Mobile IP (MIP)

Introduction and Basics of Mobile IP

Mobile IP is intended to enable nodes to move from one IP subnet to another. It is just as suitable for mobility across homogeneous media as it is for mobility across heterogeneous media. That is, Mobile IP facilitates node movement from one Ethernet segment to another as well as it accommodates node movement from an Ethernet segment to a wireless LAN, as long as the mobile node's IP address remains the same after such a movement.

One can think of Mobile IP as solving the "macro" mobility management problem. It is less well suited for more "micro" mobility management applications --for example, handoff amongst wireless transceivers, each of which covers only a very small geographic area. As long as node movement does not occur between points of attachment on different IP subnets, link-layer mechanisms for mobility (i.e., link-layer handoff) may offer faster convergence and far less overhead than Mobile IP.

MIPv4 vs. MIPv6

MIPv4 is an instance of Mobile IP that provides mobility support for IPv4. However, we will not use this protocol in our architecture; we will use MIPv6 instead because it offers better performance and quality of service than MIPv4.

In MIPv4, The foreign agent detunnels and delivers datagrams to the mobile node that were tunneled by the mobile node's home agent. For datagrams sent by a mobile node, the foreign agent may serve as a default router for registered mobile nodes (RFC 3344). The fact that all data packets sent to the mobile node are tunneled through the home agent adds significant delays to data packets received by the mobile node and is a very inefficient method for data delivery.

MIPv6 (RFC 3775) solves this problem as data packets directed to the mobile node are sent directly from the correspondent node to the mobile node. This greatly reduces the delay of received packets at the mobile node and eliminates inefficient tunneling of data.
MIPv6

Mobile Internet Protocol version 6 (IPv6) allows an IPv6 node to be mobile—to arbitrarily change its location on an IPv6 network—and still maintain reachability. Connection maintenance for mobile nodes is not done by modifying Transport layer protocols, but by handling the change of addresses at the Internet layer using Mobile IPv6 messages, options, and processes that ensure the correct delivery of data regardless of the mobile node's location.

Mobile IPv6 allows an IPv6 node to be mobile—to arbitrarily change its location on an IPv6 network—and still maintain existing connections. When an IPv6 node changes its location, it might also change its link. When an IPv6 node changes its link, its IPv6 address might also change in order to maintain connectivity. There are mechanisms to allow for the change in addresses when moving to a different link, such as stateful and stateless address autoconfiguration for IPv6. However, when the address changes, the existing connections of the mobile node, which are using the address assigned from the previously connected link, cannot be maintained and are ungracefully terminated.

The key benefit of Mobile IPv6 is that even though the mobile node changes locations and addresses, the existing connections through which the mobile node is communicating are maintained. To accomplish this, connections to mobile nodes are made with a specific address that is always assigned to the mobile node, and through which the mobile node is always reachable. Mobile IPv6 provides Transport layer connection survivability when a node moves from one link to another by performing address maintenance for mobile nodes at the Internet layer.

Mobile IPv6 Components

The figure shows the components of Mobile IPv6. The components of Mobile IPv6 are the following:

**Home Link**

This is the link that is assigned the home subnet prefix, from which the mobile node obtains its home address. The home agent resides on the home link.
**Home Address**

An address assigned to the mobile node when it is attached to the home link and through which the mobile node is always reachable, regardless of its location on an IPv6 network. If the mobile node is attached to the home link, Mobile IPv6 processes are not used and communication occurs normally. If the mobile node is away from home (not attached to the home link), packets addressed to the mobile node's home address are intercepted by the home agent and tunneled to the mobile node's current location on an IPv6 network. Because the mobile node is always assigned the home address, it is always logically connected to the home link.

**Home Agent**

A router on the home link that maintains registrations of mobile nodes that are away from home and the different addresses that they are currently using. If the mobile node is away from home, it registers its current address with the home agent, which tunnels data sent to the mobile node's home address to the mobile node's current address on an IPv6 network and forwards tunneled data sent by the mobile node.
Although the figures in this report show the home agent as the router connecting the home link to an IPv6 network, the home agent does not have to serve this function. The home agent can also be a node on the home link that does not perform any forwarding when the mobile node is at home.

**Mobile Node**

It is an IPv6 node that can change links, and therefore addresses, and maintain reachability using its home address. A mobile node has awareness of its home address and the global address for the link to which it is attached (known as the care-of address), and indicates its home address/care-of address mapping to the home agent and Mobile IPv6-capable nodes with which it is communicating.

**Foreign Link**

This is a link that is not the mobile node's home link.

**Care-of address**

An address used by a mobile node while it is attached to a foreign link. For stateless address configuration, the care-of address is a combination of the foreign subnet prefix and an interface ID determined by the mobile node. A mobile node can be assigned multiple care-of addresses; however, only one care-of address is registered as the primary care-of address with the mobile node's home agent. The association of a home address with a care-of address for a mobile node is known as a binding. Correspondent nodes and home agents keep information on bindings in a binding cache.

**Correspondent Node**

This is an IPv6 node that communicates with a mobile node. A correspondent node does not have to be Mobile IPv6-capable. If the correspondent node is Mobile IPv6-capable, it can also be a mobile node that is away from home.
Mobile IPv6 Transport Layer Transparency

To achieve Transport layer transparency for the home address while the mobile node is assigned a care-of address, Mobile IPv6-capable nodes use the following:

- When a mobile node that is away from home sends data to a correspondent node, it sends the packets from its care-of address and includes the mobile node's home address in a Home Address option in a Destination Options extension header.

- When the correspondent node receives the packet, it logically replaces the source address of the packet (the care-of address) with the home address stored in the Home Address option.

- When a Mobile IPv6-capable correspondent node sends data to a mobile node that is away from home, it sends the packets to the care-of address and includes a Type 2 Routing extension header containing a single address, the mobile node's home address. When the mobile node receives the packet, it processes the Type 2 Routing header and logically replaces the destination address of the packet (the care-of address) with the home address from the Type 2 Routing header.

If a correspondent node is not Mobile IPv6-capable, then packets sent between the correspondent node and the mobile node that is away from home are exchanged via the home agent. The correspondent node sends packets to the mobile node's home address. These packets are intercepted by the home agent and tunneled to the mobile node's care-of address. The mobile node tunnels packets destined for the correspondent node to the home agent, which forwards them to the correspondent node. This indirect method of delivery, known as bidirectional tunneling, although inefficient, allows communication between mobile nodes that are away from home and correspondent nodes that are not Mobile IPv6-capable.

The following two Figures show the exchange of packets directly from a mobile node on a foreign network to a correspondent node; the feature that gives MIPv6 an advantage over tunneling in MIPv4.
IPv6 Header
- Source Address is CoA
- Destination Address is CNA
- Destination Options Header
- Home Address Option with HoA
- Upper Layer PDU

IPv6 Header
- Source Address is CNA
- Destination Address is CoA
- Type 2 Routing Header
- HoA
- Upper Layer PDU
HARDWARE IMPLEMENTATION

TENTATIVE NETWORK ARCHITECTURE

In order to implement the architecture proposed in the previously discussed software simulations, a simple instance of the proposed architecture, shown in the figure (proposed network architecture realization), was intended to be realized in hardware. At least one of the routers shown should be enabled with IPv6, mobility support and home agent capabilities. As IPv6 is not yet deployed on a large scale at the time of writing this paper, these routers are not available in the local market and rarely found in the international market. The following are some of the very few available routers.

- Cisco 1800 series routers (two routers would be needed, each at $1000 or only one router needed but additional network cards will have to be installed and VLANs will have to be created) excluding taxes and shipping.

- Cisco 2800 series routers (two routers would be needed, each at $6000 or only one router needed but additional network cards will have to be installed and VLANs will have to be created) excluding taxes and shipping. As the hardware implementation in this project is intended to be a low cost solution, and due to a limited budget and to the unavailability and high cost of the routers in the local and international markets, finding an alternate low cost solution was a necessity.
LOW COST ALTERNATIVE

Through an extensive research on the internet, it was found that PCs with modest features could be turned into routers with all their functionalities included. Furthermore, these PCs could be enabled to operate under MIPv6. This allows the hardware realization of the network to be at a very low cost, which is a practical solution for the purpose of a prototype demonstration. In addition, it could be used by a variety of small businesses wishing to incorporate the benefits of WiFi mobility into their infrastructure. One way to turn PCs into routers with Mobile IPv6 Home Agent capabilities is through first providing the PCs with routing tables and then installing the application Mobile IP for Linux (MIPL) and a router advertisement daemon (RADVD). The MN and CN should be also MIPv6 enabled so MIPL should be also installed on both. See Appendices A and B for an explanation of operating systems, Linux and MIPL.

TEST BED AND SYSTEM REQUIREMENTS

The test bed network is a fully functional MIPv6 network and consists of a PC-based Correspondent Node (CN), two PC-based routers: the Home Agent (HA) and the Foreign Router (FR), and a PC-based Mobile Node. The test bed was used to perform MIPv6 handoff and measure handoff delay. The following subsections will describe the nodes, connections and Access Points used in the network. After that the modified network architecture is presented followed by a description of the address space used, configurations and startup procedures of the components of the network.
TEST BED AND SYSTEM REQUIREMENTS

In this section, the test bed that was used to verify the software results will be presented. The test bed network is a fully functional MIPv6 network and consists of a PC-based Correspondent Node (CN), two PC-based routers: the Home Agent (HA) and the Foreign Router (FR), and a PC-based Mobile Node. The test bed was used to perform MIPv6 handoff and measure handoff delay. The following subsections will describe the nodes, connections and Access Points used in the network. After that the modified network architecture is presented followed by a description of the address space used, configurations and startup procedures of the components of the network.

NODES

Correspondent Node (CN)

- One Dell Desktop computer.
- Intel Pentium 4 Processor.
- 512 MB RAM.
- One 10/100 Ethernet card (built in on the motherboard or any Ethernet card having a Realtek 8139 chip will be recognized automatically by Fedora Core 5 & 6).
- Fedora Core 5 distribution of the Linux Operating System.
- Linux kernel version 2.6.16. See Mobile IPv6 HOWTO for installation instructions
- MIPL 2.0.2 as the MIPv6 implementation software. See Mobile IPv6 HOWTO for installation instructions
- VLC media player as the server-client application. See HOWTO VIDEO LAN CLIENT for installation instructions

Home Agent (HA)

- One Dell Desktop computer.
- Intel Pentium 4 processor.
- 512 MB RAM.
- Two 10/100 Ethernet cards (built in on the motherboard or any Ethernet card having a Realtek 8139 chip will be recognized automatically by Fedora Core 5 & 6)
- One network card connects the HA to the AP, the other connects the HA to the CN.
- Fedora Core 5 distribution of the Linux Operating System
- Linux kernel version 2.6.16.
- MIPL 2.0.2 as the MIPv6 software implementation.
- Radvd router advertisement daemon.
Foreign Router (FR)

- One Dell Desktop computer.
- Intel Pentium 4.
- 512 MB RAM.
- Two 10/100 Ethernet cards (built in on the motherboard or any Ethernet card having a Realtek 8139 chip will be recognized automatically by Fedora Core 5 & 6)
- One network card connects the FR to the AP; the other connects the FR to the CN.
- Fedora Core 5 distribution of Linux Operating System.
- Linux kernel version 2.6.16.
- Radvd router advertisement daemon.

Mobile Node (MN)

- One Dell Desktop computer.
- Intel Pentium 4 processor.
- 512 MB RAM.
- One wireless 802.11g USB NETGEAR network card.
- Fedora Core 5 distribution of the Linux Operating System.
- Linux kernel version 2.6.16.
- MIPL 2.0.2 as the MIPv6 implementation software.
- VLC media player as the client-server application.

ACCESS POINTS (APS)

- Two Linksys WAP54G Access Points.

The APs use 802.11g and give data rates up to 54Mbps; they are also backward compatible with 802.11b. The transmit power of these APs is preset to 30mW (15 dBm).

CONNECTIONS

- The CN, HA and FR are connected to the switch by CAT 5 Ethernet cables, as shown in the following figure.

- The two APs are connected to the Ethernet cards on the HA and FR by cross cables.
**MODIFIED NETWORK ARCHITECTURE**

**ADDRESS SPACE**

The network is connected as shown in the figure.

- **CN**: It is connected to the network through 1 Ethernet card with the address 2001:660:5503:276a::3 (Global address). It is connected to both the HA and the FR through a switch.

- **HA**: It has 2 Ethernet cards. One is connected to the CN via the switch and has the address 2001:660:5503:276a::2. The other is connected to the MN through the Access Point and has the address 2001:660:5503:276b::1

- **FR**: It has 2 Ethernet cards. One is connected to the CN via the switch and has the address 2001:660:5503:276a::1. The other is connected to the Access Point and has the address 2001:660:5503:276c::1

- **MN**: It has one wireless interface with home address 2001:660:5503:276b::4. The MN after traveling to the foreign network acquires the following care of address 2001:660:5503:276c:20f:b5ff:fed0:9f26
**CONFIGURATION**

After installing the required software on the network nodes and connecting them properly as explained in the previous sections, each node should be configured as follows.

**Correspondent Node (CN)**

The following commands should be typed in a terminal window on the CN.

```plaintext
# ifconfig eth0 inet6 add 2001:660:5503:276a::3/64
```

The first command statically assigns an IPv6 address for the interface specified (eth0 in this example). The second and third commands specify the static routes for data going through the CN. For example, the second command means: “any packet having the destination address prefix 2001:660:5503:276b::/0 should be routed to interface 2001:660:5503:276a::2.

A configuration file for MIPL should be created and saved as mip6d.conf in the directory `/usr/local/etc`. The contents of this file are as follows:

```plaintext
# This is the mip6d Correspondent Node configuration file
NodeConfig CN;
DebugLevel 10;
## Support route optimization with MNs
DoRouteOptimizationCN enabled;
```

**Home Agent (HA)**

The following commands should be typed in a terminal window on the HA.

```plaintext
# ifconfig eth0 inet6 add 2001:660:5503:276b::1/64
# ifconfig eth1 inet6 add 2001:660:5503:276a::2/64
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
# echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

Here, eth0 is the interface (network card) connected to the AP, and eth1 is the interface connected to the switch. The "echo" commands (third to sixth command) are used to set network flags. A configuration file for MIPL should be created and saved as mip6d.conf in the directory `/usr/local/etc`. The contents of this file are as follows:
# Mobile IPv6 configuration file: Home Agent
# filename: /usr/local/etc/mip6d.conf
NodeConfig HA;
DebugLevel 10;
Interface "eth0";
## IPsec configuration
UseMnHalIPsec enabled;
IPsecPolicySet {
HomeAgentAddress 2001:660:5503:276b::1;
HomeAddress 2001:660:5503:276b::4/64;
IPsecPolicy HomeRegBinding UseESP;
IPsecPolicy MobPfxDisc UseESP;
IPsecPolicy TunnelMh UseESP;
}

Here, eth0 is the interface where the PC acts as a HA, which is the interface connected to the AP. If IPsec is not used, then the option UseMnHalPsec should be disabled.

The command: UseMnHalIPsec enabled;
Should be replaced by: UseMnHalIPsec disabled;

If IPsec is used, then a security association file that sets the keys and encryption between the HA and the MN should be created and saved as sa.conf in the directory /usr/local/etc. The contents of this file are as follows:

##----------------------------------------------------------------------------------------
## IPsec MN -> HA, CN (BU) and HA, CN -> MN (BA) Transport Mode
##----------------------------------------------------------------------------------------
add 2001:660:5503:276b::4
    2001:660:5503:276b::1
    esp 0001
    -m transport
    -E null
    -A null;
add 2001:660:5503:276b::1
    2001:660:5503:276b::4
    esp 0002
    -m transport
    -E null
    -A null;
Add 2001:660:5503:276b::4
  2001:660:5503:276b::1
  esp 0016
  -m tunnel
  -E null
  -A null;
add 2001:660:5503:276b::1
  2001:660:5503:276b::4
  esp 0017
  -m tunnel
  -E null
  -A null

A configuration file for the router advertisement daemon should be created and saved as `radvd.conf` in the directory `/etc`. The contents of this file are as follows:

```plaintext
interface eth0
  { AdvSendAdvert on;
    MinRtrAdvInterval 0.03;
    MaxRtrAdvInterval 0.07;
    AdvIntervalOpt off;
    AdvHomeAgentFlag on;
    HomeAgentLifetime 10000;
    HomeAgentPreference 20;
    AdvHomeAgentInfo on;
    prefix 2001:660:5503:276b::1/64
      { AdvRouterAddr on;
        AdvOnLink on;
        AdvAutonomous on;
        AdvPreferredLifetime 10000;
        AdvValidLifetime 12000;
      };
  };
```
Here also, eth0 is the interface connected to the AP.

**Foreign Router (FR)**

The following commands should be typed in a terminal window of the FR.

```
# ifconfig eth0 inet6 add 2001:660:5503:276c::1/64
# ifconfig eth1 inet6 add 2001:660:5503:276a::1/64
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
# echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

Here, eth0 is the interface (network card) connected to the AP, and eth1 is the interface connected to the switch. A configuration file for the router advertisement daemon should be created and saved as `radvd.conf` in the directory `/etc`. The contents of this file are as follows:

```ini
interface eth0
{
    AdvSendAdvert on;
    AdvIntervalOpt off;
    MinRtrAdvInterval 0.03;
    MaxRtrAdvInterval 0.07;
    AdvHomeAgentFlag off;
    prefix 2001:660:5503:276c::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    }
}
```

**Mobile Node (MN)**

The following commands should be typed in a terminal window on the MN.

```
# depmod -a
# modprobe ndiswrapper
# iwlist wlan0 scan
# ifconfig wlan0 up
# iwconfig wlan0 mode Managed essid "WiFox1"
# ifconfig wlan0 inet6 add 2001:660:5503:276b::4/64
# echo 0 > /proc/sys/net/ipv6/conf/wlan0/forwarding
# echo 1 > /proc/sys/net/ipv6/conf/wlan0/autoconf
# echo 1 > /proc/sys/net/ipv6/conf/wlan0/accept_ra
# echo 1 > /proc/sys/net/ipv6/conf/wlan0/accept_redirects
```
The first two commands initialize ndiswrapper and the wireless network card driver (see Mobile IPv6 HOWTO for more details). The third command scans the available APs and other wireless devices and is used to make sure that the wireless network card operates normally. Here, wlan0 is the wireless network card. The fourth and fifth commands associate the wireless network card with the AP connected to the HA, here the AP has the essid “WiFox1”.

A configuration file for MIPL should be created and saved as mip6d.conf in the directory /usr/local/etc. The contents of this file are as follows:

```
# Mobile IPv6 configuration file: Mobile Node
#
# filename: /etc/mip6d.conf
NodeConfig MN;
DebugLevel 10;
MnDiscardHaParamProb enabled;
Interface "wlan0";
MnHomeLink "wlan0" {
    HomeAgentAddress 2001:660:5503:276b::1;
    HomeAddress 2001:660:5503:276b::4/64;
}

##
## IPsec configuration
##
UseMnHalIPsec enabled;
IPsecPolicySet {
    HomeAgentAddress 2001:660:5503:276b::1;
    HomeAddress 2001:660:5503:276b::4/64;
    IPsecPolicy HomeRegBinding UseESP;
    IPsecPolicy MobPfxDisc UseESP;
    IPsecPolicy TunnelMh UseESP;
}
```

If IPsec is not used, then the option UseMnHalIPsec should be disabled.

The command: UseMnHalIPsec enabled;
Should be replaced by: UseMnHalIPsec disabled;

If IPsec is used, then a security association file that sets the keys and encryption between the HA and the MN should be created and saved as sa.conf in the directory /usr/local/etc. The contents of this file are as follows:
## IPsec MN -> HA, CN (BU) and HA, CN -> MN (BA) Transport Mode

```
add  2001:660:5503:276b::4
     2001:660:5503:276b::1
     esp 0001
     -m transport
     -E null
     -A null;
add  2001:660:5503:276b::1
     2001:660:5503:276b::4
     esp 0002
     -m transport
     -E null
     -A null;
```

## IPsec MN -> HA (HoTI) and HA -> MN (HoT) Tunnel Mode

```
add  2001:660:5503:276b::4
     2001:660:5503:276b::1
     esp 0016
     -m tunnel
     -E null
     -A null;
add  2001:660:5503:276b::1
     2001:660:5503:276b::4
     esp 0017
     -m tunnel
     -E null
     -A null;
```

**STARTING THE NETWORK**

After setting the network, connecting it and configuring it as explained in the previous sections, the network should be a fully functioning IPv6 network. All nodes should be able to ping all other nodes on the network successfully. The ping command has the following syntax:

```
ping6 2001:660:5503:276b::4
```

where 2001:660:5503:276b::4 is the address that needs to be pinged. The ping command sends and receives small ICMP packets to and from the specified address to test the connectivity of this address to the pinging node.
Starting IPsec, RADVD and MIPv6

If IPsec is used, it needs to be started. The following commands need to be written in a terminal window on both HA and MN:

```
# setkey -FP
# setkey -F
# setkey -f /usr/local/etc/sa.conf
```

Then, radvd router advertisement daemon needs to be started. The following command needs to be typed on both HA and FR in a terminal window:

```
# radvd
```

Finally, MIPL has to be started. The following command needs to be typed in a terminal window on CN, HA then MN, respectively:

```
# mip6d
```

The network is then MIPv6 enabled. Mobility can be tested by pinging the mobile node on its home address while it is moving from the home network to the foreign network (moving from the range of one AP to the other). The ping will temporarily stop during handoff and then resume when the MN is in the foreign network. Mobility can also be tested using an application such as VLC media player installed on both MN and CN. The CN can send streaming video to the MN, and the video application on the MN will temporarily halt during handoff then resume normally. Binding Updates sent to the CN will indicate that Route Optimization is being used to reduce unnecessary tunnels. This means that data sent from CN goes to MN directly through the FR without first going to HA then FR.

MOBILE IP FOR LINUX INSTALLATION

Following are the steps needed to obtain a fully working network according to the implemented test bed in our thesis:

Placing and Un-taring the Linux Kernel
Place the kernel linux-2.6.16.tar.bz2 into /usr/src folder.
Un-tar (Unzip) the Linux kernel 2.6.16
```
# cd /usr/src
# tar jxvf linux-2.6.16.tar.bz2
```

Make a link to the folder Linux
```
# ln -s linux-2.6.16 Linux
# cd Linux
```

Placing and Un-taring the MIPL source
Place the mobile IPv6 application mipv6-2.0.2.tar.gz into /usr/local/src.
Un-tar (Unzip) the Mobile IPv6 2.0.2 application
```
# cd /usr/local/src
# tar zxfv mipv6-2.0.2.tar.gz
```
Applying the MIPv6 patch
# zcat /usr/local/src/mipv6-2.0.2-linux-2.6.16.patch.gz | patch –p1 --dry-run
(The –dry-run command is used to check that the patch will apply correctly)
# zcat /usr/local/src/mipv6-2.0.2-linux-2.6.16.patch.gz | patch -p1

Compiling and Installing the new kernel version
# make clean

Create the file .config by typing the command
# make menuconfig
In the blue screen that will appear click exit and then save to save the kernel configuration
In the terminal command type the following to edit the file .config:
# gedit .config
Make sure these options are present to support MIPv6:
CONFIG_EXPERIMENTAL=y
CONFIG_SYSVIPC=y
CONFIG_PROC_FS=y
CONFIG_NET=y
CONFIG_INET=y
CONFIG_IPV6=y (most probably not set)
CONFIG_IPV6_MIP6=y (most probably not set)
CONFIG_XFRM=y
CONFIG_XFRM_USER=y
CONFIG_XFRM_ENHANCEMENT=y (most probably not set)
CONFIG_IPV6_TUNNEL=y (most probably not set)
CONFIG_IPV6_ADVANCED_ROUTER=y (most probably not set)
CONFIG_IPV6_MULTIPLE_TABLES=y (most probably not set)
CONFIG_IPV6_SUBTREES=y (most probably not set)
CONFIG_ARPD=y (most probably not set)
To enable IPSec (IP Security), make sure to include these options:
CONFIG_INET6_ESP=y
CONFIG_NET_KEY=y
CONFIG_NET_KEY_MIGRATE=y

To be sure that all the correct options are set and that the kernel was correctly configured to support MIPv6: run chkconf_kernel.sh from the Mobile IPv6 folder (/usr/local/src/mipv6-2.0.2). This is a small shell script included in the MIPL tarball. It checks that all the options needed by MIPL are included. If some options are missing, the function will report the missing options. If the script reports any missing options, fix it before you proceed.

In the terminal command type the following:
# cd /usr/local/src/mipv6-2.0.2
# ./chkconf_kernel.sh /usr/src/Linux
Installing the Linux kernel

As Linux got matured, the size of the kernels generated by the users exceeded the limits imposed by some computer architectures. In order to overcome such a problem, the kernel was split over discontiguous memory regions. This is operation is done through a file called zImage or bzImage (big zImage) which is a compressed kernel image that is in arch/i386/boot. In order to install the new kernel into the system, a bzImage file must be created first through the following steps:

# cd /usr/src/Linux
# make bzImage

Next, are the steps needed to create the loadable modules:

# make modules
# make modules_install

mkinitrd is an initial image used by the kernel in order to preload the block device modules which are needed to access the root file system. This procedure makes it simple to build and use modular device drivers. The following commands are to be followed in order to make the mkinitrd file.

# mkinitrd /boot/initrd-2.6.16.img 2.6.16
# cp arch/i386/boot/bzImage /boot/bzImage-2.6.16
# cp System.map /boot/System.map-2.6.16

Creating a booting entry for the new kernel in the boot file:

# ln -s /boot/System.map-2.6.16 /boot/System.map
In /boot/grub/ edit the menu.lst file and add the following lines:
title <ANYNAME> (2.6.16)
root (hd0,0)
kernell /boot/bzImage-2.6.16 ro root=LABEL=/ rhgb quiet
initrd /boot/initrd-2.6.16.img

Change the default booting of the system to the one just created by changing the default number to the number of the new booting entry taking in consideration that the first booting entry is numbered 0.

Save the file
Reboot

Installing Mobile IPv6

Configure, compile and install the source code. The --enable-vt option is included to configure. This option will enable a virtual terminal listening on localhost port 7777.

# cd usr/local/src/mipv6-2.0.2
# CPPFLAGS=-I/usr/src/Linux/include ./configure --enable-vt
# make
# make install

Installing Radvd

Download radvd-1.0-1.rhl73.i386.rpm and then double click on it.