

MIPL HOWTO (pt)

**Manual de instalação e configuração da aplicação de
Mobilidade IPv6 para Linux - versão Portuguesa**

Versão: mipv6-2.0-rc3_v1

22 de Janeiro de 2006

1. Introdução

Este documento descreve o software e os procedimentos para configurar e usar mobilidade IPv6 em linux (MIPL).

A [RFC 3775, “Mobility support in IPv6”](#), definida pelo IETF, esclarece o que é, e o porquê da mobilidade IPv6.

O site www.mobile-ipv6.org, é o site oficial da mobilidade IPv6 para Linux (MIPL), gerido e mantido por um grupo de trabalho da Universidade de Helsínquia na Finlândia, responsáveis pela implementação prática do draft “Mobility Support in IPv6” n°24, levando à sua evolução para a RFC 3775. Neste site encontra-se disponível o software MIPL, juntamente com alguma informação, documentação e links úteis. Encontra-se também um bom [MIPL HOWTO \(Mobile IPv6 for linux \(en\) by Lars Strand\)](#), embora que já desactualizado e algo incompleto, no qual se baseia este documento.

1.1. O que é a Mobilidade IPv6?

“(...)Cada nó móvel é sempre identificado pelo seu home address (HoA), independentemente do seu ponto de ligação à Internet. Enquanto situado longe da sua rede original (home network), o nó móvel tem também atribuído um care-of-address (CoA), que fornece informação da sua localização. Os pacotes IPv6 enviados para o HoA do nó, são reenviados de forma transparente, pelo Home Agent (HA), para o CoA. O protocolo permite que os nós IPv6 armazenem o registo da ligação entre um HoA e um CoA de um nó móvel, podendo assim enviar pacotes destinados a esse nó móvel directamente para o seu CoA.(...)”

[RFC 3775 “Mobility Support in IPv6”, Abstract, pag. 0.](#)

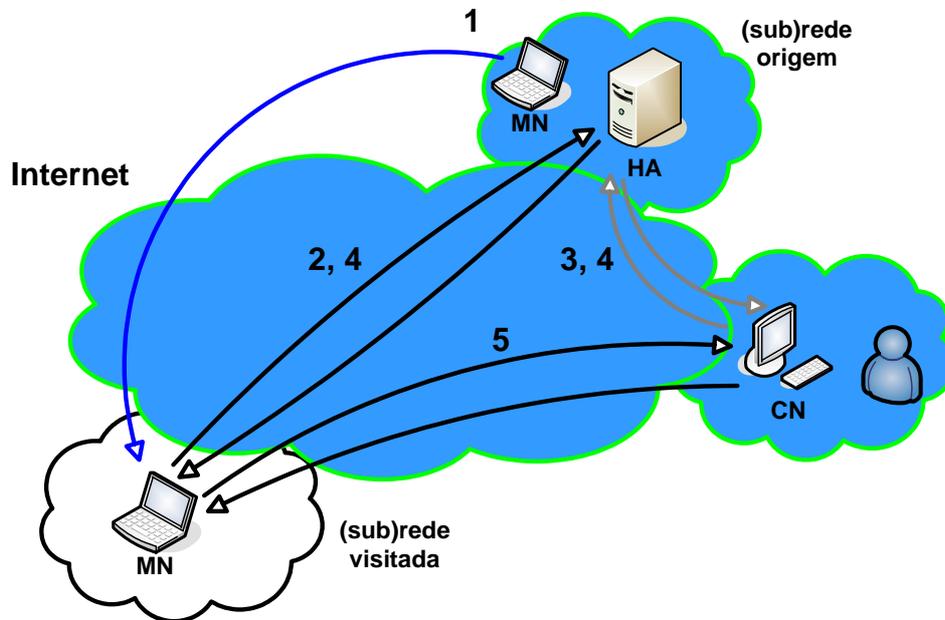
1.2. Porquê Mobilidade IPv6?

“(...)Sem suporte específico para mobilidade IPv6 (MIPv6), os pacotes destinados para um nó móvel (PC ou Router), não estarão aptos a alcançar esse nó enquanto este se encontrar fora do seu troço de ligação original (o home link original do nó móvel e cujo prefixo de subrede é igual ao do seu HoA), uma vez que o encaminhamento é baseado no prefixo de subrede existente no campo do endereço IP de destino do pacote. De modo a manter a comunicação, mesmo que ocorra movimento, o nó móvel poderá mudar o seu IP cada vez que se move para um novo link, mas, neste caso, o nó móvel não poderá manter o transporte e as ligações das camadas superiores quando muda de localização. O suporte de mobilidade em IPv6 é particularmente importante, uma vez que os computadores móveis irão constituir uma maioria, ou pelo menos uma fracção significativa da população da Internet durante o tempo de vida do IPv6.(...)”

[RFC 3775 “Mobility Support in IPv6”, Introduction, pag. 5 e 6.](#)

1.3. Como funciona?

Neste ponto estão explicados os passos básicos do funcionamento de MIPv6.



Mobilidade IPv6

1. O nó móvel (MN) desloca-se para uma rede diferente, e recebe um novo care-of-address (CoA).
2. O MN regista o seu movimento (binding update (BU)) no Home Agent (HA)(o novo CoA é associado ao Home Address(HoA) no HA). O HA envia um binding acknowledgement(BA) para o MN.
3. Um nó, designado de Correspondent Node (CN), quer comunicar com o MN, usando o seu HoA. O HA intercepta os pacotes destinados ao MN.
4. O HA envia por túnel todos os pacotes destinados ao MN usando o seu CoA.
5. O MN responde para o CN usando o seu CoA (fazendo um registo no CN caso este suporte MIPv6), comunicando directamente com este (otimização de Rota), ou pode enviar por túnel todos os pacotes para o HA, sendo este responsável por os enviar para o CN.

Ver figura “Mobilidade IPv6” no topo.

Para mais detalhes, ler a [RFC 3775, "Mobility Support in IPv6"](#).

2. IPv6

O IP versão 6 (IPv6) [\[RFC - 2460\]](#) é a nova versão do protocolo da Internet (IP), designado como o sucessor do IP versão 4 (IPv4) [\[RFC 0791\]](#). As mudanças mais significativas do IPv4 para o IPv6 são as enumeradas de seguida:

- Expansão das capacidades de endereçamento
- Simplificação do formato do cabeçalho
- Melhoramento do suporte de extensões e opções
- Capacidade de fluxo por labels
- Capacidades de autenticação e privacidade

Deverá haver conhecimento básico do processo de auto-configuração IPv6 stateless, para compreender como funciona a mobilidade IPv6 (MIPv6), especificado em “IPv6 Stateless Address Autoconfiguration” no [\[RFC2462\]](#).

Para mais informações genéricas do IPv6, poderá ser consultada a página do [IETF's IPv6 Working Group](#).

3. Mobilidade IPv6 para Linux

Existem duas implementações de Mobilidade IPv6 para Linux disponíveis.

A Universidade de Lancaster no Reino Unido tem a mais antiga (<http://www.cs-ipv6.lancs.ac.uk/MobileIP/>). A última implementação foi desenvolvida para o Kernel 2.1.90, de acordo com o especificado no draft 5 do IETF de Mobilidade em IPv6 (o actual RFC 3775). No entanto o código e o sítio web não são actualizados desde 1998, por isso esta solução é considerada obsoleta.

A outra implementação, é a “Helsinki University of Technology's MIPL Project”, que é actualizada periodicamente. O último kernel suportado é o 2.6.11 (ultima actualização do software em 31-10-2005). No site <http://www.mobile-ipv6.org/>, encontra-se disponível o [patch e as user space tools](#), bem como [documentação](#), [links](#), e uma [Mailing List](#) bastante solicitada, com o respectivo arquivo.

Como é óbvio, a implementação escolhida para a Test bed, foi a disponibilizada pela universidade de Helsínquia.

3.1. Aplicar o patch ao kernel

Enquanto se aguarda pela inclusão da Mobilidade IPv6 no kernel do Linux, tal como já sucede com o IPv6, há que recorrer à aplicação de um Patch ao Kernel. A implementação modifica a pilha IPv6 do Kernel, por isso é necessária a sua recompilação. De seguida é apresentado este processo detalhado.

Nota! Este processo permite instalar o suporte para as funcionalidades de HA, MN e CN. Apesar de não ser possível manter mais do que uma funcionalidade em simultâneo, o modo escolhido irá depender das configurações posteriores.

Notas e considerações antes de começar a implementação:

A distribuição usada para implementar a Test Bem foi a Fedora Core 3 (FC3).

A distribuição Fedora mais recente é o Fedora Core 4 (FC4), e inicialmente foi esta distribuição que começou por ser usada, no entanto alguns problemas levaram a abdicar desta em função da FC3. Os motivos desta opção em detrimento da primeira são enunciados de seguida.

No FC4, o kernel usado é o kernel-2.6.11-1.1369_FC4 (na linha de comando: `rpm -q kernel`). Este deriva do 2.6.11 ao qual foram aplicadas algumas modificações (patches) exclusivamente para a sua inclusão na distribuição, no entanto a sua versão não deixa de ser a 2.6.11. Por isso houve uma tentativa de aplicar o patch neste kernel, que se revelou infrutífera. Ocorreram falhas (failed hunks) que permitiram concluir que teria de ser usada a versão 2.6.11 “pura”.

Após o download desta, aplicar o patch foi fácil e não ocorreram problemas, no entanto o processo de recompilação revelou-se uma tarefa complicada, nomeadamente a recompilar os módulos (Make Modules), cujo processo não foi possível completar devido a uns erros ocorridos.

Após várias tentativas falhadas, pensou-se que o problema seria devido ao facto de o kernel ser igual ao usado na distribuição, e de alguma forma estaria a ocorrer um conflito no processo recompilação.

Optou-se então por usar a distribuição do FC3, que traz de origem o kernel-2.6.9-1.667. Nesta não existiu nenhum problemas nem a aplicar o Patch ao kernel 2.6.11, nem no processo de recompilação deste. Como a máquina usada era bastante rápida o processo completo de recompilação também o foi.

De seguida são apresentados os passos necessários para implementar suporte de MIPv6 numa máquina Linux.

1. Fazer download da ultima versão do código fonte do MIPv6 para Linux e respectivo Patch no sítio <http://www.mobile-ipv6.org/>. A ultima versão disponível à data deste documento (sítio actualizado em 31-10-2005) é a *mip6-2.0-rc3*, sendo o respectivo patch o *mip6-2.0-rc3-linux-2.6.11.patch*, tal como indicado no nome, para o kernel 2.6.11.

```
# cd /usr/local/src
# wget http://www.mobile-ipv6.org/software/download/mip6-2.0-rc3.tar.gz
# wget http://www.mobile-ipv6.org/software/download/mip6-2.0-rc3-linux-2.6.11.patch.gz
# tar zxfv mip6-2.0-rc3.tar.gz
# tar zxfv mip6-2.0-rc3-linux-2.6.11.patch.gz
```

2. Fazer download e descompactar a versão do kernel correspondente no sítio oficial, <ftp.kernel.org>, ou no mirror mais próximo.

```
# cd /usr/src
# wget ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-2.6.11.tar.bz2
# tar jxvf linux-2.6.11.tar.bz2
# cd linux-2.6.11
```

3. Antes de aplicar o patch, podemos ver a configuração existente através de um qualquer make *config, e.g., make menuconfig.

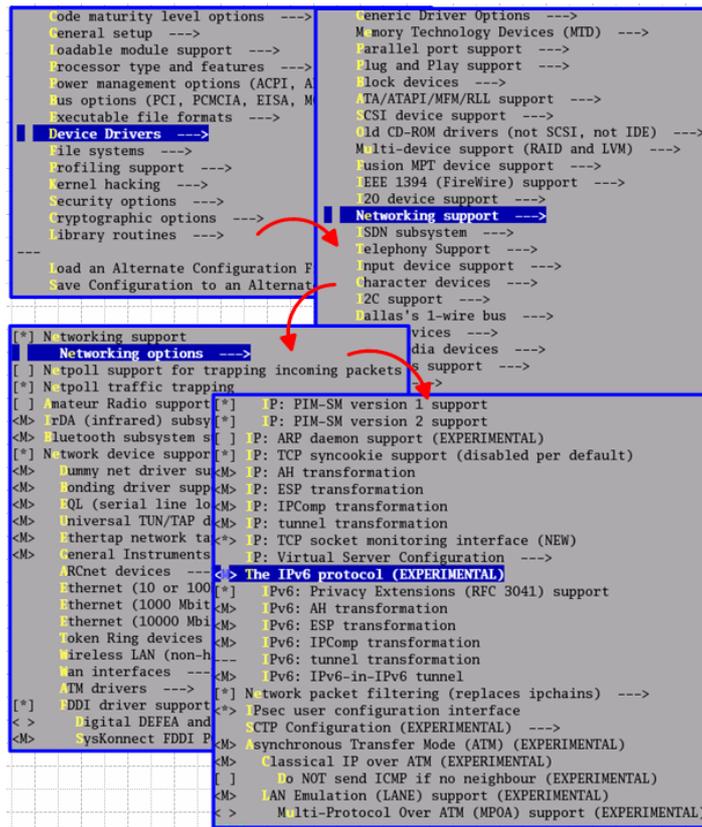
```
# make menuconfig
```

No menu de configuração:

```
Device Drivers --->
  Networking support --->
    Networking options --->
```

Neste menu poderá ser visto as opções de suporte IPv6, e a inexistência de opções de MIPv6. Antes de avançar deverá sair do menu de configuração e fazer o make clean.

```
# make clean
```



Configurações disponíveis antes de aplicar o patch.

4. Verificar se o Patch pode ser aplicado correctamente:

```
# patch -p1 --dry-run < /usr/local/src/mipv6-2.0-rc3-linux-2.6.11.patch
```

5. A opção --dry-run, verifica se o Patch será aplicado correctamente. Se ocorrerem erros (failed hunks) algo está incorrecto e o processo não poderá continuar, devendo ser revistos os passos anteriores. Se tudo correr bem então há que aplicar efectivamente o Patch:

```
# patch -p1 < /usr/local/src/mipv6-2.0-rc3-linux-2.6.11.patch
```

6. Agora a árvore do kernel está pronta para a configuração. Como referido anteriormente, as opções de MIPv6 estão em “Device Drivers > Networking support > Networking options”. É possível verificar que novas opções apareceram, nomeadamente opções de MIPv6. A figura seguinte mostra essas mesmas opções.

```

[*] IP: PIM-SM version 2 support
[ ] IP: ARP daemon support (EXPERIMENTAL)
[*] IP: TCP syncookie support (disabled per default)
<M> IP: AH transformation
<M> IP: ESP transformation
<M> IP: IPComp transformation
<M> IP: tunnel transformation
<*> IP: TCP socket monitoring interface
IP: Virtual Server Configuration --->
<M> The IPv6 protocol (EXPERIMENTAL)
[*] IPv6: Privacy Extensions (RFC 3041) support
<M> IPv6: AH transformation
<M> IPv6: ESP transformation
<M> IPv6: IPComp transformation
--- IPv6: tunnel transformation
<M> IPv6: IPv6-in-IPv6 tunnel
[ ] IPv6: advanced router (EXPERIMENTAL) (NEW)
[ ] IPv6: Mobility (EXPERIMENTAL) (NEW)
[ ] Network packet filtering (replaces ipchains) --->
<*> Transformation user configuration interface
[ ] Transformation migrate database (EXPERIMENTAL) (NEW)
[ ] Transformation Debug Message (NEW)
CTP Configuration (EXPERIMENTAL) --->
<M> Asynchronous Transfer Mode (ATM) (EXPERIMENTAL)
<M> Classical IP over ATM (EXPERIMENTAL)

[*] IP: PIM-SM version 2 support
[ ] IP: ARP daemon support (EXPERIMENTAL)
[*] IP: TCP syncookie support (disabled per default)
<M> IP: AH transformation
<M> IP: ESP transformation
<M> IP: IPComp transformation
<M> IP: tunnel transformation
<*> IP: TCP socket monitoring interface
IP: Virtual Server Configuration --->
<M> The IPv6 protocol (EXPERIMENTAL)
[*] IPv6: Privacy Extensions (RFC 3041) support
<M> IPv6: AH transformation
<M> IPv6: ESP transformation
<M> IPv6: IPComp transformation
--- IPv6: tunnel transformation
<M> IPv6: IPv6-in-IPv6 tunnel
[ ] IPv6: advanced router (EXPERIMENTAL)
[*] IPv6: policy routing
[*] IPv6: source address based routing
[*] IPv6: Mobility (EXPERIMENTAL)
[*] IPv6: Mobility Debug Message
--- Network packet filtering (replaces ipchains) --->
<*> Transformation user configuration interface
[ ] Transformation migrate database (EXPERIMENTAL) (NEW)
[ ] Transformation Debug Message (NEW)

```

Configurações após aplicar o patch.

O “M” significa Módulo, e o “*” é embutido no kernel.

- Para ter a certeza de que todas as opções estão correctamente seleccionadas, poderá ser usado o script `chkconf_kernel.sh` (indicando o kernel a verificar), que é um pequeno script shell que vem junto com as userspace tools `mipv6`.

```

# cd /usr/local/src/mipv6-2.0-rc3
# ./chkconf_kernel.sh /usr/src/linux-2.6.11

```

O output do script deverá ser semelhante ao apresentado de seguida:

```

Checking kernel configuration...
Using /mipv6/linux-2.6.11/.config
Warning: CONFIG_IPV6 should be set to y (m)
Warning: CONFIG_IPV6_TUNNEL should be set to y (m)
Warning: CONFIG_INET6_ESP should be set to y (m)
Warning: CONFIG_NET_KEY should be set to y (m)
Warning: CONFIG_NET_KEY_MIGRATE should be set to y (not supported)

Above 5 options may conflict with MIPL.
If you are not sure, use the recommended setting.

[root@localhost mipv6-2.0-rc3]# _

```

Deverá ser editado mais uma vez o menu de configuração, seleccionando as seguintes opções com “*:”

```

< > PF_KEY sockets
[*] PF_KEY migration interface (EXPERIMENTAL)

```

```
<*> The IPv6 protocol (EXPERIMENTAL)
[*] IPv6: Privacy Extensions (RFC 3041) support
<*> IPv6: AH transformation
<*> IPv6: ESP transformation
<*> IPv6: IPComp transformation
--- IPv6: tunnel transformation
<*> IPv6: IPv6-in-IPv6 tunnel
[*] IPv6: advanced router (EXPERIMENTAL)
[*] IPv6: policy routing
[*] IPv6: source address based routing
[*] IPv6: Mobility (EXPERIMENTAL)
[*] IPv6: Moblity Debug Message
[*] Network packet filtering (replaces ipchains) --->
<*> Transformation user configuration interface
```

Após gravar estas alterações, o output do script deverá ser o seguinte:

```
Checking kernel configuration...
Using /mipv6/linux-2.6.11/.config

All kernel options are as they should.

[root@localhost mipv6-2.0-rc3]#
```

Caso não seja, deverá ser revista a configuração, alterando-a em conformidade com o resultado do output.

8. Neste ponto, o kernel estará pronto a ser compilado e instalado.

Dica: Para distinguir facilmente este kernel de outros, poderá ser alterada a variável "EXTRAVERSION" no `/usr/src/linux/Makefile` para por exemplo "FC3-k2.6.11-MIPv6-2.0-rc3".

```
# cd /usr/src/linux-2.6.11
# make clean
# make dep
# make bzImage
# make modules
# make modules_install
# make install
```

9. Se tudo correu bem, ótimo!! Poderá então editar o `/boot/grub/grub.conf` e fazer o reboot. Se durante algum dos passos ocorreu algum erro, então será mais complicado...os passos anteriores deverão ser revistos com cuidado.

Ler o [Linux Kernel HOWTO](#) para instruções detalhadas em como aplicar o patch, compilar e instalar o novo kernel.

3.2. Userspace tools

As userspace tools mip6d (em versões anteriores o mip6d não existia, existindo sim o mipdiag), os ficheiros de configuração e os init scripts devem ser instalados para obter as funcionalidades da mobilidade IPv6.

```
# cd /usr/local/src/mipv6-2.0-rc3
# CPPFLAGS='-isystem /usr/src/linux-2.6.11/include' ./configure
# make && make install
```

De notar que para executar o ./configure tem de ser dada a localização do novo kernel.

3.3. MIPv6 device node

O módulo MIPv6 necessita de uma nova entrada de nó no dispositivo. Usar o comando:

```
# mknod /dev/mipv6_dev c 0xf9 0
```

3.4. Arranque automático no boot

1. *Fedora core 3:*

No HOWTO original está descrito um método para iniciar o MIPv6 automaticamente no boot, mas uma vez que existem configurações a fazer antes de iniciar o daemon, o arranque automático no boot será feito de outra maneira.

Será criado um script com todas as configurações a fazer. A título de exemplo, poderá ser chamado de mipv6_ConfFile, e poderá ficar na directoria /mipv6.

Então poderá ser adicionado ao ficheiro /etc/rc.local a seguinte linha de código:

```
cd /mipv6; ./mipv6_ConfFile
```

O script rc.local é executado no boot após todos os scripts de configurações terem sido executados.

2. *Outras distribuições:*

Não foram testadas outras distribuições para além do fedora core 3. No HowTo original, são apresentados comandos para o Debian e para o Slackware, no entanto não é possível dizer que estes estão desactualizados. No entanto tal como no caso apresentado para o FC3, existem configurações a fazer, pelo que pode ser usado um método igual ao apresentado.

De referir ainda que a partir da versão 2 do MIPL, para iniciar o mipv6 já não usa o mobile-ip6, mas sim o daemon mip6d.

4. Test bed

Neste momento deverá existir um kernel MIPL funcional com um patch correctamente aplicado, userlevel tools instaladas e activação automática no arranque. Se alguma coisa correu mal, deverá verificar as secções anteriores com atenção.

4.1. Testcase

Os endereços usados no MIPL HowTo são site-local. No entanto, estes endereços já não existem. Quem faz a gestão dos endereços IPv6 é a IANA (Internet Assigned Numbers Authority), e em <http://www.iana.org/assignments/ipv6-address-space> é possível verificar como estão alocados actualmente. A tabela seguinte mostra os prefixos que se encontram alocados. À data deste documento todos os outros se encontravam reservados.

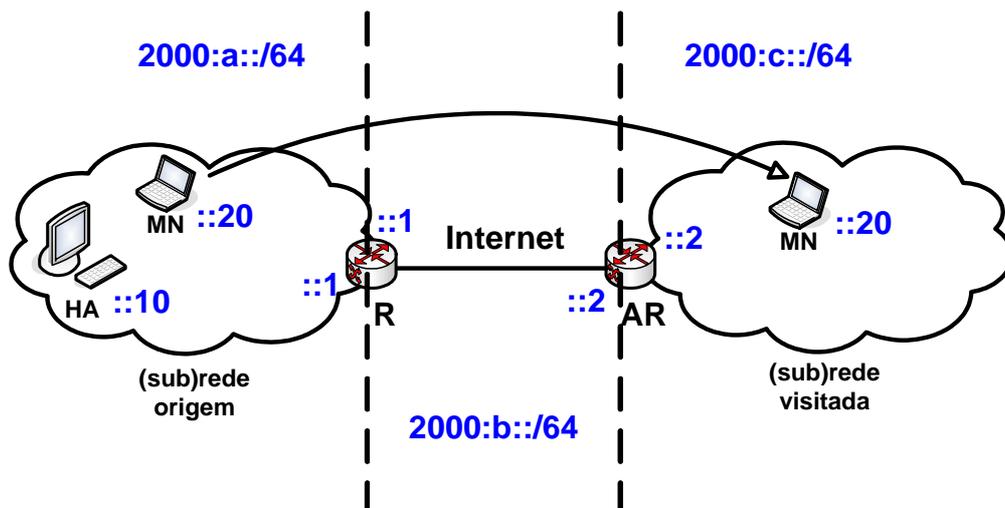
Prefixo IPv6	Alocação	Referência
2000::/3	Global Unicast	[RFC3513]
FC00::/7	Unique Local Unicast	[RFC4193]
FE80::/10	Link Local Unicast	[RFC3513]
FF00::/8	Multicast	[RFC3513]

Alocação de endereços pela IANA [last updated 05 October 2005].

Os endereços a usar nesta test-bed serão do tipo Global. Os Unique Local Unicast também não estão pensados para serem encaminhados na Internet global, ficando limitados a uma área mais restrita tal como um site. Os multicast, como é óbvio também não podem ser usados.

O cenário de teste consiste em 4 nós:

1. *HA - Home Agent*: O HA está localizado na rede origem com o endereço 2000:a::10.
2. *MN - Mobile Node*: Quando o MN está na rede origem, tem o endereço 2000:a::211:11ff:fe59:e99. Quando se move para uma rede destino, gera automaticamente um novo care-of-address, e.g., se for para a rede 2000:c::/64 irá ficar com o CoA 2000:c::211:11ff:fe59:e99.
3. *R - Router*: Este é o router que interliga a rede origem com a Internet. Tem uma interface na rede origem com o endereço 2000:a::1, e a outra interface para o exterior com o endereço 2000:b::1.
4. *AR - Access Router*: O link entre o AR e o R é a suposta Internet, bastando para o caso de teste usar um cabo cruzado. O AR tem duas interfaces, uma que liga à "Internet" com o endereço 2000:b::2, e a outra na rede destino com o endereço 2000:c::2.



Mobile IPv6 testbed.

4.2. Configuração Passo-a-Passo

Para o cenário anterior, foi usado apenas um cabo cruzado para simular a Internet, e foi usado um hub em cada rede, sendo necessário apenas mudar o cabo do terminal móvel de um hub para o outro para testar a mobilidade. No HowTo original foram usadas interfaces wireless nas redes origem e visitada, mas o princípio de funcionamento é o mesmo.

4.2.1. Configurar uma rede IPv6 funcional

Antes de começar a testar a mobilidade IPv6, é necessário configurar a rede com encaminhamento totalmente funcional. Todos os nós deverão estar acessíveis (para tal poderá testar-se com o utilitário ping). Se por exemplo, o AR não conseguir alcançar o HA, então não haverá troca de mensagens MIPv6.

Será dada uma breve instrução dos comandos necessários para configurar a rede com IPv6. Para mais informações sobre configurações IPv6, poderá ser consultado o HowTo de Peter Bieringer's, [Linux IPv6 HOWTO](#). Para configurar as interfaces serão usados os seguintes comandos:

```
#/sbin/ip link set dev <interface> {up|down}
#/sbin/ifconfig <interface> {up|down}

#/sbin/ip -6 addr add <ipv6_address>/<prefixlength> dev <interface>
#/sbin/ifconfig <interface> inet6 add <ipv6_address>/<prefixlength>
```

Para configurar as rotas:

```
#ip -6 route flush all
#ip -6 route add <ipv6_address>/<prefixlength> via <gateway>
#route -A inet6 add <ipv6_address>/<prefixlength> gw <gateway>
```

1. *R*: O router da rede origem tem duas interfaces, uma para a “Internet e outra para a rede origem. Deve, por isso, ter activo o encaminhamento (forwarding).

```
#Interface eth0
ip link set dev eth0 down
ip link set dev eth0 up
ip -6 addr add 2000:a::1/64 dev eth0

#Interface eth1
/sbin/ifconfig eth1 down

#Interface eth2
/sbin/ifconfig eth2 down
/sbin/ifconfig eth2 up
/sbin/ip -6 addr add 2000:b::1/64 dev eth2

#Activate IPv6 forwarding
/sbin/sysctl -w net.ipv6.conf.all.forwarding=1
/sbin/sysctl -w net.ipv6.conf.all.autoconf=0
/sbin/sysctl -w net.ipv6.conf.all.accept_ra=0
/sbin/sysctl -w net.ipv6.conf.all.accept_redirects=0

#Configuring Routes
ip -6 route flush all
ip -6 route add ::/0 via 2000:b::2
```

2. *AR*: O router de acesso (na rede visitada), também tem duas interfaces, uma para a Internet, e a outra para a sua rede. Também neste o encaminhamento deve estar activo.

```
#interface eth0
/sbin/ifconfig eth0 down
/sbin/ifconfig eth0 up
/sbin/ip -6 addr add 2000:b::2/64 dev eth0

#interface eth1 - realtek
ip link set dev eth1 down
ip link set dev eth1 up
/sbin/ip -6 addr add 2000:c::2/64 dev eth1

#Activate IPForwarding
/sbin/sysctl -w net.ipv6.conf.all.forwarding=1
/sbin/sysctl -w net.ipv6.conf.all.autoconf=0
/sbin/sysctl -w net.ipv6.conf.all.accept_ra=0
/sbin/sysctl -w net.ipv6.conf.all.accept_redirects=0

#Routes
ip -6 route flush all
ip -6 route add ::/0 via 2000:b::1
```

3. *HA*: O Home Agent tem uma interface. Deverá ter o forwarding activo, porque ele usa encaminhamento normal para reenviar os pacotes capturados na interface física para a interface túnel virtual.

```
#interface eth0 - Intel
/sbin/ifconfig eth0 down
/sbin/ifconfig eth0 up
```

```

/sbin/ip -6 addr add 2000:a::10/64 dev eth0

#Activate IPForwarding
/sbin/sysctl -w net.ipv6.conf.all.forwarding=1
/sbin/sysctl -w net.ipv6.conf.all.autoconf=0
/sbin/sysctl -w net.ipv6.conf.all.accept_ra=0
/sbin/sysctl -w net.ipv6.conf.all.accept_redirects=0

#Routes
ip -6 route flush all
ip -6 route add ::/0 via 2000:a::1

```

4. *MN*: O nó móvel possui apenas uma interface de rede. O encaminhamento deve ser desligado, mas deverá aceitar auto configuração e router advertisements.

```

#interface eth1 - Intel
#/sbin/ifconfig eth1 down
/sbin/ifconfig eth1 up
/sbin/ip -6 addr add 2000:a::20/64 dev eth1

#Activate IPForwarding
/sbin/sysctl -w net.ipv6.conf.all.forwarding=0
/sbin/sysctl -w net.ipv6.conf.all.autoconf=1
/sbin/sysctl -w net.ipv6.conf.all.accept_ra=1
/sbin/sysctl -w net.ipv6.conf.all.accept_redirects=1

#Routes
ip -6 route flush all
ip -6 route add ::/0 via 2000:a::1

```

É preferível o uso do sysctl a mudar as variáveis do proc (echo "0" > /proc/sys/net/ipv6/conf/eth0/), pois o sysctl muda os parâmetros do kernel em runtime.

Se no cenário forem usados APs nas redes origem e visitada, os terminais deverão usar o seguinte comando de modo a garantir conectividade com estes:

```
iwconfig eth0 mode managed essid <ESSID> enc off
```

A encriptação é desligada para efeitos de teste. Outras configurações no terminal poderão ser necessárias em função das configurações do AP, ver “man iwconfig”.

Se no cenário forem usados placas de rede unicamente, então deverá ser usado:

```
iwconfig eth0 mode ad-hoc essid <ESSID> enc off
```

Nota: Neste momento, com os endereços IPv6 e rotas estáticas configurados, deverá existir conectividade entre todos os nós. Poderá ser usado o ping6 para verificar esta condição.

4.2.2. Configuração da Mobilidade IPv6

A última configuração é a do ficheiro `mip6d.conf`. Se não for dito nada ao `mip6d`, ele procura o ficheiro de configuração em `"/usr/local/etc/mip6d.conf"`, pode no entanto ficar noutra localização, indicando ao `mip6d` qual `"mip6d -c <dir>/<fileName>".` Este ficheiro não é criado no momento da instalação, portanto terá de ser criado manualmente. Na pasta extra das `userspace tools` encontram-se 3 exemplos de configuração do CN, HA e MN. Estes ficheiros também se encontram em anexo neste HOWTO.

Para auxiliar a tarefa de configuração poderão ser consultadas as páginas do manual electrónico:

- Man `mipv6`
- Man `mip6d`
- Man `mip6d.conf`

1. CN: O CN poderá ser o AR, e deverá ter no ficheiro de configuração os seguintes parâmetros.

```
# This is an example of mip6d Correspondent Node configuration file

NodeConfig CN;

## If set to > 0, will not detach from tty
DebugLevel 0;

## Support route optimization with MNs
DoRouteOptimizationCN enabled;
```

2. HA: O ficheiro de configuração do HA deverá ter os seguintes parâmetros.

```
# This is an example of mip6d Mobile Node configuration file

#Common Options
#####

#Set the mode witch the daemon will run
NodeConfig HA;

## Support route optimization with other MNs
DoRouteOptimizationCN enabled;

#Options Specific to Home Agent and Mobile Node
#####

#Specifies the interface and options. If no options "Interface "eth0"
Interface "eth0";

#We won't use IPSec
UseMnHaIPsec disabled;

#Home Agent Specific Options
#####

#END
```

3. *MN*: O ficheiro de configuração do MN deverá ter os seguintes parâmetros.

```
# This is an example of mip6d Mobile Node configuration file

#Common Options
#####

#Set the mode witch the daemon will run
NodeConfig MN;

## Support route optimization with other MNs
DoRouteOptimizationCN enabled;

#Options to Home Agent e Mobile Node
#####

#Specifies the interface and options. If no options "Interface "eth0"
Interface "eth1";

#
UseMnHaIPsec disabled;

#Mobile Node Specific Options
#####

#
SendMobPfxSols enabled;

## Use route optimization with CNS
DoRouteOptimizationMN enabled;

#
UseCnBuAck enabled; #Default disabled;

MnHomeLink "eth1" {

    HomeAddress 2000:a::20/64;

    HomeAgentAddress 2000:a::10; #Default ::

}
#END
#####
```

4. De seguida inicie o mip6d no CN, HA e MN (a linha de código poderá ser adicionada aos scripts apresentados no ponto 4.2.1.):

```
mip6d -c /usr/local/etc/mip6d.conf
```

5. Para ver o output do debug na consola, usar a opção `-d` com um valor de 1 a 10 (0 não mostra nada, maior que 1 bloqueia a consola. Não é erro! É suposto acontecer! Depois são mostradas mensagens de debug na sequência de eventos):

```
mip6d -d<num> #num={0;10}
```


Ler a [RFC2462](#) para mais detalhes sobre a configuração Stateless (IPv6 Stateless Address Autoconfiguration).

Para configurar o RADVD na interface do AR, deverá ser editado primeiro o ficheiro `/etc/radvd.conf`. Um exemplo dessa configuração é apresentada de seguida:

```
# cat /etc/radvd.conf
interface eth1
{
    AdvSendAdvert on;
    AdvIntervalOpt on;

# These settings cause advertisements to be sent every 3-10 seconds.
    MinRtrAdvInterval 1;
    MaxRtrAdvInterval 3;

    AdvHomeAgentFlag off;

    prefix 2000:c::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

Após editado o ficheiro, pode-se iniciar o serviço:

```
# /etc/init.d/radvd start
```

Após iniciar o daemon `radvd`, poderá ser usado o `ethereal` para visualizar se as mensagens estão a ser enviadas periodicamente, ou então recorrer ao comando **radvdump**:

```
[root@localhost ~]# radvdump
Router advertisement from fe80::230:4fff:fe0a:49a0 (hoplimit 255)
Received by interface eth1
    # Note: {Min,Max}RtrAdvInterval cannot be obtained with
radvdump
    AdvCurHopLimit: 64
    AdvManagedFlag: off
    AdvOtherConfigFlag: off
    AdvHomeAgentFlag: off
    AdvReachableTime: 0
    AdvRetransTimer: 0
    Prefix 2000:c::/64
        AdvValidLifetime: 2592000
        AdvPreferredLifetime: 604800
        AdvOnLink: on
        AdvAutonomous: on
        AdvRouterAddr: on
    AdvSourceLLAddress: 00 30 4F 0A 49 A0
    AdvIntervalOpt:
        AdvInterval: 10
```

Nota! Quando é usado o radvd no HA e é activada a auto configuração no MN, então a interface deste irá configurar um endereço por auto configuração (que será supérfluo) para além do endereço estático.

4.2.4. Configurar radvd no HA

Para permitir que o MN saiba quando regressa à rede origem, deverá existir nesta uma entidade que envia RAs, que neste caso será o HA. Deverá então ser activado o RADVD no HA, mas não sem antes editar o ficheiro /etc/radvd.conf.

```
interface eth0
{
    AdvSendAdvert on;
    AdvIntervalOpt on;

# These settings cause advertisements to be sent every 1-3 seconds.

    MinRtrAdvInterval 1;
    MaxRtrAdvInterval 3;

    AdvHomeAgentFlag on;

HomeAgentLifetime 10000;
HomeAgentPreference 20;
AdvHomeAgentInfo on;

    prefix 2000:a::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
        AdvPreferredLifetime 500;
        AdvValidLifetime 600;
    };
};
```

Uma vez mais o ethereal ou o **radvdump** no HA para verificar o envio dos RAs.

```
# radvdump
Router advertisement from fe80::202:2dff:fe54:d11e (hoplimit 255)
Received by interface eth0
# Note: {Min,Max}RtrAdvInterval cannot be obtained with
radvdump
AdvCurHopLimit: 64
AdvManagedFlag: off
AdvOtherConfigFlag: off
AdvHomeAgentFlag: on
AdvReachableTime: 0
AdvRetransTimer: 0
Prefix fec0:106:2700::2/64
    AdvValidLifetime: 12000
    AdvPreferredLifetime: 10000
    AdvOnLink: on
    AdvAutonomous: on
    AdvRouterAddr: on
AdvSourceLLAddress: 00 02 2D 54 D1 1E
AdvHomeAgentInfo:
```

```
HomeAgentPreference: 20
HomeAgentLifetime: 1000
```

Após a configuração dos RAs no HA, poderá ser vista a alteração provocada no MN.

```
[root@localhost ~]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:11:11:59:0E:99
          inet6 addr: 2000:a::211:11ff:fe59:e99/64 Scope:Global❶
          inet6 addr: fe80::211:11ff:fe59:e99/64 Scope:Link❷
          inet6 addr: 2000:a::20/64 Scope:Global❸
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:924 errors:0 dropped:0 overruns:0 frame:0
          TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:93984 (91.7 KiB)  TX bytes:8770 (8.5 KiB)
          Base address:0xbc00 Memory:ff8e0000-ff900000
```

- ❶ O endereço supérfluo auto configurado. Uma vez que no MN a auto configuração foi activada e o HA envia RAs, o MN gera um novo endereço combinando o prefixo do HA e o seu próprio endereço MAC
- ❷ O endereço link-local address gerado no boot.
- ❸ O endereço IPv6 configurado estaticamente.

4.3. Configurar a interface do MN para definir o endereço estático

A interface poderá estar configurada para obter o endereço automaticamente via rede. Isso irá interferir com os testes, por isso tem de se mudar a configuração:

- Aceder ao menu de configuração das interfaces:
Menu>Network Device Control>Configure
- Seleccionar a interface de saída do MN e editar a sua configuração
Interface eth0>edit
- Seleccionar a opção:
Statically Set Ip Addresses>Ok

4.4. Desligar interfaces que não são usadas

Para prevenir resultados indesejados, aconselha-se a desligar todas as interfaces que não são usadas no cenário. Por exemplo, se o MN tiver duas interfaces de rede, então deverá ser desligado o que não é usado.

5. Testes

5.1. Teste primário

Após realizadas todas as configurações mostradas anteriormente é necessário realizar alguns testes para confirmar o funcionamento do mipv6. Se forem usados cenários wireless (com APs e placas wireless), devem ser configurados ESSIDs diferentes na rede origem e rede visitada.

Quando é iniciado o mip6d no MN, é possível verificar o envio de mensagens (router solicitations) multicast.

```
[root@localhost ~]# tcpdump -i eth1 -vv ip6 or proto ipv6
(...)
10:32:53.274337  :: > ff02::2: [icmp6 sum ok] icmp6: router
solicitation (len 8, hlim 255)
10:32:53.320868 fe80::230:4fff:fe0a:49a0 > ff02::1: icmp6: router
advertisement(chlim=64, pref=medium, router_ltime=9, reachable_time=0,
retrans_time=0)[ndp opt] (len 64, hlim 255)
10:32:53.322753  :: > ff02::16: HBH (rtalert: 0x0000) (padn)icmp6:
type-#143 [hlim 1] (len 56)
10:32:53.467738  :: > ff02::1:ff59:e99: [icmp6 sum ok] icmp6: neighbor
sol: who has 2000:c::211:11ff:fe59:e99 (len 24, hlim 255)
```

5.2. Detecção de movimento

A detecção genérica de movimento usa mensagens “Neighbor Unreachability Detection” para detectar quando é que o default router não está bi-direccionalmente alcançável, o que neste caso força o nó a descobrir um novo default router (geralmente num novo link).

Para verificar o que acontece, poderá ser usado o Ethereal para ver a troca de mensagens, e alguns comandos em diferentes janelas de consola, como por exemplo:

```
# watch ifconfig eth0
# watch route -A inet6
# tcpdump -i eth0 -vv ip6 or proto ipv6
```

Para se mover para uma nova rede, no caso do cenário com cabo, basta simplesmente mudar o cabo de um hub para outro. No caso do cenário wireless, pode ser desligado o AP da rede origem e ligado o da rede destino (processo não aconselhável) ou então pode ser indicado por comando ao MN para se ligar à outra rede

```
# iwconfig eth0 essid visitnet
```

O MN muda assim para outra rede, e uma vez que está a enviar "router solicitation" (multicast), o AR responde com o seu prefixo. O MN procede então à sua autoconfiguração, criando um novo endereço IPv6 a partir do prefixo recebido e do seu próprio MAC. O comando “ifconfig eth0” permite ver a o novo endereço.

```
[root@localhost ~]# ifconfig eth0
eth1      Link encap:Ethernet  HWaddr 00:11:11:59:0E:99
          inet6 addr: 2000:c::211:11ff:fe59:e99/64 Scope:Global1
          inet6 addr: 2000:a::211:11ff:fe59:e99/64 Scope:Global2
          inet6 addr: fe80::211:11ff:fe59:e99/64 Scope:Link3
          inet6 addr: 2000:a::20/64 Scope:Global4
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:924 errors:0 dropped:0 overruns:0 frame:0
          TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:93984 (91.7 KiB)  TX bytes:8770 (8.5 KiB)
          Base address:0xbc00 Memory:ff8e0000-ff900000
```

- ¹ O novo Care of Address, gerado combinando o MAC com o prefixo do AR.
- ² O endereço supérfluo da rede origem.
- ³ O endereço link local gerado no boot
- ⁴ O Home Address

Quase ao mesmo tempo, o MN irá enviar o binding update para o HA. Na janela tcpdump é possível ver vários pacotes enviados para o HA. O ethereal também pode ser usado para verificar o envio de mensagens quando existe movimento, como por exemplo os bindings updates enviados para o HA após configurar o CoA.

No. .	Time	Source	Destination	Protocol	Info
371	1219.039303	2000:c::211:11ff:fe59:e99	2000:a::10	MIPv6	Binding Update
372	1219.039303	2000:c::211:11ff:fe59:e99	2000:a::10	MIPv6	Binding Update

```

Frame 371 (110 bytes on wire (110 bytes captured)
Ethernet II, Src: Intel_59:0e:99 (00:11:11:59:0e:99), Dst: PlanetTe_0a:49:a0 (00:30:4f:0a:49:a0)
Internet Protocol Version 6
  Version: 6
  Traffic Class: 0x00
  Flowlabel: 0x00000
  Payload length: 56
  Next header: IPv6 destination option (0x3c)
  Hop limit: 64
  Source address: 2000:c::211:11ff:fe59:e99
  Destination address: 2000:a::10
Destination Option Header
  Next header: Mobile IPv6 (0x87)
  Length: 2 (24 bytes)
  PadN: 4 bytes
  Option Type: 201 (0xc9) - Home Address option
  Option Length: 16
  Home Address: 2000:a::20 (2000:a::20)
Mobile IPv6
  Payload protocol: IPv6 no next header (0x3b)
  Header length: 3 (32 bytes)
  Mobility Header Type: Binding update (5)
  Reserved: 0x00
  Checksum: 0x7141
  Binding Update
    Sequence number: 2224
    1... .. = Acknowledge (A) flag: Binding Acknowledgement requested
    ..1.. .. = Home Registration (H) flag: Home Registration
    ...0.. .. = Link-Local Compatibility (L) flag: No Link-Local Address Compatibility
    ...0.. .. = Key Management Compatibility (K) flag: No Key Management Mobility Compatibility
    Lifetime: 65535 (262140 seconds)
  Mobility Options
    PadN: 2 bytes
    Alternate care-of address: 2000:c::211:11ff:fe59:e99 (2000:c::211:11ff:fe59:e99)

```

Binding update enviado para o HA

É possível verificar que a mensagem tem origem no CoA do MN com destino ao HA. No Destination Option Header podemos verificar a indicação do Home Address. No cabeçalho Mobility options segue a indicação do CoA, que é igual ao source address do cabeçalho IPv6.

Na sequência do movimento foi enviado mais do que 1 pacote. Isto justifica-se pelo facto de o MN ter configurado mais do que um endereço da rede origem, porem isto não

está especificado na RFC e não devia acontecer, uma vez que só 1 desses 2 endereços é que é o HoA, e só será criado um túnel entre o MN e HA com base neste endereço.

No. -	Time	Source	Destination	Protocol	Info
371	1219.039303	2000:c::211:11ff:fe59:e99	2000:a::10	MIPv6	Binding Update
372	1219.039303	2000:c::211:11ff:fe59:e99	2000:a::10	MIPv6	Binding Update


```

Frame 372 (110 bytes on wire, 110 bytes captured)
Ethernet II, Src: Intel_59:0e:99 (00:11:11:59:0e:99), Dst: PlanetTe_0a:49:a0 (00:30:4f:0a:49:a0)
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 56
  Next header: IPv6 destination option (0x3c)
  Hop limit: 64
  Source address: 2000:c::211:11ff:fe59:e99
  Destination address: 2000:a::10
Destination Option Header
  Next header: Mobile IPv6 (0x87)
  Length: 2 (24 bytes)
  PadN: 4 bytes
  Option Type: 201 (0xc9) - Home Address Option
  Option Length: 16
  Home Address: 2000:a::211:11ff:fe59:e99 (2000:a::211:11ff:fe59:e99)
Mobile IPv6
  Payload protocol: IPv6 no next header (0x3b)
  Header length: 3 (32 bytes)
  Mobility Header Type: Binding Update (5)
  Reserved: 0x00
  Checksum: 0x9af5
  Binding Update
    Sequence number: 40472
    1... .... = Acknowledge (A) flag: Binding Acknowledgement requested
    .1. .... = Home Registration (H) flag: Home Registration
    ..1. .... = Link-Local Compatibility (L) flag: Link-Local Address Compatibility
    ...0 .... = Key Management Compatibility (K) flag: No Key Management Mobility Compatibility
    Lifetime: 65535 (262140 seconds)
  Mobility Options
    PadN: 2 bytes
    Alternate care-of address: 2000:c::211:11ff:fe59:e99 (2000:c::211:11ff:fe59:e99)
  
```

Binding update enviado para o HA

5.3. ping6

O comando ping6 pode ser usado para testar a conectividade antes durante e após o handover. O ideal será antes do handover colocar o MN a pingar a uma máquina, por exemplo a interface 1 do AR, e colocar esta máquina a pingar ao MN. Recorde-se que neste cenário a máquina AR foi configurada para funcionar também como CN. Esta configuração não é necessária para haver conectividade, mas sim para o processo de optimização de rotas. O uso do traceroute6 é desaconselhado.

No caso do cenário wired, haverá sempre perda de pacotes, dependendo da rapidez da mudança do cabo. No caso do cenário sem fios é suposto não haver perda de pacotes.

Ping do MN para o AR durante o handover:

```

[root@localhost ~]# ping6 2000:c::2
PING 2000:c::2(2000:c::2) 56 data bytes
64 bytes from 2000:c::2: icmp_seq=0 ttl=63 time=0.347 ms
64 bytes from 2000:c::2: icmp_seq=1 ttl=63 time=0.372 ms
ping: sendmsg: Invalid argument
ping: sendmsg: Invalid argument
From 2000:a::20 icmp_seq=7 Destination unreachable: Address unreachable
64 bytes from 2000:c::2: icmp_seq=8 ttl=62 time=0.584 ms
64 bytes from 2000:c::2: icmp_seq=9 ttl=62 time=0.612 ms
  
```

O Ping do AR para o MN também pode ser usado durante o handover para verificar que o MN está alcançável antes e depois. O ethereal pode ser usado para verificar que existe otimização de rotas entre o MN e o AR/CN (o host AR do cenário foi configurado anteriormente como CN).

Fazendo o ping do host R para o MN, uma vez que aquele não foi configurado como MN, é possível verificar que a comunicação é feita através do HA em ambos os sentidos.

Outro comando que pode ser usado para verificar o percurso dos pacotes é o `traceroute6`, mas nos testes com `mipv6` os resultados deste comando são muito inconsistentes devido ao uso de túneis, e por isso se desaconselha o seu uso.

5.4. Tabela de encaminhamento IP do Kernel

Algo interessante no MIPv6 é que ele muda a sua default route para o túnel que criou do MN para o HA:

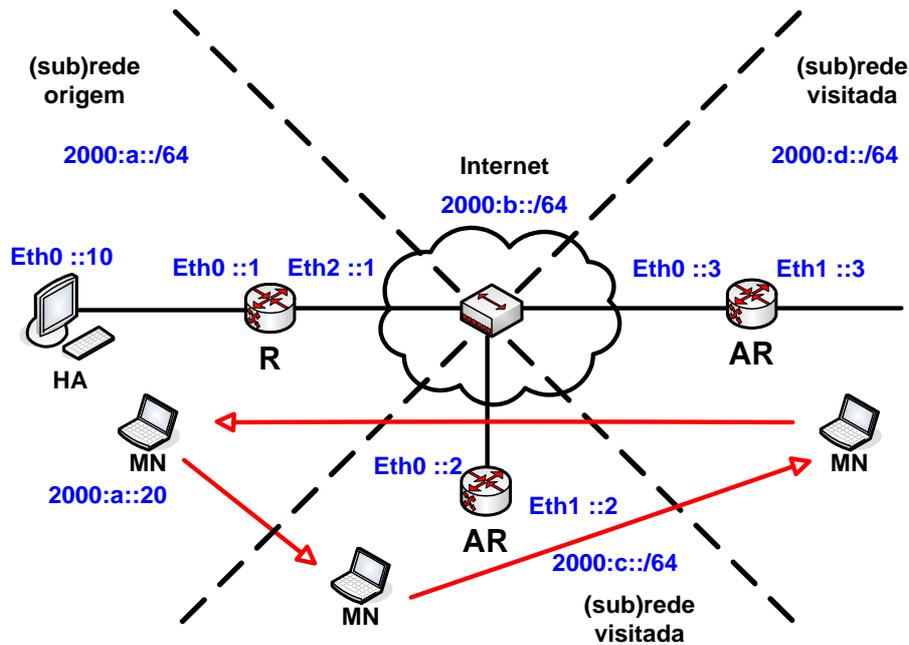
```
# route -A inet6
Kernel IPv6 routing table
Destination      Next Hop          Flags Metric Ref    Use Iface
::/0              ::                UD     64    0      0 ip6tnl1
```

Caso não mude, uma rota poderá ser adicionada manualmente:

```
# ip route ::/0 via dev ip6tnl
```

5.5. Mover-se através de várias redes

Mover-se ao longo de várias redes visitadas não é muito diferente do que viajar através de uma. Há que ter em conta é que por cada rede visitada será criado um novo endereço por auto configuração.



O MN a mover-se por várias redes..

1. O MN visita uma rede, auto configura um CoA e faz um binding update no HA.
2. O MN move-se de uma rede visitada para outra e faz novo BU no HA.
3. O MN regressa à rede origem

O AR na rede D está configurado exactamente como o da rede C, excepto que está configurado noutra rede. No caso do uso de cenários com APs, terá de se configurar outro ESSID no AP desta rede diferente dos outros, ou no caso de se usarem interfaces wireless nos terminais terá de se atribuir um ESSID ao AR da rede D:

```
# iwconfig eth0 essid <ESSID da rede D>
```

5.6. Regresso à rede origem

Para fazer o regresso à rede origem, no caso do cenário wired basta ligar o cabo do MN ao hub da rede origem, e no caso do cenário wireless basta mudar o ESSID ao qual o MN está associado:

```
# iwconfig eth0 essid <ESSID da rede origem>
```

Como o HA envia periodicamente mensagens radvd com o bit HA activo (AdvHomeAgentFlag enable), o MN irá detectar que regressou à rede origem.

Poderá verificar-se que o MN se registou correctamente analisando a informação da binding cache do HA, que deverá estar vazia. Isto poderá ser feito através do output do Debug do HA.

5.7. Teste em comunicações - smooth handover

Para verificar a funcionalidade do MIPv6 numa comunicação real, poderá ser usado o GnomeMeeting (ver figura abaixo). Este programa permite estabelecer uma comunicação de voz e vídeo extremo-a-extremo.

Após instalar e configurar o programa em duas máquinas, poderá ser iniciada uma comunicação, e durante esta poderão ser feitas algumas mudanças de rede de modo a ver handovers suaves, mas só para o caso de um cenário wireless como é óbvio. A versão do GnomeMeeting terá de ter suporte de IPv6.



Imagem do uso do gnomeMeeting para testar o roaming entre as redes wireless.

5.7. Outros Testes

Outros testes poderão ser feitos usando aplicações para gerar tráfego e medir os parâmetros da comunicação, tais como o MGEN e o TRPR.

6. FAQ

Esta FAQ foi construída com base na existente no HowTo original, foi actualizada, e foram acrescentados novos tópicos com base na mailing list oficial do MIPv6 e em diversas pesquisas feitas no âmbito deste trabalho.

1. *Q: Porque é que é necessário criar a entrada /dev/mipv6_dev?*

A: O ficheiro dev serve, principalmente, para que a ferramenta mipv6 possa fazer modificações aos parâmetros do kernel, usando chamadas ioctl através do ficheiro do dispositivo. O mknod cria um ficheiro de dispositivo especial com parâmetros reconhecíveis pelo mipv6.

2. *Q: O MIPL suporta IPsec?*

A: A partir da versão 2.6 do MIPL, que foi implementado suporte para IPsec de raiz. A [RFC 3776] define este processo.

3. *Q: Como se pode controlar os parâmetros de mobilidade tais como o uso de IPsec na comunicação ou o uso ou não de optimização de rotas?*

DoRouteOptimizationMN enabled;
UseMnHaIPsec enabled;
(...) Ver man mip6d.conf

4. *Q: Será que diferentes redes wireless podem ter diferentes chaves WEP?*

A: Sim, mas terá de ser mudado aquando da mudança para a nova rede, uma vez que o MIPv6 do MIPL não consegue realizar este processo automaticamente.

5. *Q: Se o MN viajar ao longo de várias LANs, regressando depois a casa, a interface irá guardar todos os endereços IPv6 auto configurados de todas as redes que visitou? Haverá uma maneira de remover estes endereços?*

A: Os endereços têm um tempo de vida definido nos router advertisements, findo o qual são eliminados. Também poderão ser eliminados manualmente:

```
# ifconfig eth0 inet6 del <ipv6-address>/<prefix-length>
```

6. *Q: A máquina B tem duas interfaces com duas sub redes diferentes atribuídas. Quando a máquina A pinga a máquina B, esta não responde. A máquina A sabe onde a máquina B está!*

A: A máquina B poderá não saber onde está a máquina/rede B. Deverá ser adicionada uma rota:

```
# ip route add 2000:a::1/64 via 2000:c::2
```

ou

```
# route -A inet6 add 2000:a::1/64 gw 2000:c::2 dev eth0
```

7. *Q: Como se configura um default gateway em IPv6?*

A: Usando a tradicional rota:

```
# route -A inet6 add default gw <ipv6-host>
```

Ou o commando ip:

```
# ip -6 route add ::/0 via <ipv6-host>
```

8. *Q: Porque é que o terminal envia um endereço multicast em vez do anycast, quando envia o router solicitation?*

A: Porque quer uma resposta de todos os routers, não de qualquer um. A ideia é receber todos os parâmetros e escolher o melhor default router.

9. *Q: Porque é que o MN por vezes não consegue detectar o movimento?*

A: Ele pensa que o seu router anterior ainda está alcançável. Isto resulta de tempos de vida elevados dos router advertisements. Estes valores poderão ser manipulados, editando o ficheiro /etc/radvd.conf (nos routers que enviam os RA, e.g. no caso do cenário serão os routers AR e HA), alterando os valores mínimos e máximos dos intervalos de envio destes. Consultar man radvd.conf para mais informação.

10. *Q: Posso usar endereços do tipo a::/64 de modo a simplificar e poupar tempo nos testes?*

A: Não, não deverão ser usados. Apesar de ser possível configurar as interfaces com estes endereços, não será possível configurar o gateway. Os comandos :

```
# route -A inet6 add default gw <ipv6-host>
```

```
# ip -6 route add ::/0 via <ipv6-host>
```

não suportam este tipo de endereços, apenas os começados pelos prefixos alocados pela IANA (<http://www.iana.org/assignments/ipv6-address-space>), nomeadamente endereços globais 2000::/3.

11. *Q: Posso usar endereços site-local, com prefixo fec0::64?*

A.: Os endereços FEC0::/10 foram anteriormente definidos como Site-local, no entanto esta definição foi destituída em Setembro de 2004 pela RFC 3879.

12. *Q: O libnetlink é distribuído com o MIPL?*

A: Uma versão modificada do libnetlink é distribuída com o MIPL. Esta versão não inclui todas as funcionalidades da package da libnetlink, apenas as necessárias para o funcionamento do MIPL.

13. Existem algumas implementações disponíveis para testar ou ver o modo de operação do MIPv6?

A: Na mailing List vêm referenciados algumas aplicações desenvolvidas para o efeito

<http://www.bullopensource.org/mipv6/index.php>

<http://www.cavone.com/mipv6-analyzer/>

14. Q: No debug aparece o erro “DoD failed” quando existe o handover, e a mobilidade não funciona!

A: O MIPL usado vai no 3º pré-lançamento antes de sair a versão final, portanto poderão existir alguns erros, falhas ou alguns processos a melhorar. Mudando o HoA na interface do MN e no ficheiro mip6d.conf é o suficiente para voltar a testar a mobilidade

15. Existem implementações publicas do RFC 4068 "Fast Handovers for Mobile IPv6" e/ou RFC 4140 "Hierarchical Mobile IPv6 Mobility Management HMIPv6)".

A: Ver os sítios:

<http://www.ctie.monash.edu.au/ipv6/>

<http://www.tkn.tu-berlin.de/research/hmip/>

7. Recursos

1. Mobile IPv6 for Linux <http://www.mobile-ipv6.org/>
2. Linux Mobile IPv6 HowTo <http://www.tldp.org/HOWTO/Mobile-IPv6-HOWTO/>
3. Mobile IPv6 Working Group (IETF) <http://www.ietf.org/html.charters/mip6-charter.html>
4. IPv6 Working Group (IETF) <http://www.ietf.org/html.charters/ipv6-charter.html>
5. RFC 3775 Mobility for IPv6 <http://www.ietf.org/rfc/rfc3775.txt?number=3775>
6. RFC 3776 Using IPsec to Protect MIPv6 Signaling Between MN and HA <http://www.ietf.org/rfc/rfc3775.txt?number=3775>
7. RFC 2460 Internet Protocol, Version 6 (IPv6) Specification <http://www.ietf.org/rfc/rfc2460.txt>
8. RFC2461 Neighbor Discovery for IP Version 6 (IPv6) <http://www.ietf.org/rfc/rfc2461.txt>
9. RFC2462 IPv6 Stateless Address Autoconfiguration <http://www.ietf.org/rfc/rfc2462.txt>
10. Peter Bieringer's Linux IPv6 HOWTO (en) <http://ldp.linux.no/HOWTO/Linux+IPv6-HOWTO/>
11. Current Status of IPv6 Support for Networking Applications http://www.deepspace6.net/docs/ipv6_status_page_apps.html
12. MIPL MailingList Archives <http://www.mobile-ipv6.org/pipermail/mipl/>
13. Endereçamento IPv6 <http://www.iana.org/assignments/ipv6-address-space>

9. Anexos

Neste ponto são colocados alguns anexos úteis para a configuração do mipl.

9.1. Configuração do mip6d.conf

Nesta secção são mostrados exemplos de configuração do ficheiro “/usr/local/etc/mip6d.conf” para as entidades CN, MN e HA.

9.1.1. Exemplo de configuração do CN

```
#ficheiro /usr/local/etc/mip6d.conf
# This is an example of mip6d Correspondent Node configuration
file

NodeConfig CN;

## If set to > 0, will not detach from tty
DebugLevel 10;

## Support route optimization with MNs
DoRouteOptimizationCN enabled;
```

9.1.1. Exemplo de configuração do HA

```
#ficheiro /usr/local/etc/mip6d.conf
# This is an example of mip6d Home Agent configuration file

NodeConfig HA;

## If set to > 0, will not detach from tty
DebugLevel 10;

## List of interfaces where we serve as Home Agent
Interface "eth0";
Interface "eth1";

##
## IPsec configuration
##

UseMnHaIPsec enabled;

## Key Management Mobility Capability
#KeyMngMobCapability disabled;

IPsecPolicySet {
    HomeAgentAddress 3ffe:2620:6:1::1;

    HomeAddress 3ffe:2620:6:1::1234/64;
    HomeAddress 3ffe:2620:6:1::1235/64;
```

```

    IPsecPolicy HomeRegBinding UseESP;
    IPsecPolicy MobPfxDisc UseESP;
    IPsecPolicy TunnelMh UseESP;
}

##
## It is possible to specify multiple IPsecPolicySet in order to configure
## different value for such address.
#IPsecPolicySet {
#
## One HA is for one IPsecPolicySet.
#   HomeAgentAddress 3ffe:2620:6:1::1;
#
## It is possible to specify multiple home addresses when they use
## the same configuration.
#   HomeAddress 3ffe:2620:6:1::1236/64;
#   HomeAddress 3ffe:2620:6:1::1237/64;
#
## BU/BA protection
#   ## IPsec protocols;
#   ## Choose one of below when using with tunnel protection like
#   ## HoTI/HoT or Tunnel payload. Otherwise, multiple protocols
#   ## can be specified.
#   IPsecPolicy HomeRegBinding UseESP;
#   IPsecPolicy HomeRegBinding UseAH;
#
## MPD protection
#   IPsecPolicy MobPfxDisc UseESP;
#
## Tunnel HoTI/HoT protection
#   IPsecPolicy TunnelMh UseESP;
#   ## To protect only HoTI/HoT exactly in MH, use below instead.
#   #IPsecPolicy TunnelHomeTesting UseESP;
#
#   ## Tunnel payload protection
#   #IPsecPolicy TunnelPayload UseESP;
#}

```

9.1.1. Exemplo de configuração do MN

```

#ficheiro /usr/local/etc/mip6d.conf
# This is an example of mip6d Mobile Node configuration file

NodeConfig MN;

## If set to > 0, will not detach from tty
DebugLevel 10;

## Support route optimization with other MNs
DoRouteOptimizationCN enabled;

## Use route optimization with CNs
DoRouteOptimizationMN enabled;

```

```

UseCnBuAck disabled;

Interface "eth0" {
    MnIfPreference -1;
}
Interface "eth1" {
    MnIfPreference 1;
}

MnRouterProbesRA 1;
MnRouterProbesLinkUp 0;

MnHomeLink "eth0" {
    HomeAgentAddress 3ffe:2620:6:1::1;
    HomeAddress 3ffe:2620:6:1::1234/64;

    #          proto  type  code  route opt.
#   MnRoPolicy ICMP  129   0     enabled;
#   MnRoPolicy ICMP  any   any   disabled;
#          #      proto  dst   src   route opt.
#   MnRoPolicy TCP   80   any   any   enabled;
#   MnRoPolicy TCP   any   any   disabled;
}

##
## IPsec configuration
##

UseMnHaIPsec enabled;

## Key Management Mobility Capability
KeyMngMobCapability disabled;

IPsecPolicySet {
    HomeAgentAddress 3ffe:2620:6:1::1;
    HomeAddress 3ffe:2620:6:1::1234/64;

    IPsecPolicy HomeRegBinding UseESP;
    IPsecPolicy MobPfxDisc UseESP;
    IPsecPolicy TunnelMh UseESP;
}

```

9.2. Man do MIPL

9.1.1. man mipv6

```

MIPV6(7)                               Linux Programmer's Manual
MIPV6(7)

NAME
    mipv6 - Linux Mobile IPv6 protocol extensions

SYNOPSIS
    #include <netinet/in.h>

```

```
#include <netinet/ip6.h>
#include <netinet/icmp6.h>
#include <netinet/ip6mh.h>
```

DESCRIPTION

Linux 2.6 optionally implements extensions for Mobile IPv6 protocol in the IPv6 stack. This man page describes extensions to the IPv6 API, proposed in draft-chakrabarti-mobileip-mipext-advapi-02.

Linux Man Page

2003-12-24

MIPv6(7)

9.1.1. man mip6d

mip6d(1) Mobile IPv6 Daemon mip6d(1)

NAME

mip6d - MIPL Mobile IPv6 protocol implementation

SYNOPSIS

mip6d [options]

DESCRIPTION

Mobile IPv6 implementation

OPTIONS

-V, --version
Display version information and copyright

-, -h, --help
Display this help text

-c <file>
Read configuration from <file>. If option is not present, default path (/usr/local/etc/mip6d.conf) is used.

These options override values read from config file:

-d <number>
Set debug level (0-10)

-C, --correspondent-node
Node is CN

-H, --home-agent
Node is HA

-M, --mobile-node
Node is MN

FILES

/usr/local/etc/mip6d.conf

SEE ALSO

mip6d.conf(5), mipv6(7),
RFC3775: Mobility Support in IPv6,
RFC3776: Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents

May 16, 2005

mip6d(1)

(END)

9.1.1. man mip6d.conf

```
mip6d.conf(5)           Mobile IPv6 Daemon Configuration           mip6d.conf(5)

NAME
    mip6d.conf - MIPL Mobile IPv6 Configuration file

SYNOPSIS
    /usr/local/etc/mip6d.conf

DESCRIPTION
    MIPL Mobile IPv6 daemon's configuration file

    Below is a list of currently supported configuration options. All configuration lines are terminated with a semicolon. Sub-sections are enclosed in { and }. Strings are quoted with double quotes.

COMMON OPTIONS
    The file contains the following common definitions:

    NodeConfig CN | HA | MN;

        Indicates if the daemon should run in Correspondent Node, Home Agent or Mobile Node mode.

        Default: CN

    DebugLevel number;

        Indicates the debug level of the daemon. If the value is greater than zero, the daemon will not detach from tty (i.e. debug messages will be printed on the controlling tty).

        Default: 0

    DoRouteOptimizationCN boolean;

        Indicates if a node should participate in route optimization with a Mobile Node.

        Default: enabled

    NonVolatileBindingCache boolean;

        This option is currently ignored. Binding cache is always stored in volatile memory, and is not retained between shutdown and startup.

OPTIONS COMMON TO HOME AGENT AND MOBILE NODE
    These options are used both in the Home Agent and Mobile Node:

    Interface name;

    Interface name {
        MnIfPreference number;
        IsRouter boolean;
        IfType CN | HA | MN;
    }

        Specifies an interface and options associated with it. If no options are present, Interface can be terminated with semi-colon. This is used for home agent to specify which interfaces are used for HA operation. For the home agent to function properly, a Router Advertisement daemon (e.g. radvd) must broadcast advertisements with the Home Agent bit and Home Agent Informa-
```

tion Option set on these interfaces. This option is also used by multihomed Mobile Nodes to define which interfaces are used by it.

Only option MnIfPreference is used with Mobile IPv6. It sets interface preference value for an interface, in a multi-homed mobile node. Preference is relative only to other preference values, and may be a signed number.

IsRouter and IfType are only used with NEMO.

UseMnHaIPsec boolean;

Indicates if the MN-HA MIPv6 signalling should be protected with IPsec.

Default: enabled

KeyMngMobCapability boolean;

If dynamic keying with MIPv6-aware IKE is used, this options should be enabled. It turns on the K-bit for binding updates and binding acknowledgements.

Default: disabled

```
IPsecPolicySet {
    HomeAgentAddress address;
    HomeAddress address/length;
    IPsecPolicy ...
    ...
}
```

IPsecPolicySet is a set of policies to apply for matching packets. A policy set can contain multiple HomeAddress options, but only one HomeAgentAddress option. For home agent, home agent address field contains its own address, and home address fields may contain any number of mobile nodes for which the same policy applies.

IPsecPolicy has the following format:

IPsecPolicy type UseESP UseAH UseIPComp;

Field type can be one of HomeRegBinding, MobPfxDisc, ICMP, any, TunnelMh, TunnelHomeTesting, or TunnelPayload. At least one of the IPsec protocols must be specified. The any option protects all transport mode communication between the MN and HA. Tunnel protection only allows exactly one protocol. Set IPsec protocol X with the "UseX" flag. Although not required, one can specify explicitly "NoX" (e.g. UseESP NoAH NoIPComp).

HOME AGENT SPECIFIC OPTIONS

The following definitions are ignored unless the node is configured as a HA:

SendMobPfxAdvs boolean;

Controls whether home agent sends Mobile Prefix Advertisements to mobile nodes in foreign networks.

SendUnsolMobPfxAdvs boolean;

Controls whether home agent send unsolicited Mobile Prefix Advertisements to mobile nodes in foreign networks.

MinMobPfxAdvInterval number;

Sets a minimum interval (in seconds) for Mobile Prefix Advertisements.

Default: 600

MaxMobPfxAdvInterval number;

Sets a maximum interval (in seconds) for Mobile Prefix Advertisements.

Default: 86400

MOBILE NODE SPECIFIC OPTIONS

The following definitions are ignored unless the node is configured as a MN:

SendMobPfxSols boolean;

Controls whether mobile node sends Mobile Prefix Solicitations to the home network.

DoRouteOptimizationMN boolean;

Indicates if the Mobile Node should initialize route optimization with Correspondent Nodes.

Default: enabled

UseCnBuAck boolean;

Indicates if the Acknowledge bit should be set in Binding Updates sent to Correspondent Nodes.

Default: disabled

MnRouterProbesRA number;

Indicates how many times the MN should send Neighbor Unreachability Detection probes to its old router after receiving a Router Advertisement from a new one. If the option is set to zero, the MN will move to the new router straight away.

Default: 0

MnRouterProbesLinkUp number;

Indicates how many times the MN should send Neighbor Unreachability Detection probes to its old router before initiating router discovery when the link goes up after being down.

Default: 0

MnRouterProbeTimeout decimal;

Indicates how long (in seconds) the MN should wait for a reply during a access router Neighbor Unreachability Detection probe. If this parameter isn't set, the Retransmit Timer value for

the

last link is used. The Retransmit Timer is typically defined as one second. If the parameter is set, the MN will use the minimum of MnRouterProbeTimeout and Retransmit Timer, thus allowing faster handoffs.

Default: 0

MnHomeLink name {
 HomeAddress address/length;
 HomeAgentAddress address;

```

MnRoPolicy ...
    ...
}

```

Each MnHomeLink definition has a name. This is the name (enclosed in double quotes) of the interface used for connecting to the physical home link. To set up multiple Home Addresses on the Mobile Node, you need to define multiple MnHomeLink structures. The interface names don't have to be unique in

these

definitions. All the home link specific definitions are detailed below:

```
HomeAddress address/length;
```

Address is an IPv6 address, and length the prefix length of the address, usually 64. This option must be included in a home link definition.

```
HomeAgentAddress address;
```

Agent.

Address is the IPv6 address of the Mobile Node's Home

DHAAD is used if it is the unspecified address ::.

Default: ::

The route optimization policies are of the form:

```
MnRoPolicy proto sport/type dport/code boolean;
```

Any number of these policies may be defined. If no policies are defined default behavior depends on the DoRouteOptimizationMN option.

The fields for a route optimization policy entry are as follows: proto is either a protocol number or one of the pre-defined protocols ICMP, TCP, UDP, or SCTP. sport/type is source port for all protocols using ports, and type for ICMP. dport/code is destination port, or code for ICMP. Last field boolean sets route optimization either enabled or disabled for packets matching this entry.

For sport/type and dport/code fields a wildcard value "any" is allowed.

EXAMPLES

A Correspondent Node example:

```

NodeConfig CN;
DoRouteOptimizationCN enabled;

```

A Home Agent example:

```

NodeConfig HA;
Interface "eth0";
Interface "eth1";
UseMnHaIPsec enabled;
IPsecPolicySet {
    HomeAgentAddress 3ffe:2620:6:1::1;
    HomeAddress 3ffe:2620:6:1::1234/64;
    HomeAddress 3ffe:2620:6:1::1235/64;
}

```

```

        IPsecPolicy HomeRegBinding UseESP;
        IPsecPolicy TunnelMh UseESP;
    }

```

A Mobile Node example:

```

NodeConfig MN;

DoRouteOptimizationCN enabled;

DoRouteOptimizationMN enabled;

UseCnBuAck enabled;

MnHomeLink "eth0" {
    HomeAgentAddress 3ffe:2620:6:1::1;
    HomeAddress 3ffe:2620:6:1::1234/64;

    #          proto  type  code  route opt.
MnRoPolicy   ICMP    any   any   disabled;
MnRoPolicy   ICMP    129   0     enabled;
    #          proto  dst   src   route opt.
MnRoPolicy   TCP     any   any   disabled;
MnRoPolicy   TCP     80    any   enabled;
}

UseMnHaIPsec enabled;

IPsecPolicySet {
    HomeAgentAddress 3ffe:2620:6:1::1;
    HomeAddress 3ffe:2620:6:1::1234/64;

    IPsecPolicy HomeRegBinding UseESP;
    IPsecPolicy TunnelMh UseESP;
}

```

SEE ALSO

mip6d(1), mipv6(7),

RFC3775: Mobility Support in IPv6,

RFC3776: Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents

May 16, 2005

mip6d.conf(5)

(END)

#####

#####

Usage: mip6d [options]

Options:

```

-V, --version          Display version information and copyright
-?, -h, --help        Display this help text
-c <file>             Read configuration from <file>

```

These options override values read from config file:

```

-d <number>          Set debug level (0-10)
-C, --correspondent-node Node is CN
-H, --home-agent     Node is HA
-M, --mobile-node    Node is MN

```

For bug reporting, see URL:<http://www.mobile-ipv6.org/bugs/>.

9.2. radvd.conf

9.1.1. Man radvd.conf

```
RADVD.CONF(5) RADVD.CONF(5)

NAME
    radvd.conf - configuration file of the router advertisement daemon
    radvd

DESCRIPTION
    This file describes the information which is included in the router
    advertisement (RA) of a specific interface.

    The file contains one or more interface definitions of the form:

    interface name {
        list of interface specific options
        list of prefix definitions
    };

    All the possible interface specific options are detailed below. Each
    option has to be terminated by a semicolon.

    Prefix definitions are of the form:

    prefix prefix/length {
        list of prefix specific options
    };

    Prefix can be network prefix or the address of the interface. The
    address of interface should be used when using Mobile IPv6 extensions.

    All the possible prefix specific options are described below. Each
    option has to be terminated by a semicolon.

    Decimal values are allowed only for MaxRtrAdvInterval and MinRtrAdvIn-
    terval. Decimal values should be used only when using Mobile IPv6
    extensions.

INTERFACE SPECIFIC OPTIONS
    AdvSendAdvert on|off

        A flag indicating whether or not the router sends periodic
        router advertisements and responds to router solicitations.

        This option no longer has to be specified first, but it needs to
        be on to enable advertisement on this interface.

        Default: off

    UnicastOnly on|off

        Indicates that the interface link type only supports unicast.
        This will prevent unsolicited advertisements from being sent,
        and will cause solicited advertisements to be unicast to the
        soliciting node. This option is necessary for non-broadcast,
        multiple-access links, such as ISATAP.

        Default: off

    MaxRtrAdvInterval seconds

        The maximum time allowed between sending unsolicited multicast
        router advertisements from the interface, in seconds.
```

Must be no less than 4 seconds and no greater than 1800 seconds.

Minimum when using Mobile IPv6 extensions: 1.5.

Default: 600 seconds

MinRtrAdvInterval seconds

The minimum time allowed between sending unsolicited multicast router advertisements from the interface, in seconds.

Must be no less than 3 seconds and no greater than $0.75 * \text{MaxRtrAdvInterval}$.

Minimum when using Mobile IPv6 extensions: 0.05.

Default: $0.33 * \text{MaxRtrAdvInterval}$

AdvManagedFlag on|off

When set, hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. The use of this flag is described in RFC 2462.

Default: off

AdvOtherConfigFlag on|off

When set, hosts use the administered (stateful) protocol for autoconfiguration of other (non-address) information. The use of this flag is described in RFC 2462.

Default: off

AdvLinkMTU integer

The MTU option is used in router advertisement messages to insure that all nodes on a link use the same MTU value in those cases where the link MTU is not well known.

If specified, i.e. not 0, must not be smaller than 1280 and not greater than the maximum MTU allowed for this link (e.g. ethernet has a maximum MTU of 1500. See RFC 2464).

Default: 0

AdvReachableTime milliseconds

The time, in milliseconds, that a node assumes a neighbor is reachable after having received a reachability confirmation. Used by the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 2461). A value of zero means unspecified (by this router).

Must be no greater than 3,600,000 milliseconds (1 hour).

Default: 0

AdvRetransTimer milliseconds

The time, in milliseconds, between retransmitted Neighbor Solicitation messages. Used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 2461). A value of zero means unspecified (by this router).

Default: 0

AdvCurHopLimit integer

The default value that should be placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The value should be set to the current diameter of the Internet. The value zero means unspecified (by this router).

Default: 64

AdvDefaultLifetime seconds

The lifetime associated with the default router in units of seconds. The maximum value corresponds to 18.2 hours. A lifetime of 0 indicates that the router is not a default router and should not appear on the default router list. The router lifetime applies only to the router's usefulness as a

default

router; it does not apply to information contained in other message fields or options. Options that need time limits for their information include their own lifetime fields.

Must be either zero or between MaxRtrAdvInterval and 9000 seconds.

Default: 3 * MaxRtrAdvInterval

AdvSourceLLAddress on|off

When set, the link-layer address of the outgoing interface is included in the RA.

Default: on

AdvHomeAgentFlag on|off

When set, indicates that sending router is able to serve as Mobile IPv6 Home Agent. When set, minimum limits specified by Mobile IPv6 are used for MinRtrAdvInterval and MaxRtrAdvInterval.

Default: off

AdvHomeAgentInfo on|off

When set, Home Agent Information Option (specified by Mobile IPv6) is included in Router Advertisements. AdvHomeAgentFlag must also be set when using this option.

Default: off

HomeAgentLifetime seconds

The length of time in seconds (relative to the time the packet is sent) that the router is offering Mobile IPv6 Home Agent services. A value 0 must not be used. The maximum lifetime is 65520 seconds (18.2 hours). This option is ignored, if AdvHomeAgentInfo is not set.

If both HomeAgentLifetime and HomeAgentPreference are set to their default values, Home Agent Information Option will not be sent.

Default: AdvDefaultLifetime

HomeAgentPreference integer

The preference for the Home Agent sending this Router Advertise-

ment. Values greater than 0 indicate more preferable Home Agent, values less than 0 indicate less preferable Home Agent. This option is ignored, if AdvHomeAgentInfo is not set.

If both HomeAgentLifetime and HomeAgentPreference are set to their default values, Home Agent Information Option will not be sent.

Default: 0

AdvIntervalOpt on|off

When set, Advertisement Interval Option (specified by Mobile IPv6) is included in Router Advertisements. When set, minimum limits specified by Mobile IPv6 are used for MinRtrAdvInterval and MaxRtrAdvInterval.

Default: off

PREFIX SPECIFIC OPTIONS

AdvOnLink on|off

When set, indicates that this prefix can be used for on-link determination. When not set the advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some of the addresses belonging to the prefix being on-link and others being off-link.

Default: on

AdvAutonomous on|off

When set, indicates that this prefix can be used for autonomous address configuration as specified in RFC 2462.

Default: on

AdvRouterAddr on|off

When set, indicates that the address of interface is sent instead of network prefix, as is required by Mobile IPv6. When set, minimum limits specified by Mobile IPv6 are used for MinRtrAdvInterval and MaxRtrAdvInterval.

Default: off

AdvValidLifetime seconds|infinity

The length of time in seconds (relative to the time the packet is sent) that the prefix is valid for the purpose of on-link determination. The symbolic value infinity represents infinity (i.e. a value of all one bits (0xffffffff)). The valid lifetime is also used by RFC 2462.

Default: infinity

AdvPreferredLifetime seconds|infinity

The length of time in seconds (relative to the time the packet is sent) that addresses generated from the prefix via stateless address autoconfiguration remain preferred. The symbolic value infinity represents infinity (i.e. a value of all one bits (0xffffffff)). See RFC 2462.

Default: 604800 seconds (7 days)

Base6to4Interface name

If this option is specified, this prefix will be combined with the IPv4 address of interface name to produce a valid 6to4 prefix. The first 16 bits of this prefix will be replaced by 2002 and the next 32 bits of this prefix will be replaced by the IPv4 address assigned to interface name at configuration time. The remaining 80 bits of the prefix (including the SLA ID) will be advertised as specified in the configuration file. See the next section for an example.

If interface name is not available at configuration time, a warning will be written to the log and this prefix will be disabled until radvd is reconfigured.

This option enables systems with dynamic IPv4 addresses to update their advertised 6to4 prefixes simply by restarting radvd or sending a SIGHUP signal to cause radvd to reconfigure itself.

Note that 6to4 prefixes derived from dynamically-assigned IPv4 addresses should be advertised with a significantly shorter lifetime (see the AdvValidLifetime and AdvPreferredLifetime options).

For more information on 6to4, see RFC 3056.

Default: 6to4 is not used

EXAMPLES

```
interface eth0
{
    AdvSendAdvert on;
    prefix fec0:0:0:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

It says that router advertisement daemon should advertise (AdvSendAdvert on;) the prefix fec0:0:0:1:: which has a length of 64 on the interface eth0. Also the prefix should be marked as autonomous (AdvAutonomous on;) and as on-link (AdvOnLink on;). All the other options are left on their default values.

To support movement detection of Mobile IPv6 Mobile Nodes, the address of interface should be used instead of network prefix:

```
interface eth0
{
    AdvSendAdvert on;
    prefix fec0:0:0:1::4/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

For 6to4 support, include the Base6to4Interface option in each prefix section. When using a dynamic IPv4 address, set small prefix lifetimes to prevent hosts from retaining unreachable prefixes after a new IPv4 address has been assigned.

```
interface eth0
{
    AdvSendAdvert on;

    # Advertise at least every 30 seconds
```

```

MaxRtrAdvInterval 30;

    prefix 0:0:0:5678::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        Base6to4Interface ppp0;

        # Very short lifetimes for dynamic addresses
        AdvValidLifetime 300;
        AdvPreferredLifetime 120;
    };
};

```

Since 6to4 is enabled, the prefix will be advertised as 2002:WWXX:YYZZ:5678::/64, where WW.XX.YY.ZZ is the IPv4 address of ppp0 at configuration time. (IPv6 addresses are written in hexadecimal whereas IPv4 addresses are written in decimal, so the IPv4 address WW.XX.YY.ZZ in the 6to4 prefix will be represented in hex.)

FILES

```

/usr/sbin/radvd
/etc/radvd.conf
/var/run/radvd/radvd.pid
/var/log/radvd.log

```

CREDIT

The description of the different flags and variables is in large parts taken from RFC 2461.

RFCS

Narten, T., E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998

Thomson, S., and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

Deering, S., and R. Hinden, Editors, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

Conta, A., and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)", RFC 2463, December 1998.

Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

Carpenter B., Moore K., "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001. (6to4 specification)

SEE ALSO

```

radvd(8), radvdump(8)

```

radvd 0.7.2
(END)

29 Mar 2001

RADVD.CONF(5)