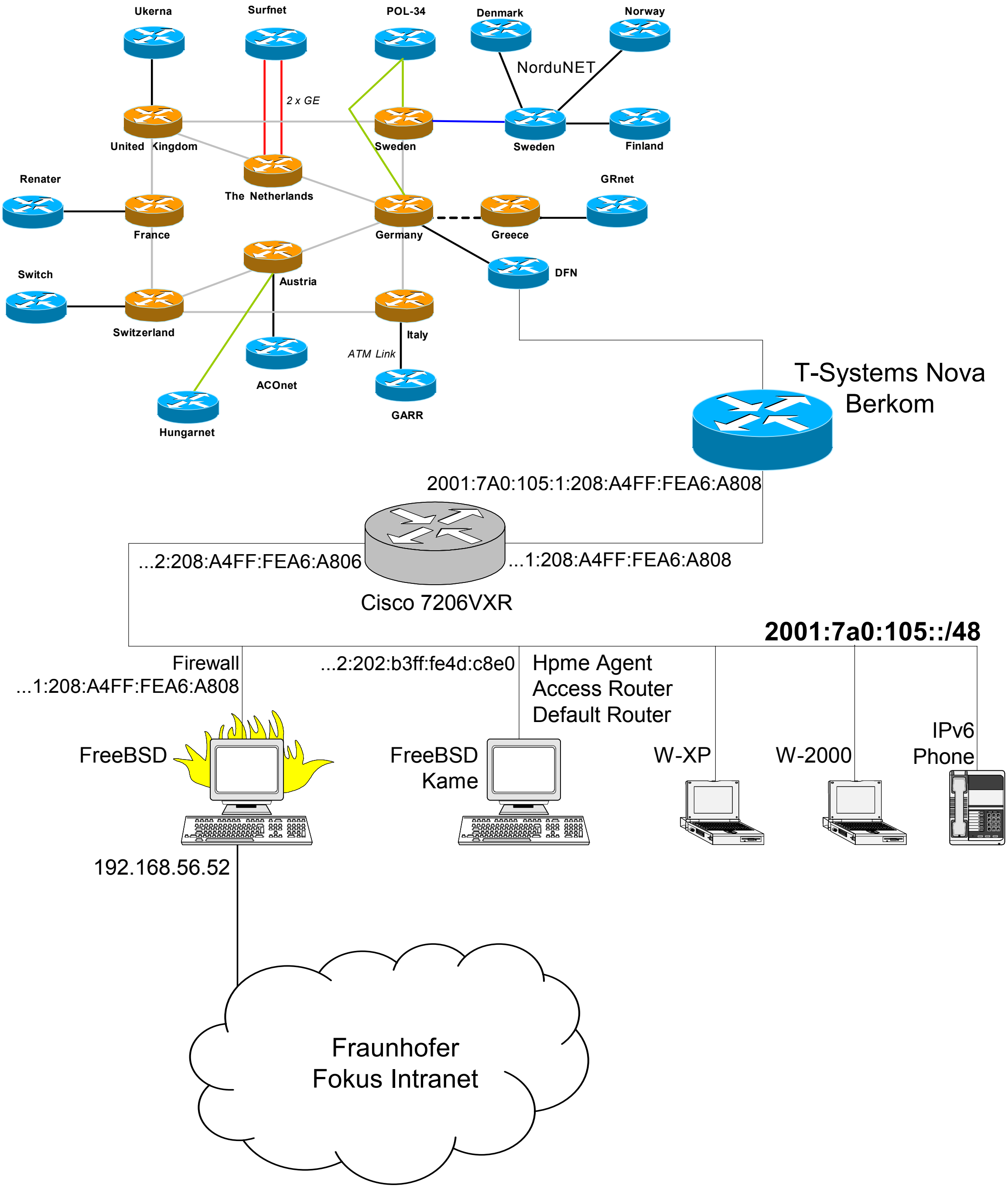


Figure 5: FhG Fokus 6NET testbed

The Fokus testbed is made up of the different local components:

- End systems with different operating systems
  At the leaves of the network, standard PCs with different operating systems (Windows 2000/XP, Linux, FreeBSD) are installed. They are used for testing IPv6 network applications like video conferencing, web surfing, downloading audio and video streams, IP telephony applications etc.
- Home Agents
  Mobile IPv6 Home Agent implementations as offered by the KAME project and Linux based MIPL.

- IP telephone
  BonePhone is a telephony application (developed for 6NET WP5) using SIP for audio data communication negotiation.
  Features:
  - Handle multiple calls at a time.
  - Lets mute/unmute calls as onw wishes by SIP signalling.
  - Supports multiple line profiles from 33.6 Kbit up to 1 Mbit of available bandwidth. Users can even define their own line profiles.
  - Users may choose addresses either manually or from phonebook.
  - Two call rendering types are available now, a simple short one and a more visual one. Users may switch between them at runtime.
  - Automatic call acceptance without asking the user to pick up the phone.
- End systems for remote access
  Remote access to hosts in the local test bed is provided. This offers the possibility to perform distributed tests even if the remote partner is not available. These machines can also be used for troubleshooting purposes.

8.2.2 Testbed components

Host	System	OS	Patch	Type	IP Address
trolloc-1	Cisco 7206VXR [1]	IOS		router	2001:7a0:105:2:208:a4ff:fea6:a806 ...1:208:a4ff:fea6:a808
adrahil	PIII/500 MHz	Debian GNU/Linux Rel. 2.4.18, dual stack	mipv6-0.9.3-v2.4.18	HA	...2:202:b3ff:fe4d:c8e0
				Access router	...2:202:b3ff:fe4d:c8e0
				IPv4 gateway host	192.168.56.52
		FreeBSD 4.6-RELEASE, dual stack	kame-20020624-freebsd46-snap	HA	...2:202:b3ff:fe4d:c8e0
NN				MN	2001:7a0:105:2::/64

Table 1: IPv6 testbed components

[1] Cisco 7206VXR, 6-slot chassis, 1 AC Supply w/IP Software, 7200VXR NPE-400 (128MB default memory), 256 MB Memory for NPE-400 in 7200 Series, 1-Port Packet/SONET OC3c/STM1 Singlemode (IR) Port Adapter, Cisco 7200 Input/Output Controller with Dual 10/100 Ethernet, Cisco 7200 I/O PCMCIA Flash Disk, 128 MB Option, Cisco 72009 Series IOS IP

8.2.3 Offering a Virtual Home Network Service

One can think of a scenario in which the MN is roaming only between foreign networks, never arriving home.

For offering this kind of service one needs a HA connected to the public IPv6 infrastructure. The MN's home address has to have the same prefix as the HA's one.

The testbed offers use of a MIPv6 Home Agent for functionality and interoperability tests. Important configuration parameters for the Mobile Node are:

Home address in the network: 2001:7a0:105:2::/64

Home Agent address: 2001:7a0:105:2:202:b3ff:fe4d:c8e0

There is no security association or any other restriction whatsoever. Any Mobile Node in the network above should be able to use this HA. Later on there will be security association test possibilities as well.

8.3 Lancaster University

As simplified diagram of the Lancaster University MIPv6 testbed is shown in Figure 6. Connectivity to 6NET (via SuperJANET) is achieved through a Cisco 7206 router. The testbed is then divided into Virtual Local Area Networks (VLANs) with each subnet (generally a /64) being its own VLAN. The Extreme Summit Ethernet switch is capable of assigning VLAN tags according to IPv6 prefixes and switching them accordingly (thus, it acts as a virtual IPv6 router).

The testbed employs two different Home Agents. One using the MSR MIPv6 implementation for Windows 2000 and the other using Lancaster University's own MIPv6 implementation for Linux. Mobile and Correspondent Nodes used in the testbed are dual-boot (Windows 2000 and Linux 2.4.16) machines and can therefore use either MIPv6 implementation of the two available operating systems.

Tests at Lancaster have demonstrated that the MSR MIPv6 implementation and Lancaster University's MIPv6 implementation for Linux are interoperable. This is not surprising since one was ported from the other! However, both implementations are somewhat behind the current draft version 19 of the MIPv6 specification. Therefore, anyone looking to deploy the latest MIPv6 implementations for their testbed would be better advised looking at the MIPL and KAME implementations. Furthermore, the MSR MIPv6 implementation is no longer available from MSR's MIPv6 homepage [3].

Future plans at Lancaster involve vertical handoff analysis between GSM/GPRS and 802.11a/b networks. 802.11a hardware is not yet available in UK but will be deployed as soon as the pending UK legislation to allow free public use of the 5Ghz frequency band becomes law. Orange GSM/GPRS access for the testbed has only recently been put in place.

In addition, other MIPv6 implementations such as MIPL, KAME and Cisco (either deployed on the Cisco 7206 or another Cisco 7200 not shown in the figure) will be tested for handoff performance and interoperability. Lancaster University's MIPv6 implementation for Linux will be brought up to date with the MIPv6 specification once it is submitted for RFC status.

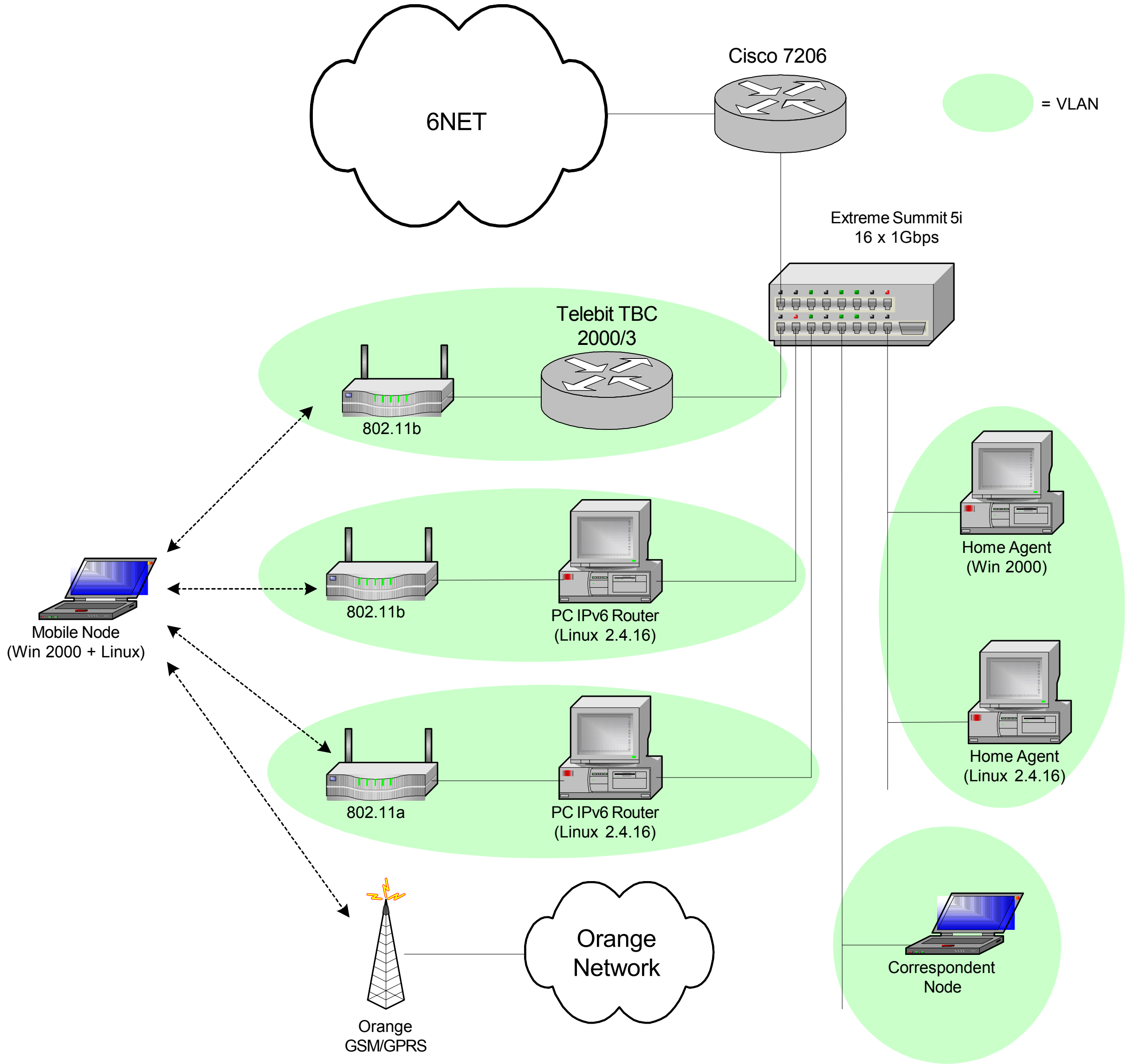


Figure 6 ULANC MIPv6 Testbed

References

[1] KAME homepage, http://www.kame.net
[2] Mobile IPv6 for Linux homepage, http://www.mipl.mediapoli.com/
[3] Microsoft Research Mobile IPv6 homepgae, http://research.microsoft.com/mobileipv6/
[4] Microsoft Research IPv6 homepage, http://research.microsoft.com/msripv6/
[5] The LandMARC homepage http://www.landmarc.net/
[6] Lancaster University MIPv6 package, http://www.cs-ipv6.lancs.ac.uk/ipv6/MobileIP/
[7] Peter Bieringer's Linux IPv6 HOWTO, http://www.bieringer.de/linux/IPv6/
[8] 6WIND homepage, http://www.6wind.com
[9] Debian GNU/Linux homepage, http://www.debian.org/
[10] David B. Johnson, Charles Perkins, "Mobility Support in IPv6" draft-ietf-mobileip-ipv6-12.txt, Internet Draft, April 2000, work in progress
[11] David B. Johnson, Charles Perkins, "Mobility Support in IPv6" draft-ietf-mobileip-ipv6-13.txt, Internet Draft, November 2000, work in progress
[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] David B. Johnson, Charles Perkins, J. Arkko, "Mobility Support in IPv6" draft-ietf-mobileip-ipv6-19.txt, Internet Draft, October 2002, work in progress
[15] The Cisco IPv6 Statement of Direction, available for download via http://www.cisco.com/warp/public/732/Tech/ipv6/ipv6_learnabout.shtml
[16] G. O'Shea, M. Roe, "Child-proof Authentication for MIPv6 (CAM)", Computer Communication Review, Vol. 31, No. 2 (April 2001)
[17] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", IETF RFC 2460, December 1998
[18] C. Perkins, "IP Mobility Support for IPv4 (revised)", IETF RFC 3220, January 2002
[19] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", IETF RFC2461, December 1998