

Fausto José Canhoto de Paiva Vieira

IP com suporte de mobilidade, em ambiente Linux

Relatório Final do Projecto

Mobile
IP

Orientador:

Prof. José Ruela

Porto, Julho de 2001

Resumo

Este documento é o relatório final de um estágio curricular integrado na *Licenciatura em Engenharia Electrotécnica e de Computadores*. Este estágio consiste num trabalho de investigação sobre mobilidade IP, realizado no INESC PORTO – *Instituto de Engenharia de Sistemas e Computadores do Porto*, no segundo semestre do ano lectivo de 2000/01.

Este relatório começa por enquadrar o trabalho nas redes de dados e em especial em sistemas de redes móveis. Depois, são introduzidos pré-requisitos conceptuais necessários à compreensão dos vários assuntos tratados neste documento.

De seguida, são apresentados os objectivos deste trabalho, através dos quais se concretizou o estudo de soluções de mobilidade do protocolo IP na versão 4 e na versão 6, utilizando um ambiente de desenvolvimento baseado em máquinas com o sistema operativo Linux.

Posteriormente, são descritos os métodos de trabalho e as etapas do projecto. Estas estão organizadas de uma forma cronológica.

Uma vez que este projecto tem uma vertente teórica e uma vertente prática, são apresentados os resultados de ambas, tendo sido a investigação teórica o ponto de partida para desenvolver o trabalho prático que se lhe seguiu. É nesta perspectiva que se pode encontrar uma descrição detalhada de todo o processo de instalação e configuração, assim como a descrição dos testes realizados, com vista a permitir um desenvolvimento futuro dos temas abordados neste trabalho.

Finalmente são apresentadas as conclusões sobre todo o trabalho desenvolvido durante o estágio.

Índice

Resumo	i
Índice	ii
Índice de figuras	iii
Índice de tabelas	iii
1. Introdução	1
2. Objectivos	2
3. Métodos de trabalho adoptados	2
4. Etapas do projecto	3
5. Resultados	4
5.1. Estudo teórico	4
5.1.1. <i>Mobile IPv4</i>	4
5.1.2. <i>Mobile IPv6</i>	6
5.1.3. <i>Cellular IP</i>	8
5.2. Trabalho prático	10
5.2.1. <i>Mobile IPv4</i>	10
5.2.2. <i>Cellular IPv4</i>	20
5.2.3. <i>Mobile IPv6</i>	25
5.2.4. <i>Compilar o kernel</i>	34
5.2.5. <i>Instalar as placas wireless Cisco Aironet 340</i>	35
5.2.6. <i>Configurar a rede em IPv6</i>	36
5. Conclusões	37
6. Legado resultante do trabalho	38
Glossário	39
Bibliografia	40
ANEXO A: Resultados da utilização do Cellular IPv4	a

Índice de figuras

Figura 1: Esquema da estrutura do Mobile IPv4	4
Figura 2: Esquema da estrutura do Mobile IPv6	6
Figura 3: Esquema da estrutura do Cellular IP	8
Figura 4: Esquema de montagem do Mobile IPv4	10
Figura 5: Esquema hierárquico de FAs no Mobile IPv4	14
Figura 6: Esquema de montagem incorrecta do Mobile IPv4	16
Figura 7: Esquema de montagem do Cellular IPv4	20
Figura 8: Esquema de montagem do Mobile IPv6	25

Índice de tabelas

Tabela 1: Opções do <i>kernel</i> na configuração do Mobile IPv4	11
Tabela 2: Configuração dos endereços na montagem do Mobile IPv4	13
Tabela 3: Teste de conectividade do Mobile IPv4	17
Tabela 4: Teste da performance do Mobile IPv4	18
Tabela 5: Teste do atraso do Mobile IPv4	19
Tabela 6: Configuração dos endereços na montagem do Cellular IPv4	21
Tabela 7: Opções do <i>kernel</i> na configuração do Mobile IPv6	26
Tabela 8: Outras opções do <i>kernel</i> na configuração do Mobile IPv6	27
Tabela 9: Configuração dos endereços na montagem do Mobile IPv6	28
Tabela 10: Teste de conectividade do Mobile IPv6	32
Tabela 11: Teste do atraso do Mobile IPv6	33
Tabela 12: Opções do <i>kernel</i> para o suporte PCMCIA	35

1. Introdução

A Internet surgiu da necessidade de ligar computadores para permitir partilhar informação e aplicações de software. Com o advento da *World Wide Web*, organizações institucionais e privadas começam a marcar presença na Internet. Mais tarde, esta passou a estar disponível a empresas e outras organizações. Algumas destas passaram a comercializar o acesso da Internet ao público em geral. A evolução deste meio fez surgir novas formas de comunicação interpessoal e de acesso à informação.

Por outro lado, as redes de dados têm sofrido uma grande evolução, impulsionadas em grande parte pela Internet. Actualmente, as redes LAN¹ e as WAN² utilizam tecnologias comuns de forma a conseguir oferecer a velocidade de uma rede local e a qualidade de serviço de uma WAN.

No entanto, surgem novas necessidades em termos de mobilidade para as quais ainda não existe uma resposta adequada. A utilização alargada dos telemóveis, motivada pelo baixo custo da rede GSM³ e novos comportamentos sociais, faz surgir um mercado para comunicações de dados em dispositivos móveis. As redes actuais para comunicações móveis ainda se baseiam no estabelecimento de circuitos para a transmissão de dados. Por sua vez, os protocolos de suporte da Internet utilizam pacotes para transporte de informação e estes caracterizam-se por utilizar bastante largura de banda em curtos intervalos de tempo. Como uma rede de circuitos oferece um canal de largura de banda fixa, a comunicação de pacotes sobre circuitos caracteriza-se por ser um serviço de baixa velocidade e com custo elevado.

Actualmente, já está em funcionamento, em alguns países, um serviço de comunicação móvel por pacotes – GPRS (*General Packet Radio Service*) – que se interliga com redes X.25 ou IP. Estas últimas são especialmente importantes porque a Internet é uma rede IP, o que implica conseguir expandir a Internet à rede GSM e a todos os telemóveis que suportem o GPRS. Apesar disso, está em desenvolvimento outro sistema – UMTS (*Universal Mobile Telecommunications System*) – que promete mais largura de banda, qualidade de serviço e funcionalidades multimédia.

Ao nível das redes de empresa, existe também uma crescente necessidade do suporte de mobilidade para redes privadas sem fios – WLAN (*Wireless Local Area Network*). Neste tipo de redes são necessárias as extensões ao protocolo IP, ao contrário do caso das redes móveis onde aquelas são opcionais.

Também, há que ter em conta que o protocolo IP tem duas versões: a versão 4, que é caracterizada por ter endereços de 32 bits; e a versão 6, também conhecida por IPng – *next generation* – que dispõe de endereços de 128 bits. Este aumento do número de bits pretende resolver a escassez de endereços IP. A criação de uma nova versão permitiu introduzir funcionalidades para responder a necessidades actuais. Destacam-se aspectos tais como a qualidade de serviço, necessária em aplicações em tempo-real, a autoconfiguração passiva ou activa, a segurança, e claro, a mobilidade, que é o assunto aqui tratado pela sua relação com o UMTS. No entanto, todos os outros aspectos são importantes em utilização de dispositivos móveis com suporte multimédia e videoconferência, garantindo privacidade e transacções seguras. Várias empresas e

¹ Rede de área local, normalmente constituída por um conjunto de estações de trabalho interligadas.

² Rede de área alargada, abrange áreas geográficas extensas, podendo interligar LANs. Exemplos de WANs são as redes telefónicas digitais ou a Internet.

³ Sistema Global para comunicações móveis, utilizado em quase todo o mundo com excepção dos EUA.

fabricantes⁴ têm estudado o uso do IPv6 em GPRS, UMTS e na mobilidade em geral e, apesar de terem perspectivas diferentes sobre o tema, estão a desenvolver suporte para os seus produtos e sistemas futuros.

O Mobile IP é um sistema que permite que um computador se desloque da sua rede de origem para outra, conseguindo retomar a conectividade e mantendo o seu endereço de rede. Tal como o IP, o Mobile IP existe na versão 4 e na versão 6.

2. Objectivos

Para a elaboração deste trabalho foram estabelecidos cinco objectivos gerais, sobre os quais se define com maior especificidade os objectivos concretos desta investigação, de acordo com a evolução do mesmo. Existem objectivos ao nível do trabalho prático, sendo estes mais imediatos uma vez que se prendem com o estudo de software já desenvolvido e outros de índole mais teórica, visto que se pretende tirar conclusões sobre sistemas em desenvolvimento.

Um primeiro objectivo é o estudo de vários aspectos relacionados com a comunicação móvel. Pode-se salientar, entre estes, as várias propostas em desenvolvimento no IETF⁵ (*Internet Engineering Task Force*) para o suporte de mobilidade e redes celulares IP, assim como a integração do IPv6 no núcleo das redes móveis de 3ª geração e propostas de migração do IPv4 para o IPv6, ou ainda a análise de várias soluções para o suporte da camada IP como o 802.11⁶ ou o GPRS.

Um segundo objectivo é a instalação e configuração de pacotes de software que implementem o IP móvel – MIPv4 (*Mobile IPv4*) – tal como está especificado pelo IETF e a análise do IP celular.

Um terceiro objectivo é a instalação e configuração de um domínio IPv6 para se poder utilizar pacotes de software que implementem o MIPv6 (*Mobile IPv6*) e verificar as melhorias no suporte da mobilidade, para além de um contacto importante com esta nova versão do IP.

Um objectivo para todo o trabalho prático é criar documentação necessária para que outros possam continuar o trabalho desenvolvido.

Um último objectivo é a análise do trabalho prático e juntamente com o estudo teórico, tirar conclusões sobre a mobilidade IP e a sua capacidade de expansão para além das redes IP, nomeadamente nas redes de comunicações móveis.

3. Métodos de trabalho adoptados

Variar a profundidade de estudo de cada assunto relacionado com o projecto, tendo em conta a sua importância para o mesmo.

Utilizar um *logbook* para a realização dos trabalhos práticos, de modo que estes sejam reproduzíveis mais tarde.

Criar um conjunto de objectivos específicos para cada trabalho prático e ordená-los, de forma a que seja possível avançar com outros trabalhos, retendo as informações essenciais para o projecto.

⁴ Nokia, Alcatel, Nortel, Cisco, Ericsson, Compaq, Siemens, Microsoft, Sun e NEC

⁵ Órgão que estabelece os *standards* para os protocolos utilizados na Internet

⁶ Protocolo do IEEE que especifica uma WLAN para redes IP

4. Etapas do projecto

Para atingir os objectivos propostos foram concretizadas as seguintes etapas:

Construção de um *site* para o projecto que seja dinâmico e permita servir de ponto de partida para o estudo de assuntos relacionados com o projecto. Este *site* deve integrar os vários recursos disponíveis na Internet sobre o tema do trabalho. Pode ser também utilizado para publicar notícias relevantes ao projecto.

Estudo de documentação variada sobre mobilidade IP, IP celular, comunicações móveis, GPRS, IPv6, WLANs e outros assuntos relacionados.

Instalação e configuração de pacotes de software que implementam MIPv4.

Teste e validação de pacotes de software que implementam MIPv4.

Avaliação contínua do desenvolvimento da mobilidade IP por parte do grupo de trabalho responsável por este assunto no IETF.

Análise do único pacote de software que implementa o IPv4 celular.

Instalação e configuração de um domínio IPv6. Instalação e configuração de pacotes de software que implementam MIPv6.

Estudo comparativo das várias soluções ao nível de funcionalidades e desempenho.

5. Resultados

5.1. Estudo teórico

As várias tecnologias de suporte de mobilidade em redes IP que estão a ser desenvolvidas no IETF, podem parecer que estão pouco relacionadas com as redes móveis de dados, nomeadamente com o UMTS. Na realidade, os participantes desses grupos de trabalho são, na sua grande maioria, investigadores dos centros I&D⁷ de grandes empresas de redes e telecomunicações.

De seguida, são descritas sumariamente as várias tecnologias, de modo a enquadrar o trabalho prático.

5.1.1. Mobile IPv4

A arquitectura do Mobile IPv4 [1] é constituída por 3 elementos:

- Mobile Node (MN) - Estação cliente com suporte de mobilidade
- Home agent (HA)- Elemento localizado na rede de origem do *Mobile Node* que captura o tráfego destinado ao MN e o encaminha para este através do encapsulamento de pacotes.
- Foreign agent (FA) - Elemento localizado na rede visitada pelo *Mobile Node* que desencapsula o tráfego e insere-o na sua rede, tendo como destino o MN.

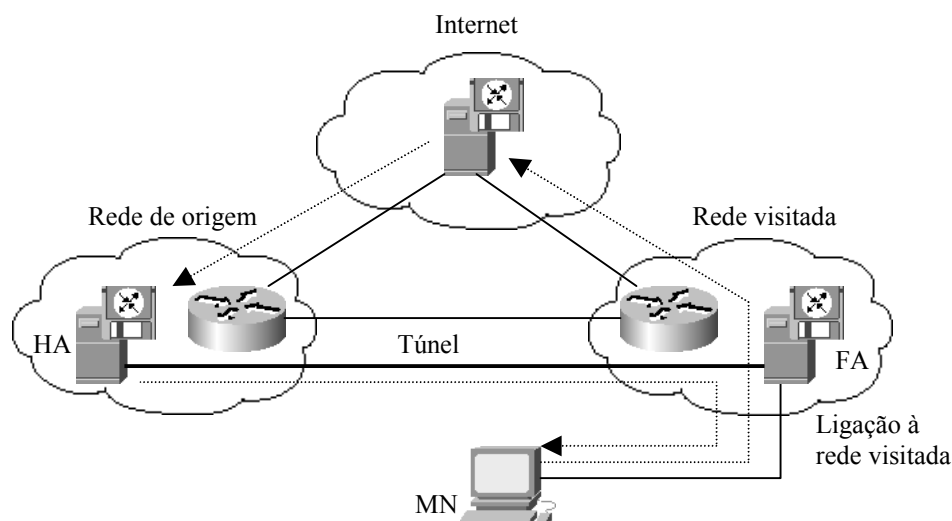


Figura 1: Esquema da estrutura do Mobile IPv4

O *Mobile Node* tem dois endereços IP:

- IP original - este mantém-se, apesar de não pertencer à rede visitada. É utilizado como endereço de origem nos pacotes enviados pelo MN.
- Care-of Address - este endereço é atribuído ao MN na rede visitada para permitir a comunicação nesta rede. O *Foreign Agent* ao desencapsular os pacotes destinados ao MN, utiliza este endereço como endereço de destino desses pacotes.

⁷ Investigação e Desenvolvimento.

A comunicação para o MN é feita de forma transparente: os pacotes enviados para o MN chegam ao *Home agent* e são reenviados por um túnel para o MN; os pacotes enviados pelo MN têm o endereço habitual, o que não revela que este se encontra noutra rede. O túnel termina no *Foreign Agent*, que envia os pacotes para o MN, excepto no caso deste assumir também funções de *Foreign Agent*. Para que tal seja possível, é necessário que a rede visitada utilize DHCP⁸ para atribuir o *Care-of Address*.

Os pacotes enviados pelo MN podem seguir directamente para o seu destino, passando pelo FA ou podem ser enviados pelo túnel bidireccional que está estabelecido entre o HA e o FA [2]. Esta segunda hipótese foi incluída posteriormente para suportar redes visitadas que contêm *routers* que utilizam o endereço de origem para prevenir ataques do tipo “IP spoofing” [3].

O HA e os FAs anunciam a sua presença com mensagens “Agent Advertisement” e o MN pode requerer estas mensagens como resposta a mensagens do tipo “Agent Solicitation”. O MN recebe as mensagens AA e determina se está na sua rede ou numa rede visitada. Quando o MN detecta que está na sua rede, este opera sem serviços de mobilidade. Ao regressar à sua rede, depois de ter estado registado numa rede visitada, este regista-se no HA com o par de mensagens “Registration Request” e “Registration Reply”.

Quando o MN detecta que se moveu para uma rede visitada, obtém o *Care-of Address* a partir de mensagens do FA ou por atribuição de um mecanismo externo como o DHCP. Nesta segunda hipótese, o MN obtém um *Co-located Care-of Address*, que é utilizado para designar quando o MN implementa funções de FA. O MN pode agora registar o *Care-of Address* no HA utilizando o par de mensagens “Registration Request” e “Registration Reply”, possivelmente via o *Foreign Agent*.

O acesso à rede por uma máquina que não pertence àquela, levanta algumas questões de segurança e por isso as mensagens entre os vários agentes têm de ser todas autenticadas [1] [4] [5] [6].

O problema do tempo que demora o MN a trocar de rede (*Hand-off*) é também uma grande preocupação, o que faz com que tenham surgido evoluções do Mobile IPv4 que têm esta questão em consideração [7] [8].

Outras evoluções do Mobile IP surgiram para implementar a optimização das rotas, de modo a que não seja necessário que todos os pacotes destinados ao MN tenham de passar pelo HA [9].

Para a gestão de objectos do Mobile IPv4, pode-se encontrar definições para sistemas de gestão de redes que utilizam o SMIPv2 [10].

⁸ Dynamic Host Configuration Protocol – permite a configuração automática de uma máquina que se ligue à rede.

5.1.2. Mobile IPv6

Para compreender o Mobile IPv6 [11] [12] a melhor maneira é estabelecer uma comparação com o Mobile IPv4.

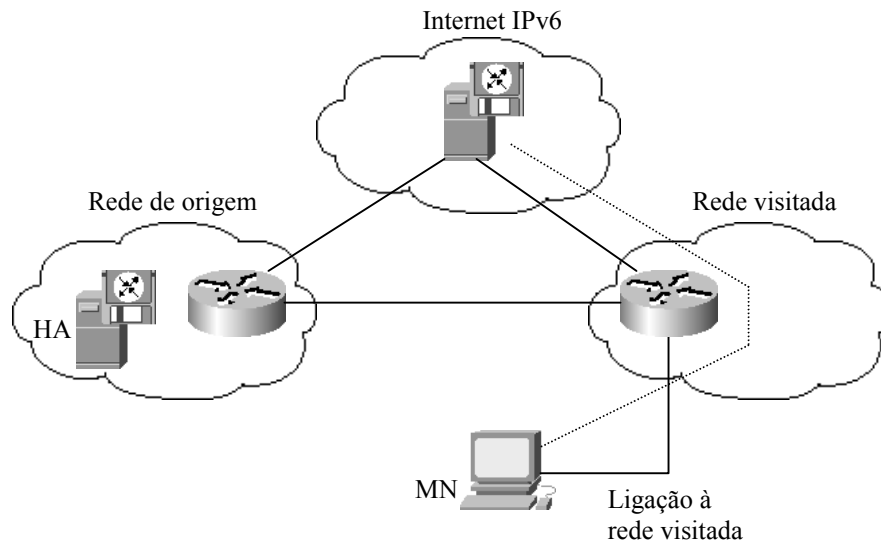


Figura 2: Esquema da estrutura do Mobile IPv6

No Mobile IPv6 não existe a entidade *Foreign Agent*. Por outro lado, passa a desempenhar um aspecto importante o *Correspondent Node*, que é qualquer máquina que comunica com o MN.

O Mobile IPv6 utiliza extensões incluídas no IPv6 para implementar o que está designado no Mobile IPv4 como optimização das rotas, ou seja, o armazenamento temporário do *Care-of Address* por parte do *Correspondent Node*, permitindo que o tráfego entre este e o MN seja feito por rotas directas.

O MN utiliza o *Care-of Address* como endereço de origem, resolvendo assim o problema dos *routers* que verificavam o endereço de origem (“*ingress filtering*”). O endereço de origem é transportado num campo opcional do IPv6. Esta característica permite que os pacotes de *multicast* tenham como origem o *Care-of Address*, que simplifica o processo das rotas. No Mobile IPv4 os pacotes de *multicast* necessitam de ser transportados por um túnel até ao HA.

A inexistência de FAs advém do facto do IPv6 permitir descobrir em que rede visitada o MN se encontra a partir do “*Neighbor Discovery*” e auto-configurar os interfaces com a utilização do “*Address Autoconfiguration*”.

O Mobile IPv6 utiliza as funções nativas do IPv6 (IPSec) para implementar o sistema de segurança e autenticação.

Os mecanismos de detecção de movimento no Mobile IPv6 permitem uma confirmação bidireccional da capacidade de comunicação do MN com o *router* da rede. Esta característica é especialmente importante em situação de ligações sem fios, em que a capacidade de comunicação é muitas vezes assimétrica.

O HA utiliza o IPv6 *Neighbor Discovery* em vez do ARP do Mobile IPv4, para interceptar qualquer pacote que tenha como destino o MN, quando este está numa rede visitada.

Existem desenvolvimentos do Mobile IPv6 em várias áreas como:

- Troca rápida de rede, aplicável em redes sem fios [13] [14].
- *Hand-off* entre redes que pertencem à mesma estrutura hierárquica[15].
- Mobilidade IPv4 e IPv6 para redes interligadas que utilizam *dual stack* [16].
- Utilização do Mobile IPv6 em rede UMTS [23] [24] [25].

5.1.3. Cellular IP

A arquitectura do Cellular IP [17] [22] é constituída por três elementos:

- Mobile Host (MH) – Dispositivo móvel que suporta *Cellular IP*.
- Base Station (BS) – Encaminha pacotes dentro da rede celular e comunica com os *Mobile Hosts*.
- Gateway – interliga a rede celular com rede exterior.

O Cellular IP surge da necessidade de suportar eficientemente a micro-mobilidade. Este problema surge quando um *Mobile Host* consegue comunicar com duas *Base Stations* ao mesmo tempo. O MH iria perder pacotes e ficar momentaneamente sem ligação se deixasse de comunicar com uma BS para comunicar com a outra. Com o Cellular IP existe um período em que o MH comunica com as duas *Base Stations* ao mesmo tempo.

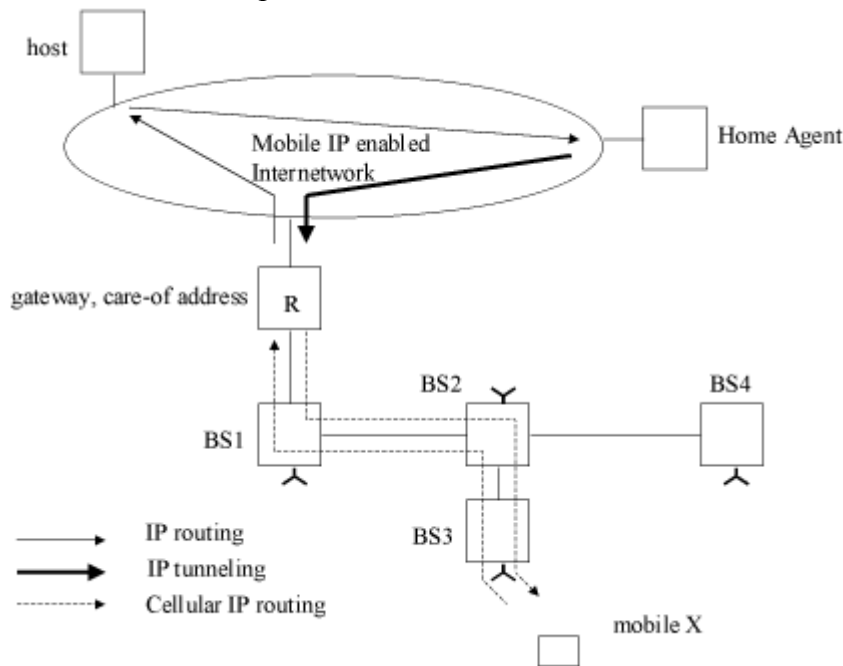


Figura 3: Esquema da estrutura do Cellular IP

O Cellular IP é constituído por uma estrutura hierárquica de *Base Stations* de modo a reduzir os procedimentos de *Hand-off*. A BS1 da Figura 3 não se apercebe se o *Mobile Host* comunica com a BS2, BS3 ou BS4. Tudo o que o BS1 necessita é de saber que a rota para *Mobile Host* é a partir da BS2.

O *Routing* é feito a partir de rotas guardadas em *cache*, tendo estas de ser actualizadas antes que a sua validade expire. O mapeamento das rotas pode ser feito por 2 métodos:

- A partir de pacotes de dados com origem no *Mobile Host* que viajam até à *Gateway*.
- A partir de pacotes específicos de actualização de rotas. Estes pacotes não contêm dados.

O *Hand Off* acontece quando o *Mobile Host* passa a receber sinais mais fortes de uma *Base Station* adjacente e sintoniza para a nova estação para transmitir um pacote de

atualização de rotas. Assim passam a existir 2 rotas para o mesmo *Mobile Host*. Não é necessário remover as rotas da *Base Station* antiga, uma vez que estas expiram ao fim do tempo de validade. Os pacotes para o *Mobile Host* serão transmitidos para ambas as rotas, enquanto estas existirem.

O *Paging* é um sistema para manter um caminho para dispositivos móveis (*Mobile Host*) quando estes se encontram em modo de baixo consumo de energia. É um sistema semelhante ao *routing* com tempos de validade maiores para as rotas de *paging* e com áreas de *paging* que agregam várias *Base Stations*, permitindo que se reduza o número de *Handoffs*.

O Cellular IP pode ser interligado com Mobile IPv4. Para o Mobile IPv6 foi criado o Cellular IPv6 [18], que utiliza as vantagens do IPv6 no seu funcionamento.

Outros desenvolvimentos no Cellular IP têm abordado os requisitos de aspectos específicos do seu funcionamento [19] [20], assim como técnicas para minimizar o impacto da mobilidade na continuidade de serviços estabelecidos previamente [21].

Certas extensões ao Mobile IPv4 e ao Mobile IPv6 utilizam conceitos do Cellular IP para melhorar o seu funcionamento [7] [8] [13] [14] [15].

5.2. Trabalho prático

5.2.1. Mobile IPv4

Descrição da instalação

A primeira questão a considerar é o hardware. São necessários 3 computadores, 2 *hubs* (concentradores) e um *router* (encaminhador).

Os computadores são designados de HA, FA e MN, devido à função que desempenham.

Os *hubs* podem ser substituídos por outro equipamento que desempenhe a mesma função. Pode-se utilizar um *switch* (comutador) em vez de um *hub* ou ainda um *switch* que permita criar VLANs⁹ por portas. Não se pode utilizar VLANs por endereço MAC visto que, ao mudar de porto no *switch*, a máquina ficaria na mesma VLAN.

O *router* pode ser substituído por uma máquina Linux com 2 interfaces de rede. Neste trabalho foi utilizada esta última opção.

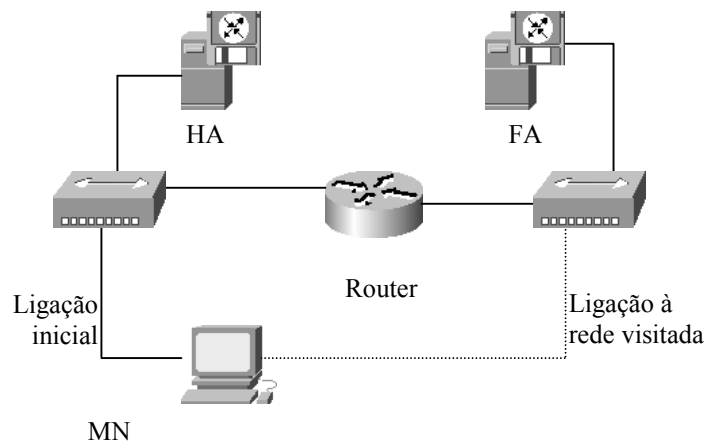


Figura 4: Esquema de montagem do Mobile IPv4

Na **Figura 4**, a ligação inicial indica que o MN está na sua rede (*home network*). Ao mudar para outra rede (*visited network*), substitui-se o cabo da ligação inicial pelo da rede visitada.

Estas máquinas devem correr o Linux Red Hat 6.0 ou superior e uma versão do *kernel* 2.2.x ou superior. Neste trabalho foi utilizado a versão 7.0 do Red Hat com a versão 2.4.2 do *kernel*.

Em relação ao *kernel*, existe um conjunto de opções de rede que têm de estar activadas para o sistema funcionar:

⁹ *Virtual LANs* – permitem separar a rede lógica da rede física e, como consequência, máquinas que pertençam a VLANs diferentes não comunicam entre si na camada de ligação, mesmo estando ligadas ao mesmo *switch*.

Opção do <i>kernel</i>	Nome do parâmetro	Uso	Opcional
Packet socket	CONFIG_PACKET	Todos	✗
Kernel/User netlink socket	CONFIG_NETLINK	Todos	✗
Routing messages	CONFIG_RTNETLINK	Todos	✗
IP: Socket Filtering	CONFIG_FILTER	MN	✓
IP: tunneling	CONFIG_NET_IPIP	Todos	✗
IP: advanced router	CONFIG_IP_ADVANCED_ROUTER	FA	✗
IP: policy routing	CONFIG_IP_MULTIPLE_TABLES	FA	✗

Tabela 1: Opções do *kernel* na configuração do Mobile IPv4

Para além das opções referidas na **Tabela 1**, deve-se usar o conjunto de opções do *kernel* que estão definidas por omissão. Estas opções devem suportar as placas de rede que vão ser utilizadas assim como todas as opções necessárias para o funcionamento normal da rede. Deve-se confirmar que o “IP Multicasting” está activado. Se for possível utilizar um único *kernel* para todas as máquinas, então deve-se activar todas as opções da **Tabela 1**. A opção “IP: Socket Filtering” é opcional embora recomendada pelos autores.

Para configurar e compilar o *kernel* pode-se consultar a secção **5.2.4. Compilar o *kernel***.

Depois de instalar em todas as máquinas o novo *kernel*, pode-se então preparar a instalação do software que implementa o Mobile IPv4. A versão testada foi o “Dynamics 0.6.2 – HUT Mobile IP”, que se pode obter em:

<http://www.cs.hut.fi/Research/Dynamics/software.html>

Estão disponíveis versões pré-compiladas ou em código fonte.

Ao escolher uma versão pré-compilada, só é necessário executar o seguinte:

- Copiar o pacote específico para cada máquina.
- Instalar a seguinte biblioteca matemática: GNU MP (gmp2 – normalmente existe em RPM).
- Executar o comando para instalar o pacote. Em Red Hat os RPMs são instalados com “rpm -i dynamics-xx-0.6.x-1.i386.rpm”.

Ao escolher a versão em código fonte, é necessário fazer o seguinte:

- Instalar a seguinte biblioteca matemática: GNU MP (gmp2 – normalmente existe em RPM).
- Instalar a versão de desenvolvimento dessa biblioteca (gmp2-devel – existe num RPM separado).
- Descomprimir o ficheiro a partir do comando “tar -zxvf dynamics-0.6.x.tar.gz”.
- Executar os seguintes comandos dentro da directoria “dynamics-0.6.x”:
 - “./configure”.

- “make”.
- “make install”.
- Repetir tudo nas 3 máquinas. Uma nova compilação não é necessária se for feito o seguinte:
 - Comprimir de novo a directoria “dynamics-0.6.x” com o comando:


```
“tar -zcvvf dynamics-0.6.x-compiled.tar.gz dynamics-0.6.x”
```
 - Copiar o ficheiro para as outras máquinas.
 - Voltar a descomprimir:

```
“tar -zxvf dynamics-0.6.x-compiled.tar.gz”
```

.
 - Executar dentro da directoria o comando “make install”.

A instalação necessita de ser feita nas 3 máquinas, no entanto só é necessário que existam os seguintes ficheiros em cada uma delas:

Mobile Node

```
/usr/sbin ou /usr/local/sbin
• dynmnd (daemon)
• dynmn_tool
• dynamics_mnsetup
/etc ou /usr/local/etc
• dynmnd.conf
```

Home Agent

```
/usr/sbin ou /usr/local/sbin
• dynhad (daemon)
• dynha_tool
• dynamics_hasetup
/etc ou /usr/local/etc
• dynhad.conf
```

Foreign Agent

```
/usr/sbin ou /usr/local/sbin
• dynfad (daemon)
• dynfa_tool
• dynamics_fasetup
• rsakeygen
• rsakeyhash
/etc ou /usr/local/etc
• dynfad.conf
• dynfad.key
```

Os vários ficheiros indicados como *daemon*¹⁰ são os executáveis que implementam o Mobile IPv4 enquanto que os ficheiros “.conf” são os ficheiros de configuração. As directorias de instalação podem variar, podendo ser especificado o seu destino durante o processo de compilação. Para mudar o destino destes ficheiros, pode-se consultar o ficheiro “INSTALL”, localizado na directoria “dynamics-0.6.x”

Depois de instalar o software, é necessário preparar a rede para o seu funcionamento. Para isso, deve-se atribuir os endereços aos interfaces das várias máquinas, assim como criar as rotas necessárias para as máquinas comunicarem entre si.

¹⁰ Programa que fica a correr em memória e que implementa um determinado serviço. Outros exemplos de *daemons* são o ftpd, lpd, telnetd, que implementam os serviços de ftp, impressão e telnetd respectivamente.

Para atribuir os endereços de rede a um determinado interface (`eth0`) utilizaram-se os seguintes comandos:

```
(HA) ifconfig eth0 192.168.242.1
(FA) ifconfig eth0 192.168.240.1
(MN) ifconfig eth0 192.168.242.2
```

Para criar as rotas entre as máquinas só é necessário utilizar o router como *default gateway*:

```
(HA) route add default gw 192.168.242.254
(FA) route add default gw 192.168.240.254
(MN) route add default gw 192.168.242.254
```

Para o Mobile IPv4 funcionar é necessário activar o *Multicast* no HA. Este tem de ser activado no interface que está na mesma rede que o MN. Neste caso é em `eth0`. Para activar o *Multicast* basta criar uma rota deste tipo.

```
route add -net 224.0.0.0 netmask 224.0.0.0 dev eth0
```

Os endereços das máquinas são os seguintes:

Máquina	Endereço	Default gateway
HA	192.168.242.1 (eth0)	192.168.242.254
FA	192.168.240.1 (eth0)	192.168.242.254
MN	192.168.242.2 (eth0)	192.168.242.254
Router	192.168.242.254 (eth0)	Não definido
	192.168.240.254 (eth1)	

Tabela 2: Configuração dos endereços na montagem do Mobile IPv4

Um aspecto final a ter em conta é o facto do FA e o HA terem funções de *Routing*, logo é necessário que o *IP forwarding* esteja activo. Para verificar se está activado pode-se executar o seguinte comando “`cat /proc/sys/net/ipv4/ip_forward`”. Se este indicar um “0” então é necessário activá-lo, passando para “1” com “`echo 1 > /proc/sys/net/ipv4/ip_forward`”.

Configuração do Mobile IPv4

Tendo a rede definida, pode-se então configurar o Mobile IPv4. Existe uma ferramenta de configuração para cada máquina e o respectivo ficheiro de configuração.

Configurar o HA

O “`dynamics_hasetup`” permite fazer configurar o HA de uma forma simples. As questões colocadas por esta ferramenta são:

- O endereço de rede do HA: “192.168.242.1”
- O endereço de rede do HA: “192.168.242.2”

Configurar o FA

O “`dynamics_fasetup`” permite configurar o FA. Esta ferramenta pergunta:

- Se o FA pertence a uma estrutura hierárquica ou simples: “s”
- O endereço de rede do FA: “192.168.242.1”

Se o FA estivesse numa estrutura hierárquica e se se respondesse “h” à primeira questão, então apareceria o seguinte esquema:

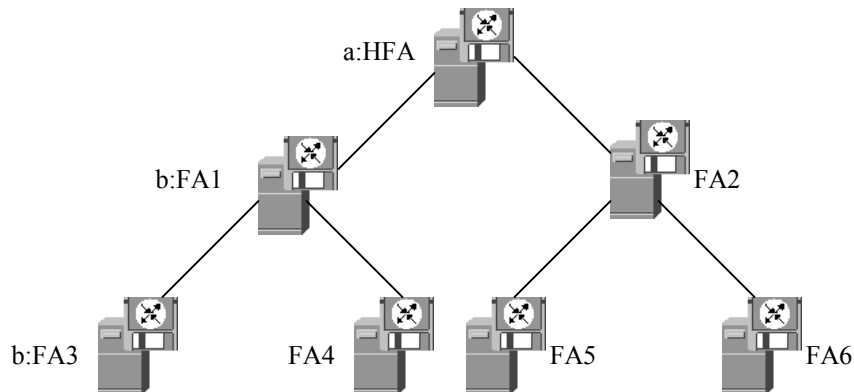


Figura 5: Esquema hierárquico de FAs no Mobile IPv4

- Qual o tipo de agente (a/b/c)
 - Ao responder a, seria pedido apenas o IP do FA.
 - Ao responder b, seria pedido apenas o IP do HFA.
 - Ao responder c, seria pedido o IP do HFA e do FA de nível superior.

Configurar o MN

O MN é configurado pelo “`dynamics-mn-setup`”. As questões são:

- O endereço IP do MN: “192.168.242.2”
- O endereço IP do HA: “192.168.242.1”
- O prefixo de rede da *home network*: “192.168.242.0”
- O comprimento da máscara do prefixo de rede: “24” (equivale a uma máscara 255.255.255.0)

Outras configurações do Mobile IPv4

O Mobile IPv4 tem um conjunto vasto de parâmetros que podem ser modificados para melhorar aspectos do seu funcionamento, da segurança ou para resolver problemas relacionados com configurações de rede diferentes das habituais. Os ficheiros de configuração das várias entidades são:

- O “`dynhad.conf`” para o HA.
- O “`dynfad.conf`” para o FA.
- O “`dynamnd.conf`” para o MN.

Estes ficheiros encontram-se na directoria “`/etc`” ou “`/usr/local/etc`”.

É possível encontrar uma documentação detalhada relativa a estes ficheiros a partir do comando “man”. Ao instalar o Mobile IPv4 numa máquina, este introduz a documentação no sistema de ajuda do Linux. Pode-se aceder a esta utilizando o “man dynamics”.

Utilizar o Mobile IPv4

Para usar o Mobile IPv4, é necessário que as 3 máquinas estejam a correr os *daemons* respectivos. Estes podem ser iniciados a partir da linha de comando, utilizando apenas o nome do executável respectivo:

- “dynhad” no HA.
- “dynfad” no FA.
- “dynmnd” no MN.

Estes executáveis correm em *background*, o que torna a resolução de problemas mais difícil. Logo, pode-se utilizar como parâmetros destes executáveis, duas opções para passarem a correr em *foreground* e em modo *debug*:

```
“(daemon) --fg --debug”
```

O *daemon* do MN tem uma opção que é importante, mas que está mal documentada. Esta opção permite reduzir para 1/3 o tempo de vida dos anúncios dos agentes. Isto é especialmente importante para ligações sem fios. Em toda a documentação está indicado que a opção é “--wireless-policy”, quando na realidade é “--wireless”. Este parâmetro tem 2 valores: “0” está inactivo e “1” está activo. Assim, para correr o *daemon* com esta opção, deve-se executar o seguinte comando (por esta ordem de parâmetros):

```
“dynmnd --fg -debug --wireless 1”
```

O Mobile IPv4 contém uma ferramenta que permite interagir com os *daemons*. No HA (“dynha_tool”) e no FA (“dynfa_tool”), esta ferramenta só permite monitorizar o funcionamento, mas no MN (“dynmn_tool”) é possível enviar comandos para criar ou destruir ligações e ainda mudar as políticas de *Hand-off*. Existe documentação sobre estas ferramentas no sistema de ajuda do Linux.

Problemas encontrados e a sua resolução

O MN não se conseguia registar por demorar tempo excessivo.

► O MN tem de estar relativamente sincronizado em termos de relógio porque o método que protege contra registos falsos no HA, a partir de mensagens de registo geradas pelo MN anteriormente, é baseado na introdução das horas na mensagem de registo, sendo esta comparada com as horas do HA. Só são aceites registos suficientemente próximos. A solução foi sincronizar o relógio do MN e mudar no ficheiro de configuração do HA o valor máximo aceite de 120 para 320.

O MN não se conseguia registar porque o endereço do HA era diferente do endereço da resposta que chegava ao FA.

► O HA não pode comunicar com o FA num interface diferente do interface na rede do MN. Esta característica foi implementada por uma questão de segurança. A montagem não pode ter o HA a funcionar como *router*:

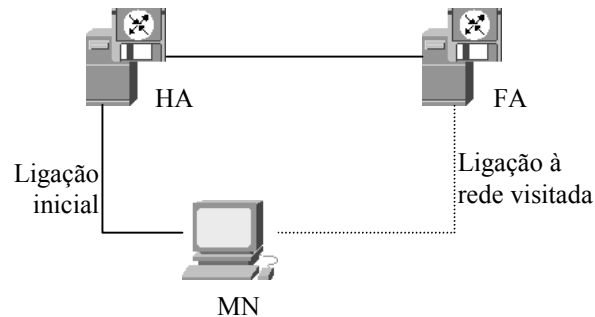


Figura 6: Esquema de montagem incorrecta do Mobile IPv4

Para o Mobile IPv4 funcionar, deve-se implementar uma montagem semelhante à **Figura 4**.

O FA não estava a anunciar no interface correcto.

► Ao contrário do HA que não suporta múltiplos interfaces, o FA está preparado para esta situação. No ficheiro de configuração do FA podem-se indicar os vários interfaces que se utilizam e qual a sua função: comunicar com o HA, com o FA ou ambos.

A documentação estava muito incompleta.

► Este software tem um conjunto variado de documentação que se pode encontrar na directoria “*dynamics-0.6.x/doc*”. Para além desta fonte de informação, pode-se encontrar ajuda específica aos vários ficheiros a partir do comando “*man dynamics*”. Esta está disponível apenas em máquinas em que se tenha feito a instalação completa. Para além destes recursos, pode-se também encontrar ajuda na Internet, a partir de um serviço de *mailing list* e um sistema de arquivo de mensagens antigas. Para aceder a estes recursos, deve-se consultar o *site* deste software.

O MN demorava muito tempo a “regressar a casa”.

► Tal como está descrito nos testes de *Hand-off* que se seguem, o MN não se conseguia registar no HA quando regressava à sua rede. Este problema não acontece se o *daemon* do MN for lançado quando este se encontra numa rede visitada.

Testes e resultados

Tendo em conta a natureza deste software e a estrutura de rede que foi implementada, optou-se pelos seguintes testes:

- **Conectividade** – verificar se é possível comunicar com o MN quando este se encontra numa rede visitada.
- **Hand-off** – testar a capacidade de mudança entre redes, tanto ao nível dos atrasos, como da gestão dos anúncios dos FAs e do HA.

- **Performance** – capacidade máxima de transferência de dados, em comparação com uma ligação normal.
- **Atraso** – valor médio do atraso introduzido pelo uso da mobilidade.

Outros testes que se adaptavam a este tipo de experiência mas que não podem ser realizados com esta montagem são os testes de:

- **Carga** – capacidade do HA suportar vários MNs ligados em redes visitadas. Isto implica um número elevado de MNs. O valor máximo de referência é de 20 ligações simultâneas.
- **Mobilidade wireless** – funcionamento do Mobile IPv4 num sistema que utilize ligações sem fios com sobreposição de áreas de cobertura. Esta versão do Mobile IPv4 só suporta placas WaveLan para a medição da intensidade do sinal.
- **Macro-mobilidade** – funcionamento do sistema quando o MN se desloca para uma rede distante do HA. O conceito de distante implica atrasos e menor largura de banda do que de uma rede local.

Testes de conectividade

Utilizou-se o comando “ping” para estes testes. Depois do MN se ter registado com sucesso no FA, fizeram-se os seguintes testes de *ping*:

Origem	Destino	Resultado
MN	HA (192.168.242.1)	✓
MN	FA (192.168.240.1)	✓
MN	Internet (193.136.28.99)	✓
Outra máquina da rede (192.168.30.66)	MN	✓

Tabela 3: Teste de conectividade do Mobile IPv4

A ligação ao resto da rede e à Internet é feita a partir do *router*, que está ligado a uma *Gateway*¹¹ que corre um serviço de NAT¹².

¹¹ Implementa serviços de nível superior ao encaminhamento. Também conhecido por *application gateway*. Utilizado historicamente como sinónimo de *router*.

¹² *Network Address Translation* – permite transformar os endereços privados em endereços públicos.

Testes de *Hand-off*

Os testes foram realizados apenas entre o HA e FA, uma vez que não existiam outros FAs. Os resultados foram os seguintes:

- Os parâmetros que estão definidos por omissão para a troca de agente fazem com que o MN escolha um novo FA apenas quando o tempo de vida do anúncio sobre o agente anterior expire (120 segundos). Este valor é calculado a partir da soma do valor do intervalo entre anúncios (30 segundos) com o valor do tempo de validade do anúncio (3×30 segundos).
- O tempo total de registo do MN no FA e a criação do túnel entre o HA e o FA é bastante baixo. Com esta configuração de rede, a comunicação entre as várias entidades é muito rápida, o que faz com que o tempo total seja inferior a 1 segundo.
- O processo de regresso a casa, a partir do registo no HA e a terminação dos túneis activos, apresenta alguns problemas: o MN recebe as mensagens de anúncio do HA e responde a estas com um pedido de registo. O HA ignora esse pedido, indicando que o identificador do túnel não corresponde a um túnel activo. O MN repete o pedido sempre que recebe um anúncio do HA, até que a informação sobre o FA expire. A partir deste momento, o MN considera que regressou à sua rede. Este problema não surge quando o MN arranca o *daemon* a partir da rede visitada.

Testes de performance

Para este teste utilizou-se a transferência de ficheiros por FTP¹³. A vantagem da utilização deste método assenta no facto de utilizar a camada TCP e de calcular automaticamente a taxa média de transferência de dados. Utilizaram-se ficheiros com tamanhos superiores a 5 Mb para que os valores obtidos sejam mais imunes a variações.

Tipo de ligação	Taxa média de transferência
MN na sua rede	2 118 Kbps
MN na rede visitada	2 108 Kbps

Tabela 4: Teste da performance do Mobile IPv4

A partir destes valores, pode-se verificar que o MN não é afectado na performance pelo facto de se encontrar numa rede visitada.

¹³ *File Transfer Protocol* – protocolo de transferência de ficheiros assente sobre a camada TCP/IP, que se baseia na utilização de uma ligação de controlo e a criação de uma ligação de dados por cada ficheiro transferido.

Testes de atraso

Os testes foram realizados entre o MN e uma outra máquina da rede. Esta pertencia a uma rede que era acessível a partir do *router*. Utilizou-se o comando “ping” para obter valores para o atraso quando o MN está ligado na sua *Home Network* ou quando está ligado ao FA.

Tipo de ligação	Mínimo	Médio	Máximo
MN na sua rede	875 μ s	922 μ s	988 μ s
MN na rede visitada	823 μ s	933 μ s	1 240 μ s

Tabela 5: Teste do atraso do Mobile IPv4

Verifica-se que a diferença de tempo entre as duas situações é mínima. Este teste foi realizado com uma montagem em que a distância em termos de *hops*¹⁴ era a mesma.

¹⁴ Número de *routers* por onde passa um determinado pacote.

5.2.2. Cellular IPv4

Descrição da instalação

O Cellular IP utiliza uma estrutura em árvore. A montagem mais simples é ter uma máquina a funcionar como *Gateway*, com duas máquinas ligadas a esta a funcionarem como *Base Stations*. A montagem é a seguinte:

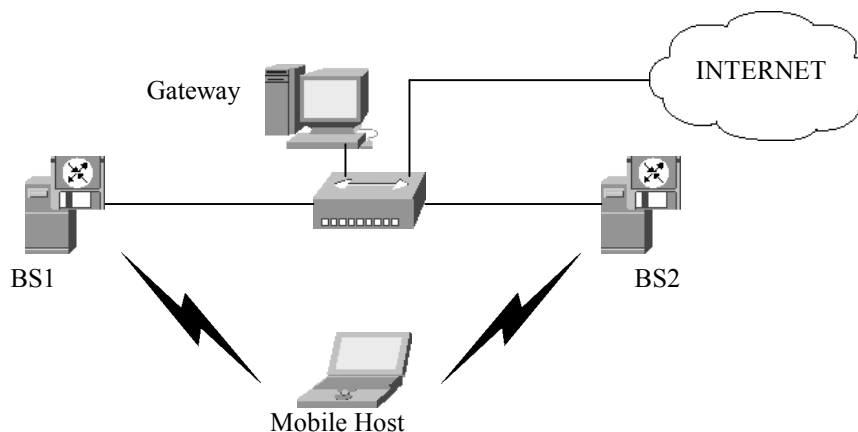


Figura 7: Esquema de montagem do Cellular IPv4

Estas máquinas necessitam de um *kernel* de Linux com a versão 2.2.12 ou 2.2.14. Este software não foi testado em outras versões. Pode-se encontrar informações sobre a instalação de novas versões na secção **5.2.4. Compilar o *kernel***. O Red Hat 6.1 utiliza a versão 2.2.14.

Para além desta versão do *kernel*, é também necessário ter instalado no *Mobile Host* o *Wish 8.0*, que é um sistema de interface gráfico que utiliza a biblioteca `tk1/tk`. O Red Hat instala por omissão este pacote.

É ainda necessário instalar a biblioteca *libpcap*. Pode-se verificar se está instalada, se existir o ficheiro `/usr/include/pcap`. Para obter uma versão em RPM para o Red Hat 6.1 deve-se procurar o ficheiro `libpcap-0.4-19.i386.rpm` num *site* como o <http://rpmfind.net>.

Para obter o pacote de software que implementa o Cellular IPv4, deve-se aceder ao seguinte *site*:

http://comet.columbia.edu/cellularip/linux_src_code.htm

Este *site* contém um conjunto variado de informação, dando especial atenção às placas WaveLan, uma vez que esta implementação do Cellular IPv4 consegue que o *Mobile Host* utilize os valores da intensidade do sinal para determinar qual a *Base Station* com que comunica. As placas Aironet¹⁵ suportam a medição destes valores, mas o software não utiliza esta funcionalidade.

Depois de fazer o *download* do `cip-1.1.tar.gz`, deve-se seguir os seguintes passos:

- Descomprimir o ficheiro com o comando `tar -zxvf cip-1.1.tar.gz`

¹⁵ A empresa Aironet foi comprada pela Cisco, passando as placas mais actuais a serem designadas por Cisco Aironet.

- Dentro da directoria “cip-1.1” existe o ficheiro “Makefile”, que é necessário editar. Este ficheiro vem configurado para as placas WaveLan e é necessário modificá-lo para suportar as placas Aironet. Na linha em que se lê “`{CC} {CFLAG} -c cipmobile.c -DWICACHE -DANCACHE_ -DDEBUG_ -DD_DEBUG_`”, deve-se modificar para “`{CC} {CFLAG} -c cipmobile.c -DANCACHE -DDEBUG_ -DD_DEBUG_`”.
- Executar o comando “make”
- Repetir o processo em todas as máquinas ou copiar a directoria e os seus conteúdos para as outras máquinas.

Para utilizar as placas de rede sem fios deve-se consultar as secção **5.2.5. Instalar as placas wireless Cisco Aironet 340.**

Para configurar o Cellular IPv4 é necessário utilizar os endereços de rede das várias máquinas.

Máquina	Endereço	Default gateway
Gateway	194.117.24.66 (eth0)	194.117.24.254
BS1	194.117.24.167 (eth0)	Não definido
	192.168.100.10 (eth1)↗	
BS2	194.117.24.135 (eth0)	Não definido
	192.168.100.20 (eth1)↗	
Mobile Host	192.168.100.30 (eth0)	Não definido

Tabela 6: Configuração dos endereços na montagem do Cellular IPv4

Configuração do Cellular IPv4

Para configurar as várias entidades existem os ficheiros “cipmobile.conf” para o *Mobile Host* e o “cipnode.conf” para as *Base Stations* e a *Gateway*.

Configurar o Mobile Host

O ficheiro de configuração “cipmobile.conf”, que se encontra na directoria “cip-1.1”, é simples:

```
wireless interface:      eth0
mobile's IP address:    192.168.100.30
air interface name:     mobile-air
route-update-time:     200          %in milliseconds
paging-update-time:    10000       %in milliseconds
active-state-timeout:  2000         %in milliseconds
handoff:                0:         %forced (=0) or SNR based (=1)
```

Neste ficheiro, os parâmetros importantes são:

- “wireless interface” – Este necessita do nome do interface da placa de rede sem fios.
- “mobile's IP address” – Este é o endereço associado ao interface.
- “handoff” – Este permite seleccionar se o *Hand-off* é forçado ou se se baseia na intensidade do sinal. Esta segunda opção não está disponível para placas Aironet.

Configurar a BS1

A configuração do BS1 requer mais parâmetros. É necessário indicar o tipo de interface que se utiliza na comunicação com o *Mobile Host* e a *Gateway*. Isto deve-se ao facto do Cellular IPv4 suportar BS1 apenas com ligações sem fios. O ficheiro “cipnode.conf” preparado para a BS1 é o seguinte:

```
GW:                                NO

IF NO, neighbor, uplink direction: (wire, eth0,
194.117.24.66) %(medium, interface name, dstIP)

leaf neighbours(s):                (wireless, eth1)

paging cache:                       YES

semisoft:                           NO

route-timeout:                      3000    %in milliseconds

paging-timeout:                     30000   %in milliseconds

max number of mobiles in cache:      100

max number of node interfaces:       10

GW IP address:                      194.117.24.66
```

Os parâmetros a ter em conta são:

- “GW” – para uma *Base Station*, este assume o valor de “NO”.
- “IF NO, neighbor, uplink direction” – indica a máquina superior na hierarquia. São necessários três valores que indicam o tipo de interface (wire/wireless), o nome do interface e o IP da máquina de destino.
- “leaf neighbours(s)” – indica em que interface o *Mobile Host* comunica com esta *Base Station*.
- “GW IP address” – indica o IP da *Gateway*.

Configurar a BS2

A configuração da BS2 é igual à da BS1, uma vez que não é necessário indicar o IP da própria máquina.

Configurar a *Gateway*

Na configuração da *Gateway*, é necessário indicar as *Base Stations* que estão ligadas directamente a esta. O ficheiro “*cipnode.conf*” preparado para a *Gateway* é o seguinte:

```
GW:                                YES
IF YES, default router's IP address: (wire, eth0,
194.117.24.254)

leaf neighbours(s):                (wire, eth0,
194.117.24.135), (wire, eth0, 194.117.24.167)

paging cache:                       YES

semisoft:                            YES

If yes then size:                   10    %in packets

route-timeout:                      3000  %in milliseconds

paging-timeout:                     30000 %in milliseconds

max number of mobiles in cache:      100

max number of node interfaces:       10

GW IP address:                      194.117.24.66
```

Os parâmetros a ter em conta são:

- “GW” – para uma *Gateway*, este assume o valor de “YES”.
- “IF YES, default router's IP address” – indica os parâmetros da ligação ao *router* de acesso ao resto da rede.
- “leaf neighbours(s)” – indica quais as *Base Stations* que comunicam directamente com a *Gateway*.
- “GW IP address” – indica o IP da própria máquina.

Utilizar o Cellular IPv4

Antes de usar o Cellular IPv4, deve-se verificar se as placas *wireless* estão a funcionar correctamente e devidamente configuradas. Existem mais informações sobre este assunto em **5.2.5. Instalar as placas *wireless* Cisco Aironet 340.**

Para correr o Cellular IPv4 na *Gateway* e nas *Base Stations*, basta executar a partir da directoria “*cip-1.1*” o comando “*./cipnode*”.

No *Mobile Host* deve-se executar o “*./cip*”.

Problemas encontrados e a sua resolução

O MH não parece estar a funcionar.

► O programa “*./cip*” arranca e cria uma caixa de diálogo que permite forçar o *Hand-off*. Logo a seguir, acontece um “*segmentation fault*” em *background*. Isto faz com que o interface fique a comunicar com um processo terminado, iludindo

o utilizador quanto ao estado do Cellular IPv4 no MH. O “./cip” foi testado em várias máquinas e em vários *kernels* e produziu sempre o mesmo resultado. Não foi possível obter qualquer ajuda do *site* deste software. Em anexo são apresentados os resultados do funcionamento das máquinas, onde se verifica que a *Gateway* e as *Base stations* funcionam correctamente, ao contrário do *Mobile Host*.

5.2.3. Mobile IPv6

Descrição da instalação

O primeiro aspecto a ter em conta é o hardware. São necessários 3 computadores. Dois destes necessitam de duas placas de rede cada.

Os computadores são designados por HA, CN e MN, devido à função que desempenham.

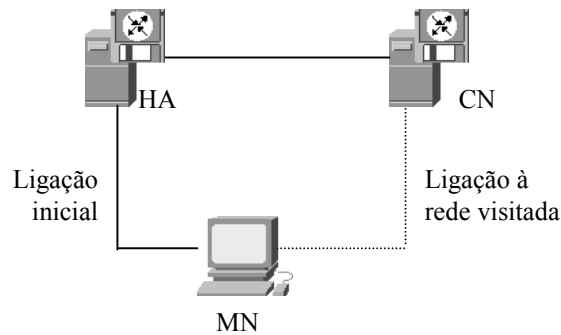


Figura 8: Esquema de montagem do Mobile IPv6

Na **Figura 8**, a ligação inicial indica que o MN está na sua rede (*home network*). Ao mudar para outra rede (*visited network*), substitui-se o cabo da ligação inicial pelo da rede visitada.

Estas máquinas devem correr o Linux Red Hat 7.0 ou superior e uma versão do *kernel* 2.4.2 que necessita ser modificada. O Red Hat 7.0 já inclui um conjunto de ferramentas necessárias para suportar o IPv6.

O Mobile IPv6 está implementado como módulo do *kernel*. Este é introduzido no *source code* do *kernel* através de um sistema de *patch*, que vai adicionar um conjunto de ficheiros ao *source code* e vai modificar outros ficheiros para que esta opção possa ser activada durante o processo de configuração.

Esta implementação do Mobile IPv6 foi preparada para a versão 2.4.2 do *kernel*. Esta pode ser usada com as versões 2.4.0 ou 2.4.1, mas não funciona com a 2.4.3 ou superior. Para obter o *source code*, deve-se consultar a secção **5.2.4. Compilar o kernel**.

Para aplicar o *patch*, deve-se obter o Mobile IPv6 em <http://www.mipl.mediapoli.com/>. Utilizou-se a versão 0.8.1 para um *kernel* 2.4.2, que estava disponível num ficheiro comprimido com o nome de “mipv6-0.8.1-v2.4.2.tar.gz”.

Para instalar, devem-se executar as seguintes etapas:

- Descomprimir o ficheiro do Mobile IPv6 com o comando “tar -zxvf mipv6-0.8.1-v2.4.2.tar.gz”
- Copiar o ficheiro “mipv6-0.8.1-v2.4.2.patch” que está dentro da directoria “mipv6-0.8.1-v2.4.2” para a directoria do *source code* do *kernel*.
- Testar se há existência de problemas com o comando “patch -p1 --dry-run < mipv6-0.8.1-v2.4.2.patch”.
- Se o comando anterior não indicar qualquer problema, pode-se correr o comando “patch -p1 < mipv6-0.8.1-v2.4.2.patch”.

- Configurar o kernel com as novas opções, executando o “make xconfig”.

Opção do <i>kernel</i>	Nome do parâmetro	Tipo	Opcional
Prompt for development and/or incomplete code/drivers	CONFIG_EXPERIMENTAL	y	x
Sysctl support	CONFIG_SYSCTL	y	x
/proc file system support	CONFIG_PROC_FS	y	x
Enable loadable module support	CONFIG_MODULES	y	x
Network packet filtering	CONFIG_NETFILTER	y	x
Unix domain sockets	CONFIG_UNIX	y	x
IP: policy routing	CONFIG_INET	y	x
The IPv6 protocol	CONFIG_IPV6	m	x
IPv6: enable EUI-64 token format	CONFIG_IPV6_EUI64	y	x
IPv6: disable provider based addresses	CONFIG_IPV6_NO_PB	y	x
IPv6: Mobility Support	CONFIG_IPV6_MOBILITY	m	x
MIPv6: AH Support	CONFIG_IPV6_MOBILITY_AH	y	✓
MIPv6: Debug messages	CONFIG_IPV6_MOBILITY_DEBUG	y	✓

Tabela 7: Opções do *kernel* na configuração do Mobile IPv6

Para além das opções referidas na **Tabela 7**, deve-se usar o conjunto de opções do *kernel* que estão definidas por omissão. Estas opções devem suportar as placas de rede que vão ser utilizadas assim como todas as opções necessárias para o funcionamento normal da rede. A opção “MIPv6: AH Support” activa o *IPSec Authentication Header* no funcionamento do Mobile IPv6, que protege os pacotes de sinalização. No entanto, máquinas com esta opção activada não interagem com outras em que a opção não esteja activada. A última opção da **Tabela 7** permite visualizar as mensagens de *debug* no sistema de *log* do Linux (normalmente em “/var/log/messages”).

Para além de todas as opções mencionadas anteriormente, a experimentação revelou que também se devem modificar as seguintes opções:

Opção do <i>kernel</i>	Nome do parâmetro	Tipo	Opcional
Set version information on all module symbols	CONFIG_MODVERSIONS	n	x
IPv6 Netfilter: IP6 tables support	CONFIG_IP6_NF_IPTABLES	y	x
IPv6 Netfilter: Limit match support	CONFIG_IP6_NF_MATCH_LIMIT	y	x
IPv6 Netfilter: netfilter MARK match support	CONFIG_IP6_NF_MATCH_MARK	y	x
IPv6 Netfilter: Packet filtering	CONFIG_IP6_NF_FILTER	y	x
IPv6 Netfilter: Packet mangling	CONFIG_IP6_NF_MANGLE	y	x
IPv6 Netfilter: netfilter MARK target support	CONFIG_IP6_NF_TARGET_MARK	y	x

Tabela 8: Outras opções do *kernel* na configuração do Mobile IPv6

Depois de ter o *kernel* compilado e instalado em todas as máquinas, é necessário copiar a directoria dos módulos (“/lib/modules/2.4.2”) para todas as máquinas.

O Mobile IPv6 é também constituído por ferramentas ao nível do utilizador, assim como *scripts* de inicialização do módulo. Para instalar estes ficheiros, deve-se:

- Entrar na directoria “mipv6-0.8.1-v2.4.2/src/userspace”
- Executar o comando “make”
- Executar o comando “make install”

Para obter um ambiente IPv6 a funcionar em Linux, deve-se consultar o <http://www.bieringer.de/linux/IPv6/>. Aqui pode-se encontrar um guia de instalação do IPv6. É importante obter deste *site* um conjunto de *scripts* em <http://www.bieringer.de/linux/IPv6/IPv6-HOWTO/scripts/current/index.html>, para integrar a configuração do IPv6 no sistema já existente para o IPv4.

É também necessário instalar em todos os routers, que neste caso são o HA e o CN, o *radvd* – Router advertisement Daemon. Este software gera as mensagens que são utilizadas pelo MN para se auto-configurar quando muda de rede. Pode-se obter o *radvd* a partir de <ftp://ftp.cityline.net/pub/systems/linux/network/ipv6/radvd/>. A versão 0.6.2 já suporta as extensões de mobilidade.

Para utilizar o Mobile IPv6 é necessário que o IPv6 esteja configurado com os endereços e rotas suficientes para que haja conectividade entre as 3 máquinas. Para

obter informações referentes à maneira de configurar a rede em IPv6 deve-se consultar a secção **5.2.6. Configurar a rede em IPv6**.

Foram criados endereços com um *scope* do tipo *site local*, e utilizou-se os últimos 64 bits dos endereços que são criados automaticamente quando o IPv6 arranca na máquina. O IPv6 pode ser activado pelo comando “`insmod ipv6`” ou no arranque da máquina, configurando os *scripts* obtidos anteriormente.

Máquina	Endereço	Rota
HA	fec0::1:<mac bits00>/64 (eth0) fec0::2:<mac bits01>/64 (eth1)	fec0:0:3::/64 via fec0::1:<mac bits10>
CN	fec0::1:<mac bits10>/64 (eth0) fec0::3:<mac bits11>/64 (eth1)	fec0:0:2::/64 via fec0::1:<mac bits00>

Tabela 9: Configuração dos endereços na montagem do Mobile IPv6

Na **Tabela 9**, os últimos 64 bits dos endereços foram substituídos por uma referência, uma vez que foram criados a partir do endereço MAC e não têm, por isso, qualquer significado especial. Por outro lado, os primeiros 64 bits identificam o tipo de endereço e a rede a que o interface pertence.

É também necessário configurar o **radvd** para que o Mobile IPv6 funcione.

Ao instalar uma versão em RPM, pode-se encontrar o ficheiro de configuração em “`/etc/radvd.conf`”. Nas versões de *source code*, existe um ficheiro dentro da directoria principal do **radvd** com o nome de “`radvd.conf.example`”. Pode-se copiar este ficheiro para “`/etc/radvd.conf`” e editá-lo.

A configuração do CN é a seguinte:

```
interface eth1
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    AdvHomeAgentFlag off;

#
# new EUI-64 prefixes
#
    prefix fec0:0:0:3::0/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

A configuração do HA é a seguinte:

```
interface eth1
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    AdvHomeAgentFlag on;

    prefix fec0:0:0:2::0/64
    {
        AdvOnLink on;
```



```

AdvAutonomous on;
AdvRouterAddr on;
};
};

```

É de realçar que no HA o campo “AdvHomeAgentFlag” está com o valor de “on”. Por omissão, vem definido o interface “eth0”, mas neste trabalho o MN liga-se ao “eth1” do HA e do CN. O campo “AdvRouterAddr” também foi mudado para “on”. O prefixo tem de acabar com um “0”.

Configuração do Mobile IPv6

A configuração do Mobile IPv6 baseia-se em apenas dois ficheiros:

- “/etc/mip6_acl.conf” - lista de acessos do HA
- “/etc/sysconfig/network-mip6.conf” – configuração do HA / MN / CN.

A lista de acessos do HA é configurada pelo seguinte sistema:

- Para permitir o acesso de todos os MN com um endereço que tenha o prefixo fec0::2::/64, adiciona-se a seguinte linha: “ALLOW fec0::2::/64”
- Para negar o acesso a todos cujos endereços têm o prefixo fec0::3:abcd::/80, adiciona-se “DENY fec0::3:abcd::/80”

Os três elementos do Mobile IPv6 são configurados num único ficheiro. São ignorados os parâmetros que não dizem respeito a essa máquina.

Configuração do CN

O CN é configurado apenas pelo primeiro parâmetro do ficheiro, ou seja, basta indicar no início o seguinte:

```
“FUNCTIONALITY=cn”
```

Ao activar o *Authentication Header* no *kernel* é necessário definir pelo menos um dos dois campos:

```
“MD5KEY=”
“SHA1KEY=”
```

O MD5 pode ter até 22 caracteres e o SHA1 até 27. É possível utilizar os algoritmos, as letras minúsculas e maiúsculas e ainda o “.” e o espaço.

Configuração do HA

O HA é configurado por 6 parâmetros, sendo 2 opcionais. O primeiro é o seguinte:

```
“FUNCTIONALITY=ha”
```

Para indicar o nível de *debug* que se pretende, utiliza-se o seguinte parâmetro:

```
“DEBUGLEVEL=0”
```

Os valores possíveis são de 0 a 7. Este valor não tem qualquer significado se o *kernel* não tiver a opção de *debug* do Mobile IPv6 activada.

Para escolher se os pacotes com um *scope* do nível *site local* são enviados para o MN, define-se o seguinte parâmetro:

```
“TUNNEL_SITELOCAL=yes”
```

Ao activar o *Authentication Header* no *kernel* é necessário definir pelo menos um dos dois campos:

```
“MD5KEY=”
```

```
“SHA1KEY=”
```

Estes campos devem ser iguais em todas as máquinas.

O último parâmetro é normalmente sempre igual, um vez que indica a localização da lista de acessos do HA.

```
“MOBILENODEFILE=/etc/mipv6_acl.conf”
```

Configuração do MN

O MN é configurado por 8 parâmetros, sendo 2 opcionais. O primeiro é o seguinte:

```
“FUNCTIONALITY=mn”
```

Para indicar o nível de *debug*, utiliza-se o seguinte parâmetro:

```
“DEBUGLEVEL=1”
```

É necessário indicar o endereço do MN:

```
“HOMEADDRESS=3ffe:2620:6:1234:abcd::2/64”
```

Também é preciso indicar o endereço do HA:

```
“HOMEAGENT=3ffe:2620:6:1234:abcd::1/64”
```

Este endereço tem de ser acessível pelo CN.

Ao activar o *Authentication Header* no *kernel* é necessário definir pelo menos um dos dois campos:

```
“MD5KEY=”
```

```
“SHA1KEY=”
```

Estes campos devem ser iguais em todas as máquinas.

O MN pode solicitar mensagens de anúncio do *router* com um intervalo mínimo definido por:

```
“RTR_SOLICITATION_INTERVAL=1”
```

Este valor está definido em segundos.

As solicitações contínuas ao *router* pelo MN, vão aumentando o seu intervalo de tempo entre elas até ao valor máximo de:

```
“RTR_SOLICITATION_MAX_SENDTIME=5”
```

Este valor está definido em segundos.

Utilizar o Mobile IPv6

O Mobile IPv6 arranca com o *script* de inicialização que é instalado na directoria “/etc/rc.d/init.d”, com o nome de “mobile-ip6”. Se o IPv6 arrancar no *boot* do Linux, pode-se incluir este *script* no conjunto de serviços que arrancar na máquina. Caso contrário, arranca-se o Mobile IPv6 com o comando “/etc/rc.d/init.d/mobile-ip6 start”. É necessário proceder do mesmo modo em todas as máquinas.

O “mipdiag” é um programa de interacção com o Mobile IPv6. Este programa é instalado normalmente em “/usr/sbin”. Existe documentação referente a este executável no sistema de ajuda do Linux.

Problemas encontrados e a sua resolução

O módulo do IPv6 não se conseguia carregar porque dava erros de “unresolved symbols”.

▶ O *kernel* tem uma opção na configuração que faz com que os símbolos dos módulos incluam informação sobre a versão. Esta opção não é aconselhada em sistemas em desenvolvimento. Neste caso verifica-se que é necessário desactivar esta opção, como está especificado na **Tabela 8**.

O HA tinha o radvd a correr mas o MN não conseguia auto-configurar o interface.

▶ No ficheiro de configuração do **radvd** existe um parâmetro que indica em que interface se devem anunciar as mensagens. Este é por omissão o “eth0” e nesta montagem o MN liga-se ao “eth1”.

O HA e o CN estavam a auto-configurar-se com as mensagens do radvd.

▶ Uma máquina com funções de *router* não se deve auto-configurar. Deve-se evitar que as mensagens do **radvd** sejam transmitidas para interfaces desnecessários. Para configurar um *router* em IPv6 deve-se consultar a secção **5.2.6. Configurar a rede em IPv6**.

Ao carregar o módulo do Mobile IPv6, o “ping6” deixava de funcionar e gerava o erro: “ping: bind icmp socket: Connect assign requested address”.

▶ Este erro acontece devido a um *bug* no Mobile IPv6. A resolução passa por verificar se o MN já se auto-configurou com as mensagens vindas do HA. Depois do interface estar configurado, deve-se remover manualmente o módulo e iniciá-lo de novo com o *script* que se encontra em “/etc/rc.d/init/mobile-ip6”.

O MN não conseguia comunicar com o CN, estando ligado ao HA.

▶ Este problema surge normalmente em duas situações: não existem rotas estabelecidas para tal ou o *forwarding* em IPv6 não está activo no CN e no HA. Para resolver ambos os casos, deve-se consultar a secção **5.2.6. Configurar a rede em IPv6**.

O “ping6” gera o erro: “connect: invalid argument”.

► O “ping6” necessita da indicação do interface que se pretende usar. Este problema deverá ser resolvido em futuras versões. Para indicar o interface, deve-se adicionar a seguinte opção “-I ethX”.

O “tracpath6” não funcionava com o MN ligado ao CN.

► Este problema prende-se com a própria questão da mobilidade, para a qual ainda só existem propostas de resolução.

Testes e resultados

Tendo em conta os testes realizados com o Mobile IPv4, optou-se pelos seguintes testes para o Mobile IPv6:

- **Conectividade** – verificar se é possível comunicar com o MN quando está ligado ao CN.
- **Hand-off** – testar a capacidade de mudança entre redes e o tempo que é necessário para tal.
- **Performance** – capacidade máxima de transferência de dados, em comparação com uma ligação normal.
- **Atraso** – valor médio do atraso introduzido pelo uso da mobilidade.

Outros testes que se adaptavam a este tipo de experiência mas que não podem ser realizados com esta montagem são os de:

- **Carga** – capacidade do HA suportar vários MNs ligados em redes visitadas. Isto implica um número elevado de MNs.
- **Mobilidade wireless** – funcionamento do Mobile IPv6 num sistema que utilize ligações sem fios.
- **Macro-mobilidade** – funcionamento do sistema quando o MN se desloca para uma rede distante do HA. O conceito de distante implica atrasos e menor largura de banda do que de uma rede local.

Testes de conectividade

Utilizou-se o comando “ping” para estes testes. Depois do MN se ter registado com sucesso no FA, fizeram-se os seguintes testes de *ping*:

Origem	Destino	Resultado
MN	HA	✓
MN	CN	✓
Outra máquina da rede	MN	✓

Tabela 10: Teste de conectividade do Mobile IPv6

A ligação à outra máquina da rede era feita a partir de um terceiro interface do CN. Esta máquina funciona como outro CN e por isso tem configurações semelhantes ao CN utilizado na montagem.

Testes de *Hand-off*

Os testes foram realizados entre o HA e CN. Os resultados foram os seguintes:

- A primeira vez que se liga o MN ao CN, este demora cerca de 6 segundos a retomar a comunicação. Este intervalo de tempo inclui a troca de ligação e a configuração do interface.
- Quando se liga o MN a uma máquina a que já esteve ligado, este demora cerca de 3 segundos a retomar a ligação. Esta situação aplica-se no caso do MN regressar ao HA ou de se ligar de novo ao CN.

Testes de performance

Verificou-se que o programa de FTP não conseguia criar uma ligação, apesar de aparentemente suportar o IPv6. Deste modo, não foi possível testar a performance do Mobile IPv6. No entanto foi possível criar uma sessão **ssh** que permitiu verificar a possibilidade de criar ligações TCP.

Testes de atraso

Os testes foram realizados entre o MN e o CN no primeiro caso, e entre o MN e HA no segundo. Esta diferença deve-se ao facto de ser necessário o mesmo número de *hops*. Utilizou-se o comando “ping6” para obter valores do atraso quando o MN está ligado na sua *Home Network* ou quando está ligado ao CN.

Tipo de ligação	Mínimo	Médio	Máximo
MN na sua rede	814 μ s	931 μ s	1 039 μ s
MN na rede visitada	832 μ s	955 μ s	1 209 μ s

Tabela 11: Teste do atraso do Mobile IPv6

Verifica-se que a diferença de tempo entre as duas situações é mínima. Em outros testes verificou-se que um “ping6” entre o MN e o CN, quando este está na rede visitada, tem um tempo inferior do que quando está na sua rede. O facto pode ser explicado pelo uso do *Route Optimization* no Mobile IPv6.

5.2.4. Compilar o *kernel*

Para compilar o *kernel* deve-se obter o *source code* a partir da Internet em <http://www.kernel.org> ou de um CD da distribuição de Linux. Este deve-se localizar no “/usr/src/linux-2.x.x”. Certas distribuições de Linux incluem na *source* um conjunto de ficheiros de configuração pré-definidos para vários sistemas. Estes ficheiros normalmente encontram-se numa subdirectoria denominada por “configs”. De qualquer maneira, pode-se sempre encontrar um ficheiro com a configuração por omissão em “arch/i386/defconfig”. Para usar este ou outros ficheiros como base para configurar o *kernel* deve-se copiá-lo para a directoria raiz da *source* com o nome de “.config”.

Para editar a configuração utiliza-se o comando “make xconfig”, estando na directoria de raiz da *source*. Mais métodos para configurar estão documentados no ficheiro “README” que se encontra nessa directoria.

Para compilar o *kernel* deve-se executar a seguinte sequência de comandos:

- “make dep”
- “make bzImage”
- “make modules”
- “make modules_install”

É preciso ter atenção que este último comando cria um conjunto de ficheiros na directoria “/lib/modules/2.x.x”. Esta terá o nome da versão do *kernel* que se compilou. Logo, esta acção irá destruir os módulos que existam para uma versão do *kernel* com o mesmo nome.

A imagem do *kernel* encontra-se em “arch/i386/boot/bzImage”. Esta deve ser copiada para “/boot” e depois modifica-se o ficheiro de configuração do “lilo” ou outro sistema de arranque de Linux para incluir este *kernel* como opção. As modificações ao “/etc/lilo.conf” seriam adicionar parâmetros tais como:

```
image=/boot/bzImage
    label=mobileIP
    read-only
    root=/dev/hda5
```

Este último parâmetro indica a localização da partição onde se encontra o “/boot” e deve ser semelhante à que consta na configuração para a imagem do *kernel* anterior. Depois de modificar este ficheiro corre-se o comando “lilo” para activar as modificações.

5.2.5. Instalar as placas *wireless* Cisco Aironet 340

Para instalar estas placas, é necessário activar o suporte PCMCIA no *kernel*.

Opção do <i>kernel</i>	Nome do parâmetro	Tipo	Opcional
PCMCIA/CardBus support	CONFIG_PCMCIA	y	✘
CardBus support	CONFIG_CARDBUS	y	✘

Tabela 12: Opções do *kernel* para o suporte PCMCIA

Depois de compilar o *kernel* e os módulos, deve-se obter uma nova versão dos módulos para esta placa, a partir do *download* das ferramentas PCMCIA com o nome de “pcmcia-cs” que existem em <http://www.sourceforge.net>.

Descomprimir o ficheiro e dentro da directoria “pcmcia-cs-3.1.xx” que foi criada, existe uma sub-directoria “wireless”.

Executar o comando “make”.

Copiar o ficheiro “airo.o” e “airo_cs.o” para a directoria dos módulos do *kernel*, substituindo os existentes.

Inserir a placa no slot PCMCIA.

Activar o modo *adhoc* com o comando “echo “Mode: adhoc” > /proc/driver/aironet/eth0/Config”.

Utilizar o mesmo SSID (por exemplo: “exp”) em todas as máquinas, inserindo-o com o comando “echo exp > /proc/driver/aironet/eth0/SSID”

Para mudar o endereço do interface utiliza-se o comando “ifconfig”, tal como as outras placas *ethernet*.

5.2.6. Configurar a rede em IPv6

Para configurar os endereços IPv6, assim como as rotas, existe um novo comando em Linux. Este permite configurar IPv4 e IPv6, pelo que é sempre necessário indicar o protocolo.

Quando se configura o IPv6, utiliza-se o comando “ip” com a opção “-f inet6”, para indicar o protocolo.

As funções mais comuns que são utilizadas com este comando estão descritas na seguinte lista:

- “ip -f inet6 addr sh” – visualiza os endereços dos vários interfaces.
- “ip -f inet6 addr add dev ethX <endereço IPv6>/M” – adiciona o endereço “<endereço IPv6>” com “M” bits de máscara, ao interface “ethX”.
- “ip -f inet6 addr del dev ethX <endereço IPv6>/M” – remove o endereço “<endereço IPv6>” com “M” bits de máscara, ao interface “ethX”.
- “ip -f inet6 route sh” – visualiza as várias rotas existentes.
- “ip -f inet6 route add dev ethX <rede IPv6> via <endereço IPv6>” – adiciona uma rota para a “<rede IPv6>” a partir do *router* com o endereço “<endereço IPv6>” que é acessível pelo interface “ethX”.
- “ip -f inet6 route del dev ethX <rede IPv6> via <endereço IPv6>” – remove a rota para a “<rede IPv6>” a partir do *router* com o endereço “<endereço IPv6>” que é acessível pelo interface “ethX”.

Outras configurações do IPv6 podem ser modificadas pela escrita no sistema “/proc” do Linux. Para configurar uma máquina como *router*, deve-se executar os seguintes comandos:

```
“echo 1 > /proc/sys/net/ipv6/conf/all/forwarding” – activa o IP forwarding.
```

```
“echo 0 > /proc/sys/net/ipv6/conf/all/autoconf” – desactiva a autoconfiguração dos interfaces.
```

Para configurar uma *workstation*, deve-se escolher uma configuração contrária à anterior, ou seja:

```
“echo 1 > /proc/sys/net/ipv6/conf/all/forwarding”.
```

```
“echo 0 > /proc/sys/net/ipv6/conf/all/autoconf”.
```

Existe ainda pouca documentação para estas funções no Linux, sendo por isso necessário recorrer à Internet.

5. Conclusões

Do trabalho prático pode-se concluir que o Mobile IPv4 está mais desenvolvido que o Mobile IPv6. No entanto, este desenvolvimento tem as suas limitações e mesmo que sejam corrigidos os problemas detectados durante a experimentação, não poderá implementar funcionalidades que são únicas dos IPv6. Do trabalho prático infere-se também que o Mobile IPv6 é mais simples, robusto e escalável do que a versão 4.

Em relação ao contacto com o IPv6, pode-se concluir que este é de facto o futuro em termos de redes de computadores e redes móveis de dados. Mesmo que não se considerasse as características e as funcionalidades acrescentadas, a questão do número de endereços disponíveis seria suficiente para chegar a esta conclusão.

O trabalho com o Cellular IPv4 permite apenas concluir que necessita de muito desenvolvimento, antes de se propor como alternativa na resolução da questão da micro-mobilidade.

Tendo em conta a investigação teórica destes assuntos, pode-se ainda concluir que o Mobile IPv6 está a evoluir para uma verdadeira solução de macro-mobilidade. Este está dependente da evolução do IPv6, que está a ter grande apoio por parte dos institutos de ensino e investigação e por parte das grandes empresas da informática e telecomunicações.

O Mobile IPv4 continua a ser desenvolvido, apesar dos esforços nesse sentido serem menores, uma vez que este já tem vários anos de evolução.

Em termos do uso do Mobile IPv6 nas redes móveis de 3ª geração, é difícil prever qual será o seu papel, uma vez que existem várias abordagens por parte das grandes empresas de telecomunicações. Empresas que apostam no Mobile IPv6 têm investido muito no seu desenvolvimento e na sua divulgação.

A análise teórica do Cellular IPv6 permite concluir que este poderá colmatar os problemas de micro-mobilidade do Mobile IPv6, mas o grau de evolução implica que a sua implementação será posterior à do Mobile IPv6.

6. Legado resultante do trabalho

Do trabalho desenvolvido neste projecto, existem um conjunto de elementos que podem ajudar outros a continuar o estudo dos temas abordados.

Neste documento pode-se encontrar um guia de instalação e de resolução de problemas sobre o Mobile IPv4 e o Mobile IPv6. Existe igualmente um estudo preparatório do Cellular IPv4.

No *site* do projecto encontram-se apontadores para os principais recursos na Internet relacionados com a mobilidade IP. Pode-se ainda encontrar um conjunto de artigos relacionados com o projecto.

Glossário

Kernel	Núcleo do sistema operativo. Fornece um interface com o <i>hardware</i> .
HA	<i>Home Agent</i> .
FA	<i>Foreign Agent</i> .
MN	<i>Mobile Node</i> .
Source	Utilizado como sinónimo de <i>source code</i> , que significa código fonte. O software que seja distribuído em <i>source code</i> , necessita de ser compilado antes de poder ser utilizado.
Home network	Rede onde se situa o <i>Home Agent</i> e a rede de origem do <i>Mobile Node</i> .
Hand-off	Processo de mudança de rede por parte do <i>Mobile Node</i> .
Wireless	Sem fios, ou seja, utilização de meios aéreos para comunicação (infravermelhos ou RF).
FTP	<i>File Transfer Protocol</i> .
Home Agent	Entidade que se situa na rede de origem do <i>Mobile Node</i> e que redirecciona o tráfego para a rede onde este se encontrar.
Foreign Agent	Entidade que se situa na rede visitada pelo <i>Mobile Node</i> e que recebe e retransmite o tráfego para o <i>Mobile Node</i> .
Mobile Node	Entidade com a capacidade de mudar de rede sem perder a conectividade.
Gateway	Implementa serviços de nível superior ao encaminhamento. Também conhecido por <i>application gateway</i> . Utilizado historicamente como sinónimo de <i>router</i> .
Mobile Host	Semelhante ao <i>Mobile Node</i> , mas no contexto do Cellular IP.
CN	Correspondent Node.
Correspondent Node	Qualquer máquina que comunica com o <i>Mobile Node</i> .
BS	<i>Base Station</i> .
Base Station	Máquina que comunica com o <i>Mobile Host</i> a partir de uma ligação sem fios.
Scope	Contexto do endereço de rede (<i>link, site, global</i>).

Bibliografia

- [1].RFC2002 – IP Mobility Support – C. Perkins – Outubro de 1996
- [2].draft-ietf-mobileip-rfc2002-bis-06.txt – IP Mobility Support for IPv4, revised– C. Perkins – Junho de 2001
- [3].RFC2344 – Reverse Tunneling for Mobile IP – G. Montenegro – Maio de 1998
- [4].RFC2794 – Mobile IP Network Access Identifier Extension for IPv4 – P. Calhoun – Março de 2000
- [5].RFC2977 – Mobile IP Authentication, Authorization, and Accounting Requirements – S. Glass, T. Hiller, S. Jacobs, C. Perkins – Março de 2000
- [6].draft-ietf-aaa-diameter-mobileip-01.txt – Diameter Mobile IP Extensions – Pat R. Calhoun, Charles E. Perkins - Março de 2001
- [7].draft-elmalki-mobileip-fast-handoffs-03.txt – Fast Handoffs in Mobile IPv4 - Karim El Malki, Hesham Soliman – Setembro de 2000
- [8].draft-ietf-mobileip-hawaii-01.txt – IP micro-mobility support using HAWAII – R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, L. Salgarelli – Julho de 2000
- [9].draft-ietf-mobileip-optim-10.txt – Route Optimization in Mobile IP - Charles Perkins, David B. Johnson – Novembro de 2000
- [10].RFC2006 – The Definitions of Managed Objects for IP Mobility Support – D. Cong, M. Hamlen, C. Perkins – Outubro de 1996
- [11].draft-ietf-mobileip-ipv6-13.txt – Mobility Support in IPv6 – David B. Johnson, Charles Perkins – Novembro de 2000
- [12]. *MobiCom' 96* – Mobility Support in IPv6 – David B. Johnson, Charles Perkins – Novembro de 1996
- [13].draft-ietf-mobileip-fast-mipv6-01.txt – Fast Handovers for Mobile IPv6 – G. Tsirtsis, A. Yegin, C. Perkins, G. Dommety, K. El-Malki, M. Khalil – Abril de 2001
- [14].draft-yegin-mobileip-nrouting-01.txt – Mobile IPv6 Neighborhood Routing for Fast Handoff – Alper E. Yegin, Mohan Parthasarathy, Carl Williams – Novembro de 2000

- [15].draft-ietf-mobileip-hmipv6-02.txt – Hierarchical MIPv6 mobility management – Hesham Soliman, Claude Castelluccia, Karim El-Malki, Ludovic Bellier – Fevereiro de 2001
- [16].draft-tsao-mobileip-dualstack-model-02.txt – Mobility Support for IPv4 and IPv6 Interconnected Networks based on Dual-Stack Model – Shiao-Li Tsao, Jen-Chi Liu, Wolfgang Boehm – Fevereiro de 2000
- [17].draft-ietf-mobileip-cellularip-00.txt – Cellular IP – A. T. Campbell, J. Gomez, C-Y. Wan, S. Kim, Z. Turanyi, A. Valco – Janeiro de 2000
- [18].draft-shelby-seamoby-cellularipv6-00.txt – Cellular IPv6 - Zach D. Shelby, Dionisios Gatzounas, Andrew Campbell, Chieh-Yih Wan – Novembro de 2000
- [19].draft-ietf-seamoby-paging-requirements-01.txt – Requirements and Functional Architecture for an IP Host Alerting Protocol – James Kempf, Claude Castelluccia, Pars Mutaf , Nobuyasu Nakajima, Yoshihiro Ohba, Ramachandran Ramjee, Yousuf Saifullah, Behcet Sarikaya, Xiaofeng Xu – Maio de 2001
- [20].draft-ietf-seamoby-ct-reqs-00.txt – General Requirements for a Context Transfer Framework - Hamid Syed, Gary Kenward, Pat Calhoun, Madjid Nakhjiri, Rajeev Koodli, Kulwinder Atwal, Mark Smith, Govind Krishnamurthi – Maio de 2001
- [21].draft-ietf-seamoby-context-transfer-problem-stat-01.txt – Problem Description: Reasons For Performing Context Transfers – O. H. Levkowitz, P. R. Calhoun, G. Kenward, H. M. Syed, J. Manner, M. Nakhjiri, G. Krishnamurthi, R. Koodli, K. S. Atwal, M. Thomas, M. Horan, P. Neumiller - Maio de 2001
- [22].Design and Analysis of Cellular Mobile Data Networks – A. Valco – Outubro de 1999
- [23].White Paper – IPv6 – Enabling the Mobile Internet – Nokia – 2001
- [24].White Paper – IPv6 – Introducing to IPv6 in 2G and 3G mobile networks – Nokia – 2001
- [25].White Paper – IPv6 – Transition to IPv6 in 2G and 3G mobile networks – Nokia – 2001

ANEXO A: Resultados da utilização do Cellular IPv4

Gateway

```
* Checking network connectivity.....
ip addr of peer machine is = 194.117.24.254
PING 194.117.24.254 (194.117.24.254) from 194.117.24.66 : 56(84) bytes
of data.64 bytes from 194.117.24.254: icmp_seq=0 ttl=255 time=2.008
msec
```

```
--- 194.117.24.254 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/mdev = 2.008/2.008/2.008/0.000 ms
inescp-gw3.inescn.pt (194.117.24.254) at 00:00:0C:00:E5:5D
temporary
```

```
* Checking network connectivity.....
ip addr of peer machine is = 194.117.24.135
PING 194.117.24.135 (194.117.24.135) from 194.117.24.66 : 56(84) bytes
of data.64 bytes from 194.117.24.135: icmp_seq=0 ttl=255 time=1.542
msec
```

```
--- 194.117.24.135 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/mdev = 1.542/1.542/1.542/0.000 ms
socrates.inescn.pt (194.117.24.135) at 00:00:1B:32:95:2F
temporary
```

```
* Checking network connectivity.....
ip addr of peer machine is = 194.117.24.167
PING 194.117.24.167 (194.117.24.167) from 194.117.24.66 : 56(84) bytes
of data.64 bytes from 194.117.24.167: icmp_seq=0 ttl=255 time=1.224
msec
```

```
--- 194.117.24.167 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/mdev = 1.224/1.224/1.224/0.000 ms
azul.inescn.pt (194.117.24.167) at 00:A0:24:9F:DB:B6
temporary
```

```
User level filter, protocol ALL, raw packet socket
User level filter, protocol ALL, raw packet socket
User level filter, protocol ALL, raw packet socket
```

Base Station 1

```
* Checking network connectivity.....
ip addr of peer machine is = 194.117.24.66
PING 194.117.24.66 (194.117.24.66) from 194.117.24.167 : 56(84) bytes
of data.
64 bytes from skyrack.inescn.pt (194.117.24.66): icmp_seq=0 ttl=255
time=945 usec

--- 194.117.24.66 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.945/0.945/0.945/0.000 ms
syrack.inescn.pt (194.117.24.66) at 00:C0:58:08:0D:37
temporary
Kernel filter, protocol ALL, raw packet socket
Kernel filter, protocol ALL, raw packet socket
Finish
Kernel filter, protocol ALL, raw packet socket
0:40:96:47:2c:d3
Hostname                : bs1.inescn.pt
```

Base Station 2

```
* Checking network connectivity.....
ip addr of peer machine is = 194.117.24.66
PING 194.117.24.66 (194.117.24.66): 56 data bytes
64 bytes from 194.117.24.66: icmp_seq=0 ttl=255 time=1.6 ms

--- 194.117.24.66 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1.6/1.6/1.6 ms
syrack.inescn.pt (194.117.24.66) at 00:C0:58:08:0D:37
temporary
User level filter, protocol ALL, raw packet socket
User level filter, protocol ALL, raw packet socket
Finish
User level filter, protocol ALL, raw packet socket
0:40:96:48:8f:f1
Hostname                : bs2.inescn.pt
```

Mobile Host

```
connection established 5
connection established 6
connection established 7
Kernel filter, protocol ALL, TURBO mode (447 frames), raw packet
socket
bgerror failed to handle background error.
Original error: error flushing "sock5": broken pipe
Error in bgerror: invoked "break" outside of a loop
```