

Categoria: **Público**

WG: **MIP6**

15.04.2004

Análise de Cenários de Mobilidade IPv6

Copyright Notice

Copyright © Task Force Portuguesa IPv6. All Rights Reserved.

1. Sumário

Este documento introduz um conjunto de requisitos e directivas relacionadas com mecanismos de *mobilidade IPv6*, abrangendo tecnologia *wireless* (e.g., WLANs) e tecnologia celular. Além dos mecanismos e configurações passíveis de utilização com IPv6 nativo, são ainda abordados e analisados *mecanismos de transição*.

O objectivo principal deste documento é o de fornecer um levantamento do estado actual de IPv6 e ainda, fornecer configurações base, que possam ajudar os operadores portugueses, e o público em geral, na migração para IPv6, em cenários de mobilidade.

O documento apresenta-se organizado do seguinte modo. Na secção dois apresenta-se terminologia utilizada no documento. Na secção três é descrito o contexto do tema abordado neste documento. Na secção quatro é apresentado o conceito de *Mobilidade IP*, incluindo uma breve comparação das características dos mecanismos de mobilidade IPv6 e IPv4. Na secção cinco descrevemos mecanismos de IPv6 para WLANs, e na secção seis para tecnologia celular. Na secção sete apresenta-se mecanismos de transição que possam ser aplicados nos cenários mencionados. Finalmente, concluímos na secção oito.

Índex

1. SUMÁRIO	1
2. TERMINOLOGIA	3
3. INTRODUÇÃO	5
4. MOBILIDADE IP	6
4.1. Funcionamento Básico	6
4.2. MIPv4 vs. MIPv6	8
5. IPV6 E WLANS	9

5.1. Suporte em Sistemas Operativos	9
5.2. Pacotes para Suporte de MIPv6	9
5.2.1. MIPL	9
5.2.2. Lancaster Mobile IPv6	10
5.2.3. USAGI	11
5.2.4. KAME MIPv6	11
5.3. Equipamento WLAN	12
5.3.1. Access Points	12
5.3.2. Placas Wireless PCMCIA e PCI	12
5.4. Plataforma Básica de IPv6 em WLANs	12
5.4.1. Linux, MIPL	14
5.4.2. KAME/MIPv6	24
6. IPV6 APLICADO A TECNOLOGIAS CELULARES	25
6.1. Suporte IPv6	27
6.2. Symbian 7.0	27
6.3. Testes	28
7. CENÁRIOS DE TRANSIÇÃO	29
7.1. Classificação dos Mecanismos de Transição	29
7.1.1. Pilha Dupla (Dual Stack)	30
7.1.2. Mecanismos de Tradução	31
7.1.3. NAT-PT	32
7.1.4. TRT - Transport Relay Translator	38
7.2. Túneis	39
7.2.1. Túneis Baseados em Endereços IPv6-IPv4 Compatíveis	40
7.2.2. 6to4	40
7.3. Cenários de Transição Aplicados a Redes GPRS	42
7.3.1. Cenário 1	43
7.3.2. Cenário 2	44
7.3.3. Cenário 3	45
8. RESUMO E CONCLUSÕES	45
9. CONTACTO DOS AUTORES	46
10. REFERÊNCIAS	47

2. Terminologia

3GPP	3rd Generation Partnership Project
ALGs	Applications Level Gateways
AP	Access Point
API	Application Program Interface
CIDR	Classless Inter-Domain Routing
CoA	Care-of-Address
DAD	<i>Duplicate Address Location</i>
DHAAD	Dynamic Home Agent Address Discovery
DHCPv6	Dynamic Host Configuration Protocol Version 6
DNS	Domain Name Server
FN	Foreign Network
FTP	File Transfer Protocol
GGSN	General Packet Radio Service Support Node
GPRS	General Packet Radio Service
HA	Home Agent
HM	Home Network
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
<i>IKE</i>	<i>Internet Key Exchange</i>
IMAP	Internet Message Access Protocol
IMS	Internet Protocol Multimedia System
IP	Internet Protocol
IPSec	Internet Protocol Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
LAN	Local Area Network
MAC	Media Access Control
MGW	Media Gateway
MIPL	Mobile IPv6 For Linux
MIPv4	Mobile Internet Protocol Version 4
MIPv6	Mobile Internet Protocol Version 6
MMS	Multimedia Messaging Service
MN	Mobile Node
MT	Mobile Terminal
MTU	Maximum Transmission Unit
NAT	Network Address Translator
NAT-PT	Network Address Translator – Protocol Translator
ND	Neighbor Discovery
NI	Nós Intermediários
P2P	Peer-to-Peer
PC	Personal Computer
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PDP	Packet Data Protocol
POP	Post Office Protocol
PPP	Point-to-Point Protocol

PSTN	Public Switched Telephone Network
QoS	Quality of Service
RA	Router Advertisement
RAM	Random Access Memory
RFC	Request For Comments
SIP	Simple Internet Protocol
SMS	Short Message Service
SMTP	Simple Mail Transport Protocol
SO	Sistema Operativo
SSH	Secure Shell
TCP	Transmission Control Protocol
TRT	Transport Relay Transport
TRT	Transport Relay Translator
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Application Protocol
WLAN	Wireless Local Access Network
XML	Extensible Markup Language

3. Introdução

A heterogeneidade que a Internet apresenta actualmente a nível de tecnologias permite o desenvolvimento, suporte e comercialização dos mais variados serviços *hipermedia* e *multimedia*, serviços estes que contribuem para a popularização da Internet e consequentemente, para o aumento exponencial dos seus utilizadores.

A recente introdução de tecnologias *wireless* como meio de acesso à Internet a partir de locais denominados de lazer público tais como parques, cafés, restaurantes, ou aeroportos reforçaram a componente de entretenimento que a Internet apresenta hoje em dia.

A adesão crescente de utilizadores e o emergir de dispositivos que requerem mais do que um interface de acesso à Internet levou à detecção de um problema básico na comunicação da Internet: a limitação do espaço de endereçamento de que o *Internet Protocol versão 4 (IPv4)* [\[IPv4\]](#) sofre: concebido na década de 60 com um intuito totalmente diferente daquele que hoje suporta, o seu desenho não previu os desafios actuais.

A limitação detectada levou à criação de mecanismos que permitem otimizar o espaço de endereçamento fornecido pelo IPv4: CIDR [\[CIDR\]](#), NATs [\[NAT\]](#), e a utilização de blocos de endereços privados [\[RFC1918\]](#) são actualmente mecanismos comuns em redes IP, que permitem uma melhor gestão do espaço de endereçamento. Mas, apesar de estas soluções permitirem aumentar a longevidade do espaço de endereçamento, não representam a solução óptima para a situação actual, nomeadamente, para os serviços disponibilizados pela Internet:

- o CIDR permite uma melhor gestão do espaço de endereçamento, mas não ajuda a otimizar a agregação e consequentemente, a sua contribuição para agregação de rotas é apenas parcial;
- para elementos que utilizam endereçamento privado, as comunicações só podem ser iniciadas pela estação que usa o endereço privado. De facto, este modelo só pode ser usado em ambientes cliente/servidor, em que a estação que usa o endereço privado funciona como cliente, enquanto que a estação que usa o endereço público desempenha o papel de servidor. Por conseguinte, é quebrado o modelo *end-to-end* [\[E2E\]](#) característico do protocolo IP e base da comunicação na Internet;
- para soluções baseadas em NAT, não é possível o uso de associações de segurança end-to-end. Adicionalmente, alguns serviços, tais como *peer-to-peer* (P2P) não funcionam na presença de um dispositivo NAT.

Ainda relacionado com endereçamento, o protocolo IP não suporta *autoconfiguração*, o que se revela um problema acima de tudo crítico para serviços IP em telefones da 3ª geração [\[RFC3314\]](#): a utilização de dispositivos com mais do que um interface de rede, e a gestão de redes heterogéneas não é possível sem um mecanismo de configuração automática de endereços.

Outros problemas detectados no protocolo IPv4 são a falta de segurança integrada, actualmente colmatada através de soluções de segurança tais como a utilização de IPSec [\[IPSec\]](#) e ainda, a impossibilidade de diferenciar tráfego: o IPv4 assume que todo o tráfego transportado tem os mesmos requisitos de rede. Consequentemente, o IPv4 não suporta *diferenciação de serviços* [\[QoS\]](#).

Este conjunto de problemas representa um pequeno subgrupo das limitações totais que levaram à criação e ao desenvolvimento durante a última década de uma nova versão do protocolo IP, a *versão seis (IPv6)* [\[IPv6\]](#) que, além de fornecer soluções para as limitações do IPv4, apresenta características que o tornam fulcral em cenários actuais que baseiem os seus serviços em IP, tais como cenários de mobilidade: o IPv6 foi pensado de base para suportar *mobilidade*, incluindo não só mecanismos de *autoconfiguração*, mas também de *optimização de*

encaminhamento e ainda, de *segurança*.

Na última década o IPv6 foi testado exaustivamente através de iniciativas tais como a rede virtual 6BONE [6BONE]. Após esta fase de testes e embora continuem a decorrer especificações com o objectivo de melhorar os seus mais variados aspectos, o IPv6 atingiu um estado de maturidade que permite a sua aplicação em cenários de produção e consequentemente, avançar para uma exploração comercial de serviços IPv6. Tal exploração só poderá ter sucesso, se desenvolvida e aplicada adequadamente.

Actualmente, a maioria dos fabricantes de equipamento de encaminhamento fornece suporte para as duas versões do protocolo IP, o que facilita a sua instalação em redes cabladas. No entanto, o mesmo não sucede no respeitante a cenários de mobilidade: os mecanismos de mobilidade IPv6, que fazem parte integrante do protocolo, encontram-se ainda em desenvolvimento, e existem escassas directivas de desenvolvimento e aplicação, que ajudem a compreender e facilitem a migração de IPv4 para IPv6.

Neste contexto, este documento apresenta o estado da arte de *mobilidade IPv6*, incluindo informação sobre os mecanismos passíveis de utilização e sua configuração. Pretende-se que esta contribuição ajude à compreensão e à aplicação das características do IPv6, de modo a facilitar o futuro desenvolvimento de soluções de mobilidade. Começamos por explicar brevemente os itens relacionados com mobilidade IPv6 e IPv4.

4. Mobilidade IP

A motivação para o desenvolvimento de mecanismos de mobilidade em redes IP representa uma consequência natural da integração de redes *wireless* na Internet. Até à introdução da tecnologia *wireless*, os utilizadores da Internet só podiam ter acesso a partir de locais pré-determinados, por exemplo, de casa, do emprego, da escola. A tecnologia *wireless* alterou o conceito de acesso à Internet: os serviços são actualmente “levados” até ao utilizador. Adicionalmente, a variedade de dispositivos móveis disponíveis continua a aumentar: telefones digitais celulares, *Personal Digital Assistants (PDAs)*, são alguns dos dispositivos que estão a alterar o conceito de acesso à Internet. No entanto, a sua utilização implica uma maior *mobilidade* não só entre diferentes localizações do ponto de vista geográfico, mas mesmo entre diferentes tecnologias.

O mecanismo denominado *Mobile IP (MIPv4)* [MIPv4] foi introduzido para colmatar algumas das necessidades requeridas pelo novo conceito de acesso, tais como a situação de um utilizador se movimentar entre duas redes, e necessitar de manter as comunicações já em curso. O MIPv4 é uma modificação do protocolo IP que permite aos elementos móveis (*Mobile Nodes, MNs*) continuar a receber datagramas independentemente do local de acesso à Internet. Por outras palavras, o objectivo máximo é permitir a dispositivos móveis manter as suas ligações enquanto se movem entre diferentes redes - *roaming* – e sem necessidade de alterar a sua configuração. Para tal, o MIPv4 atribui um identificador ao MN e ainda, à rota IP para esse nodo, o que permite independência total da camada física.

4.1. Funcionamento Básico

A motivação principal do desenvolvimento do MIPv4 foi a de permitir a um MN movimentar-se entre diferentes redes mantendo o seu endereço IP. Tal introduz vários problemas, tais como a manutenção de comunicações já em curso. Por exemplo, um utilizador com um dispositivo móvel pretende estabelecer uma transmissão de vídeo durante uma viagem de comboio. Durante essa viagem, o utilizador irá ter acesso à transmissão de vídeo a partir de diferentes redes. Supondo que o utilizador está a receber informação, esta, na forma de fluxos, tem como endereço destino o endereço inicialmente atribuído ao MN. Consequentemente, os fluxos necessitam de ser redireccionados para o novo endereço IP de modo transparente. O MIPv4 pretende portanto fornecer uma solução para este tipo de problemas, de modo transparente para

o utilizador, tal como é apresentado na Figura 1 [PER96].

Para tal, fornece ao MN dois endereços IP: o endereço inicial, normalmente fornecido pelo operador com o qual o MN tem uma subscrição (contrato) denomina-se *Home Address* e é estático, no sentido de que vai ser mantido durante a sessão. É usado, por exemplo, para identificar ligações TCP. O segundo endereço, *Care-of-Address (CoA)*, é alterado de cada vez que o MN se desloca para um novo ponto de acesso; identifica o número da rede visitada e portanto, identifica o novo local de acesso no respeitante à topologia de rede.

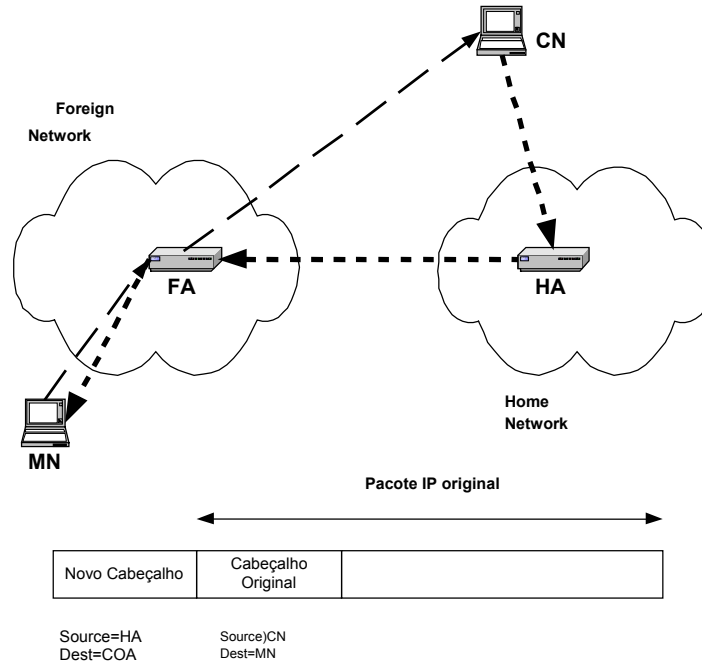


Figura 1: Modelo de Mobilidade IP

O *Home Address* permite ao MN ter conectividade IP na sua *Home Network (HN)*, i.e., na sua rede local original. Quando o MN se movimenta para outra rede – *Foreign Network (FN)* – um elemento de rede (*router*) denominado *Home Agent (HA)* na sua HN encarrega-se de interceptar todos os pacotes destinados para o MN e de os re-transmitir utilizando o CoA do MN. Para que tal seja possível, cada vez que o MN se movimenta, regista o seu novo CoA com o HA, que se encarrega de redireccionar o datagrama adequadamente. Tal implica a alteração do pacote, de modo a que o novo endereço destino seja o CoA.

Do mesmo modo, quando o CoA recebe o datagrama, este é novamente modificado, sendo o *Home Address* do MN introduzido. Portanto, o MIPv4 utiliza como técnica de redireccionamento de datagramas *tunneling*.

Três mecanismos básicos compõem o MIPv4:

- **descoberta do CoA:** baseia-se no mecanismo de detecção de Router Advertisements (RAs) [RFC1256]. Um RA é uma mensagem que transporta informação sobre *default routers*. A extensão para MIPv4 permite transmitir informação sobre um ou mais CoAs. Tais mensagens são denominadas *Agent Advertisements*, e são normalmente trocadas em intervalos regulares (ou a pedido) por HAs e FAs, que as utilizam para se darem a conhecer, ou para trocar CoAs;
- **registo do CoA:** após a obtenção de um CoA, o MN tem de dar a conhecer esse CoA

ao seu HA, de modo a permitir o redirecionamento de datagramas. O MN envia um pedido de registo contendo o seu CoA. Quando o HA recebe este pedido, adiciona a informação necessária ao redirecionamento na sua tabela de encaminhamento, aprova o pedido e envia uma resposta ao MN. Adicionalmente, pode haver autenticação/autorização envolvida no processo de registo;

- **tunneling para o CoA:** o mecanismo de redirecionamento baseia-se nos mecanismos de tunneling do IP, i.e., IP-em-IP.

As técnicas mencionadas representam os mecanismos base da mobilidade IP. No entanto, devido às características específicas de cada versão do protocolo IP, os mecanismos MIPv4 e MIPv6 apresentam características distintas, que apresentamos de seguida.

4.2. MIPv4 vs. MIPv6

O IPv6 apresenta características que permitem otimizar a mobilidade, incluindo mecanismos de configuração de endereços *stateless* [AC], e mecanismos de detecção de elementos na vizinhança, *Neighbor Discovery (ND)* [ND]. Adicionalmente, os mecanismos de endereçamento IPv6 facilitam um possível futuro *renumbering*.

O suporte de mobilidade em IPv6 (MIPv6) [MIP6] baseia-se no MIPv4, já apresentado. O MIPv6 mantém os conceitos de HN, HA, e a utilização de encapsulamento como método de redirecionamento de datagramas. Adicionalmente, introduz várias optimizações. Enquanto que a detecção de CoAs se mantém, o processo é simplificado devido aos mecanismos de autoconfiguração e ND. Tal evita a utilização de FAs.

As principais diferenças entre MIPv4 e MIPv6 são:

- maior espaço de endereçamento, permitindo atribuir a cada MN dois endereços (evitando a utilização do endereço do FA);
- obtenção automática de CoAs através de DHCPv6 ou de autoconfiguração, evitando a utilização de FAs;
- MIPv6 permite ao MN ter vários CoAs simultâneos;
- no MIPv4, quando um MN envia um datagrama a um CN, o endereço fonte do pacote corresponde ao *home address*; no MIPv6, o endereço fonte é o CoA primário. Opcionalmente, o MN pode ainda indicar o seu *home address*;
- o mecanismo de encaminhamento do MIPv4 baseia-se em *triangle routing* (sem a optimização adicional de encaminhamento); no caso de MIPv6, essa optimização encontra-se incluída de base;
- em MIPv4 (com optimização de *encaminhamento*), quando um pacote é enviado de um CN a um MN, o pacote tem de ser encapsulado e o seu endereço destino alterado para o CoA. No caso de MIPv6 o pacote utiliza a opção *Routing Header* do IPv6, sendo o endereço do último nodo intermediário alterado para o CoA e o endereço destino alterado para o *Home Address* do MN. Como estes 2 endereços representam endereços do MN, este não fica confuso;
- a utilização do cabeçalho opcional Routing Header suporta *multicast*;
- no MIPv4, sem o suporte de mecanismos especiais de tunneling e sem smooth handoffs, quando um MN se move para outra FN, os datagramas enviados ao CoA anterior são descartados. Em contraste, o MIPv6 suporta de base *smooth handoffs*: o encaminhador default da rede anterior funciona como um HA na FN, enquanto o MN se movimenta;
- o MIPv6 integra segurança de base, dado que todos os nos IPv6 integram de base mecanismos de autenticação forte e de cifra.

Nas próximas secções, passaremos a explicar diferentes cenários de mobilidade IPv6, nomeadamente, IPv6 em WLANs e IPv6 aplicado à tecnologia celular.

5. IPv6 e WLANs

Nesta secção, abordamos os mecanismos actuais de MIPv6 aplicáveis a WLANs. Apresentamos informação sobre mecanismos a utilizar e explicamos como criar uma plataforma básica de testes. Começa-se pelo suporte fornecido nos sistemas operativos (SOs) mais comuns.

5.1. Suporte em Sistemas Operativos

Nesta secção apresenta-se o estado de desenvolvimento do IPv6 para os sistemas operativos mais comuns, agrupando na Tabela 1 as características inerentes ao IPv6 por SO.

Tabela 1: Principais Características de IPv6 Suportadas em SOs

	BSD	Linux	Windows XP	Windows 2003	Mac OS	Solaris	HP-UX 11i
ICMPv6	Sim	Sim	Sim	Sim	Sim	Sim	Sim
MTU Discovery IPv6	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Neighbor Discovery	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Mecanismos de Transição	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Mobilidade	Sim	Sim	Não	Não	Sim	Não	Não
API IPv6	Sim	Sim	Sim	Sim	Sim	Sim	Sim
IPSec	Sim	Testes	Sim ¹	Sim ¹	Sim	Não	Sim
DHCPv6 ²	Testes	Testes	Não	Não	Não	Não	Sim

Embora a componente de mobilidade seja uma característica base do IPv6, os sistemas operativos mencionados não integram de base o *software* que permite utilizar MIPv6. Tal *software*, que apresentamos de seguida tem de ser instalado à parte.

5.2. Pacotes para Suporte de MIPv6

Os variados pacotes que implementam MIPv6 são descritos de seguida, tentando incluir as suas características principais, os seus problemas e ainda, as suas vantagens.

5.2.1. MIPL

O *Mobile IPv6 for Linux (MIPL)* [MIPL] é uma implementação *open-source* baseada na especificação [MIPv6]. Inicialmente desenvolvido no âmbito do *HUT Software Project*, foi

¹ IPSec funciona, mas apresenta algumas limitações.

² DHCPv6 refere-se a *stateful autoconfiguration*. Todos os SOs mencionados suportam a característica base de *stateless autoconfiguration*.

melhorado no *Telecommunications and Multimédia Lab, HUT*, no âmbito do projecto *GO/Core*. A última versão data de 8 de Janeiro de 2003 e apresenta as seguintes características básicas:

- mecanismo de segurança integrado (associações de segurança IPSec);
- recorre ao *Router Advertisement Daemon* (radvd) [\[NAR98\]](#) para suporte de autoconfiguração de endereços em cenários de mobilidade;
- a distribuição inclui o módulo *kernel*, *patches* e aplicações de monitorização e configuração (mipdiag);
- suporta versões do Red Hat superiores à versão 6.1, sendo necessário utilizar uma versão de *kernel* que esteja em conformidade com a versão MIPL a implementar;
- os pedidos de *Binding* são compatíveis com diferentes implementações de agentes locais;
- a lista de agentes locais pode ser adicionada nos dispositivos móveis;
- suporta o *Dynamic Home Agent Address Discovery (DHAAD)* para detecção automática de MN e HAs (utilizando endereços *anycast*).

Embora extremamente completo, o software apresenta os seguintes problemas:

- **implementação dos HAs :**
 - a *Hash List* que regista as informações actualizadas do posto móvel, não permite múltiplas entradas que partilhem a mesma chave. Esta propriedade impossibilita o registo de vários endereços locais para o mesmo agente local. Desta forma, apenas é suportado um agente local por dispositivo móvel;
 - é estabelecido um número fixo de túneis nos dispositivos móveis e agentes locais;
 - não é possível utilizar o mecanismo automático de gestão de chaves fornecido pelo *Internet Key Exchange (IKE)* [\[IKE\]](#) entre MN e CNS.
- **implementação dos Postos Móveis:**
 - não existe *reverse tunneling*;
 - quando um posto móvel entra numa rede estrangeira, não deverá executar o DAD (*Duplicate Address Location*) para o seu endereço local, e não deverá fazer *reply* de mensagens *ND* para o seu endereço local.

5.2.2. Lancaster Mobile IPv6

Esta implementação é considerada um marco histórico no desenvolvimento de MIPv6. A respectiva *package* está disponível em [\[LANC1\]](#) e foi desenvolvida pela Universidade de Lancaster, Computing Department. A versão actual ficou disponível em 3 de Junho de 1998, não sendo compatível com as especificações actuais do MIPv6.

As suas características principais são:

- inclui suporte para optimização do percurso;
- suporta aplicações que utilizem os protocolos UDP, TCP e ICMP;
- testado para *handovers* eficientes;
- inclui na instalação um módulo *kernel*;
- suporta *roaming MAC layer* para redes WaveLan;
- corre nas versões de *Linux Kernel* 2.1.9x e superiores.

Os seus problemas conhecidos são:

- o código fonte não é distribuído;
- suporta versões antigas de IP Móvel;
- existe pouca informação de suporte.

5.2.3. USAGI

O **UniverSAI playGround for IPv6** é um projecto [\[USAGI\]](#) japonês voluntário. O USAGI utiliza como código base o KAME (ver secção 5.2.4) e o Projecto WIDE [\[WIDE\]](#), para oferecer pilhas IPv6 e IPsec para Linux. As suas características principais são:

- baseado no Linux 2.4.21;
- suporta IPsec;
- suporta mobilidade IPv6;
- suporta mecanismos de eleição de *router*.

5.2.4. KAME MIPv6

O projecto *KAME* [\[KAME\]](#) surgiu em 1998 e nasceu do esforço conjunto de sete empresas japonesas, com o intuito de providenciar uma *stack open-source* de IPv6 e IPsec (para IPv4 e IPv6), para as variantes do sistema operativo BSD, tendo sido um dos projectos que mais impulsionou a implementação do IPv6. O suporte de mobilidade foi desenvolvido por um consórcio constituído pela Ericsson, NEC, Universidade de Keio e pelo grupo de mobilidade aferido ao projecto, nomeadamente Kame Mobile IPv6. Estas entidades estão, ainda, empenhadas em oferecer as últimas funcionalidades do MIPv6, patente na *stack* do Kame.

Actualmente, existem três implementações do MIPv6 para a *stack KAME*. No entanto, duas delas (Ericsson [\[ERICSSON\]](#), Universidade de Keio) encontram-se desactualizadas, pelo que apenas apresentamos a que se encontra actualizada, a implementação da NEC.

A implementação MIPv6 fornecida pela NEC encontra-se de acordo com as especificações mais recentes do MIPv6. Adaptada à pilha KAME, a implementação é apenas distribuída sob licença e é suportada em FreeBSD 4.x, NETBSD15. Apresenta as seguintes características:

- suporte para especificações recentes de MIPv6;
- fornece estatística no HA;
- suporta características do IKE (entre HA e MN, CN e MN);
- suporta *reverse tunneling*.

Itens em aberto são:

- detecção de vários *Home Addresses*;
- detecção de movimento assistida pela camada 2;
- *forwarding* de pacotes a partir do *link* anterior;
- configuração remota do *Home Address*.

Nesta secção, apresentámos suporte IPv6 em vários SOs e ainda, as implementações actuais de MIPv6, as suas características e problemas. Na próxima secção apresentamos equipamento WLAN passível de ser utilizado na criação de ambientes IPv6.

5.3. Equipamento WLAN

Nesta secção, apresenta-se uma comparação sobre algum equipamento WLAN, assinalando-se características específicas para IPv6.

5.3.1. Access Points

A Tabela 2 apresenta as principais especificações sobre o dispositivos aqui analisados. Teve-se em consideração os requisitos básicos de hardware para criar a bancada, dado que existe apenas um número reduzido de APs que suportam gestão remota através de interfaces IPv6.

Tabela 2: Principais Características de APs

	RoamAbout R2	AP 420 HP	Aironet 1200	AP 2220 NORTEL	AP 8500 3COM
Normas	802.11a e 802.11b	802.11b e 802.11g	802.11a, 802.11b e 802.11g	802.11a e 802.11b	802.11a e 802.11b
Nº Interfaces Wireless	2	2	2	2	2
Nº Máximo de Utilizadores	60	NE ³	NE	NE	253
Autenticação Radius	Sim	Sim	Sim	Sim	Sim

5.3.2. Placas Wireless PCMCIA e PCI

Actualmente, existe um elevado número de fabricantes de placas *wireless*, sendo a maioria desconhecida. Apesar da elevada variedade de marcas de placas *wireless* fornecida, verifica-se que todas elas usam um *chipset* de uma gama limitada de fabricantes de *chipsets*.

Sucedo ainda o caso de o mesmo vendedor usar diferentes *chipsets* nos vários produtos existentes no mercado. Por vezes, determinada placa de um certo modelo pode mesmo sofrer uma evolução para um *chipset* diferente, sendo no entanto mantida a mesma designação para o produto. Isto significa que uma placa poderá funcionar ou não dependendo da versão em questão.

A maioria das placas 802.11b no mercado utilizam o *chipset Intersil Prism II*, facilmente suportada nos mais variados SOs. Actualmente, começam a surgir tecnologias que permitem a obtenção de velocidades mais rápidas de transmissão de dados como a usada nas placas 802.11g. Estas poderão levar ao uso de diferentes *chipsets* que, muito provavelmente, apresentarão maiores limitações de compatibilidade.

As placas adequadas podem ser consultadas em [\[cards\]](#) onde se poderá encontrar uma lista exaustiva dos fabricantes de *hardware* e dos *chipsets* utilizados. Adicionalmente, informação sobre drivers pode ser obtida em [\[cards1\]](#).

5.4. Plataforma Básica de IPv6 em WLANs

Esta secção apresenta de forma detalhada toda a configuração necessária para criar uma

³ Não Especificado;

plataforma de testes MIPv6. A plataforma contempla os papéis de um cenário básico de mobilidade, i.e., criou-se uma rede (ver Figura 2) utilizando três estações fixas e uma móvel: um CN, um FA, um HA, e um MN, respectivamente. O CN mantém a comunicação com o MN, enquanto este se move da sua rede local (*Home Network, HN*) para uma outra rede (*Foreign Network, FN*). A comunicação entre o MN e os *routers* é suportada através de tecnologia 802.11b. As duas células representadas têm uma área de intersecção entre elas, pois o MN deverá ter sempre conectividade *wireless* em pelo menos uma das células.

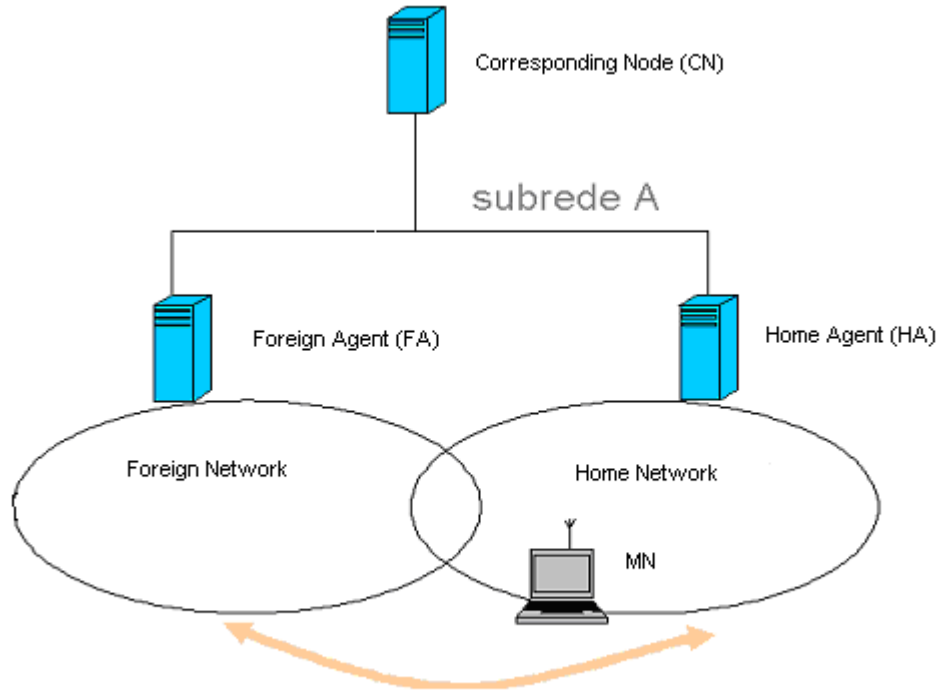


Figura 2: Estrutura da Plataforma de Testes.

O equipamento utilizado apresenta as seguintes características:

- **HA:** Celeron 1700MHz, 512 Mb RAM, 2 interfaces de rede;
- **FA:** Pentium MMX 266 Mhz, 92 Mb RAM, 2 interfaces de rede;
- **CN:** Pentium MMX 266 Mhz, 92 Mb RAM, 2 interfaces de rede;
- **MN:** Portátil Compaq AMD K6 475 Mhz, 192 Mb RAM, 1 interface de rede (Micronet 11Mbps PCMCIA RadioLink LAN Adapter);
- **HUB:** ARK AR 1009, 10 Mbps, 8 portas;
- **APs:** Micronet RadioLink 2/11Mbps Wireless LAN Access Point w/ Interbuilding Bridge, modelo número SP912;
- **Placa Wireless:** 11Mbps PCMCIA RadioLink LAN Adapter, Modelo Número SP905.

Adicionalmente, avaliaram-se na plataforma as placas apresentadas na Tabela 3 .

Tabela 3: Placas Wireless Testadas

VENDEDOR	TIPO	ID	I/F	CHIPSET
Zoom Telephonics	802.11b	ZoomAir 4100	PCMCIA	Prism2/2.5/3
Micronet	802.11b	SP 905B	PCMCIA	Prism2/2.5/3
US Robotics	802.11b+ 22Mbps	USR2216	PCI	TI

As placas *US Robotics*, nomeadamente a USR812216 com a qual se trabalhou, possuem um *chipset* TI da *Texas Instruments*TM que se revelou como uma má opção para Linux, pois não é reconhecida. Portanto, obriga a configurações extra. O seu funcionamento também se revelou instável. Os *drivers* que utilizam 'acx_100' apresentam ainda algumas limitações.

Em geral, as redes locais sem fios utilizam uma de três técnicas para transmissão de informação: transmissão em banda estreita, transmissão em espectro largo (*spread spectrum*) ou transmissão por infravermelhos. O modo utilizado nesta plataforma é o *spread spectrum*. Neste modo são utilizadas larguras de banda entre os 2.400 GHz e os 2.483 GHz, que não necessitam de autorização prévia das entidades reguladoras do espectro radioelétrico.

A configuração dos APs foi conseguida com recurso à aplicação Micronet (*Access Point Utility*) disponível com o software de instalação do modelo SP912. Para ambos os APs foram definidos diferentes ESSIDs e canais (channels), de modo a evitar conflitos de frequência rádio.

Atribuiu-se à rede estrangeira o ESSID "MIPL_foreign", no canal 6, a 2437 Mhz. Na rede local, atribuiu-se o ESSID "MIPL_local", que utiliza o canal 1 a 2412 Mhz.

Assim que todas as máquinas da plataforma estejam devidamente configuradas, pode-se começar a efectuar testes de mobilidade na plataforma. Quando o MN se encontra na sua rede local, funciona normalmente enviando e recebendo pacotes. Quando se move para uma rede estrangeira, necessita de adquirir um novo endereço IPv6 (care-of-address) através dos pacotes de autoconfiguração enviados pelo *radvd* nos *routers*. O MNI envia ao seu HA um *Binding Registration* para o informar da sua nova localização. Este pacote é recebido pelo HA que irá responder com um pacote de *Binding Acknowledgement*.

O HA irá interceptar os pacotes com destino ao MN encaminhando-os para o endereço temporário, e recorrendo a encapsulamento IPv6.

De seguida, apresentamos a configuração a efectuar a nível de software, utilizando dois pacotes de mobilidade: o MIPL, para Linux, e o KAME/MIP6, para BSD.

5.4.1. Linux, MIPL

Nos diferentes elementos móveis da bancada de testes foi instalado o RedHat 9.0, kernel 2.4.20. Para o PC que desempenha funções de HA e para o MN (computador portátil), foi instalado adicionalmente a última versão do software MIPL, versão 0.9.5.1-v24.20. A instalação requere os seguintes passos:

- 1 descompactar o kernel e mipv6: `tar zxvf linux-2.4.20.tar.gz, tar xvfz mipv6-0.9.5.1-v2.4.20.tar.gz;`
- 2 copiar para a directoria do novo kernel o patch do mipv6-0.9.5.1-v2.4.20, e executar o patch com o seguinte comando: `patch -p1 -dry-run < mipv6-0.9.5.1-v2.4.20.patch`. Caso ocorram *hunks* que falhem a execução, a instalação não pode prosseguir;
- 3 concluído o teste descrito em 2, aplicar o *patch* para alterar o código fonte do *kernel*, e assim suportar MIPv6: `patch -p1 < mipv6-0.9.5.1-v2.4.20.patch;`
- 4 configurar o kernel: `make menuconfig`. As opções fornecidas pelo patch do MIPL inclui variáveis adicionais para a configuração do módulo. Estas variáveis estão presentes na secção *network options* e são incluídas na configuração do *kernel*:

```
CONFIG_IPV6_SUBTREES – suporte experimental de source address subtree
CONFIG_IPV6_IPV6_TUNNEL – tunnelling experimental IPv6-IPv6
CONFIG_IPV6_MOBILITY – suporte experimental para a especificação do IPv6 móvel
CONFIG_IPV6_MOBILITY_HA – se for pretendido um router que desempenhe o papel de Agente Local
CONFIG_IPV6_MOBILITY_MN – para posto móvel
CONFIG_IPV6_MOBILITY_DEBUG – para aceder a mensagens de debugging durante a execução do MIPL
```

Os valores desta variável podem ser Y (incluir no *kernel*), N (não incluir no *kernel*) e M (incluir como módulo do *kernel*). Para poder executar o MIPL deverão ser seleccionadas as seguintes variáveis:

```
Code Maturity
CONFIG_EXPERIMENTAL=Y

General Setup
CONFIG_SYSCTL=Y
CONFIG_NET=Y

File Systems
CONFIG_PROC_FS=Y

Modules Support
CONFIG_MODULES=Y

Networking
CONFIG_NETFILTER=Y
CONFIG_UNIX=Y
CONFIG_INET=Y
CONFIG_IPV6 = M
CONFIG_IPV6_SUBTREES = Y
CONFIG_IPV6_IPV6_TUNNEL=M
CONFIG_IPV6_MOBILITY=M
CONFIG_IPV6_MOBILITY_HA=Y(no caso do router) ou N (no portátil)
CONFIG_IPV6_MOBILITY_MN=Y(no caso do portátil) ou N (no router)
CONFIG_IPV6_MOBILITY_DEBUG=Y
```

- 5 compilar o kernel do modo habitual: `make dep; make bzImage; make modules; make modules_install`. Após execução com sucesso, o *kernel* encontra-se preparado para trabalhar com o IPv6 e com o IPv6 Móvel;
- 6 copiar o novo *kernel* para a directoria `/boot` que habitualmente contém as imagens de *kernel*: `cp arch/i386/boot/bzImage /boot/vmlinuzMIPv6;`
- 7 criar uma nova imagem `initrd`: `/sbin/mkinitrd initrdMIPv6.img 2.4.20;`
- 8 alterar o ficheiro `/etc/lilo.conf`, de modo a contemplar o arranque do novo *kernel*:

....

```
image=/boot/vmlinuzMIPv6
label = MobileIpv6
initrd=/boot/initrdMIPv6.img
read-only
append="root=LABEL=/"
...
```

9 *actualizar o lilo:* /sbin/lilo;

10 executar o seguinte comando, para concluir a configuração do MIPv6: `mknod /dev/mip6_dev c 0xf9 0;`

11 Re-iniciar o PC.

Depois de reiniciado o PC com o novo *kernel*, compila-se e instala-se as ferramentas *userspace* do MIPL do seguinte modo:

```
host: / # cd /usr/local/mip6-0.9.5.1-v2.4.20/
host: /usr/local/mip6-0.9.5.1-v2.4.20# ./configure
host: /usr/local/mip6-0.9.5.1-v2.4.20# make
host: /usr/local/mip6-0.9.5.1-v2.4.20# make install
```

Adicionalmente, existem alguns ficheiros de configuração essenciais para o bom funcionamento do MIPL, que passamos a explicar.

5.4.1.1. Ficheiros de configuração e aplicações MIPL

Nesta secção, descrevem-se os ficheiros de configuração (incluindo opções) e ainda ferramentas pertencentes ao MIPL, que possam ajudar ao diagnóstico de vários problemas.

5.4.1.1.1. *network-mip6.conf*

O *network-mip6.conf* é o principal ficheiro para a configuração do MIPL. Este ficheiro é utilizado pelo *script* de *startup* do MIPL, para carregar os parâmetros no módulo do *kernel* e apresenta as seguintes opções:

- **FUNCTIONALITY** – permite seleccionar o tipo de estação. Esta pode actuar como um agente local (HA), posto móvel (MN) ou nó correspondente (CN). Tanto o agente local ou posto móvel têm as funcionalidades de nó correspondente. por omissão o valor é CN.
- **DEBUGLEVEL** – em situações de erro, pode ser desejável obter informação detalhada sobre a execução do MIPL. O aumento deste valor incrementa o número de mensagens específicas de *debugging*. Por omissão o valor é 0.
- **TUNNEL_SITELOCAL** – indica se as mensagens *unicast* são encapsuladas para o posto móvel quando este não se encontra na rede local. por omissão o valor é YES
- **MIN_TUNNEL_NR** – número mínimo de *tunnels* livres na *cache* do posto móvel ou agente local (valor mínimo=1). Para assegurar que os *bindings* são executados correctamente mesmo quando a carga na rede seja elevada, o número desta opção deverá ser elevado no agente local.
- **MAX_TUNNEL_NR** - número máximo de *tunnels* livres na *cache* do posto móvel ou agente local (valor mínimo=1). Para uma boa performance, o valor deverá ser superior ao **MIN_TUNNEL_NR**.
- **HOMEADDRESS** – endereço local com prefixo do posto móvel.
- **HOMEAGENT** – endereço do agente local do posto móvel.

- HOMEDEVICE – o endereço local é atribuído a uma interface de rede. Esta opção especifica qual a interface utilizada. Exemplo: eth0.

5.4.1.1.2. mipdiag

O *mipdiag* é uma ferramenta de diagnóstico e configuração do MIPL. É possível obter através destas estatísticas, informação do estado e alterar parâmetros em *runtime*. Apresenta as seguintes opções gerais:

- `-?,--help:` devolve informação de ajuda;
- `-V,--version:` devolve a versão do *mipdiag* e MIPL;
- `-c,--bcache:` imprime as linhas da *cache* de *binding*;
- `-d,--debuglevel:` [inteiro] define o nível de *debug* desde 0 até 7
- `-s,--statistics:` imprime estatísticas.

Adicionalmente, contém ainda opções a aplicar ao HA:

- `-t,--tunnel [yes|no]:` indica se as mensagens *unicast* são encapsuladas para o posto móvel quando este não se encontra na rede local.

Do mesmo modo, as opções passíveis de utilização para o MN são:

- `-h,--homeaddress endereço-ipv6/tamanho-prefixo:` define o endereço local do posto móvel. Deverá ser usado `-H` e `-i` com esta opção. Podem ser adicionados novos endereços locais no posto móvel, com a limitação que dois endereços locais não podem partilhar o mesmo agente local.
- `-H,--homeagent [endereço-ipv6/tamanho-prefixo]:` define o endereço do agente local para o posto móvel. Deverá ser usado `-h` e `-i` com esta opção. Se o endereço e o prefixo são omitidos, o DHADD é iniciado.
- `-m,--mminfo:` imprime a informação do posto móvel.
- `-l,--bulist:` imprime a lista de *binding update*.
- `-I,--ifaces:` devolve as interfaces disponíveis e as suas preferências.
- `-i,--if_name ifname:` define o nome da interface. Deverá ser usado em conjunto com `-H`, `-h` e `-P`.
- `-P,--set-if-pref preference:` define a preferência para uma interface. O posto móvel utiliza por omissão a interface com maior preferência. Se a interface actual é perdida, o posto móvel muda para interface seguinte com maior preferência e efectua um *handoff* vertical. A preferência por omissão é estabelecida pelo identificador de uma interface. Para os casos em que o posto móvel possa ter vários agentes locais, deverá ser dada maior preferência à interface local para evitar problemas com o proxy DAD. Caso contrário, o posto móvel irá tentar registar o seu endereço local no agente com a interface com a preferência mais alta.

5.4.1.2. Outras Aplicações

Nesta secção, descrevem-se os processos de instalação e de configuração de aplicações úteis em ambientes *wireless*.

5.4.1.2.1. RADVD

O *radvd* é um daemon baseado na especificação [ND]. O *daemon* escuta *Router Solicitations* (RS) e envia *Router Advertisements* (RA), mensagens estas que permitem às estações

configurar os seus endereços e outros parâmetros automaticamente, e seleccionar um *router* por omissão baseado nas informações dessas notificações. O *radvd* necessita que o *forwarding* nos *routers* esteja activo. O *forwarding* pode ser activado através de *sysctl* alterando para "1" a variável de sistema *net.ipv6.conf.all.forwarding*.

Para instalar o *daemon*, deve-se obter o código fonte [RADVD], copiá-lo para, por exemplo, a directoria */usr/local*, descompactá-lo e seguir as instruções fornecidas para o compilar. A configuração do *daemon* RADVD é efectuada através do ficheiro */etc/radvd.conf*. Este ficheiro descreve a informação incluída nos RA para os diversos interfaces:

```
Nome interface {
  Lista das opções específicas da interface (interface specific options)
  Lista das definições do prefixo (prefix definitions)
};
```

O Prefixo poderá ser o da rede ou do endereço da interface.

Das várias opções disponíveis para a definição de interface (INTERFACE SPECIFIC OPTIONS) ou do prefixo (PREFIX SPECIFIC OPTIONS), descrevemos neste documento apenas as mais relevantes, começando pelas opções a aplicar ao Interface (interface specific options):

- **AdvSendAdvert on|off:** indica se o router envia ou não RA periodicamente, ou se responde a RS. Por omissão, o seu valor é off.
- **MaxRtrAdvInterval (segundos):** tempo máximo em segundos permitido entre o envio pela interface de *multicast RA*. Este valor deverá ser superior a 4 segundos e inferior a 1800 segundos. Para as extensões de IPv6 móvel, o seu valor deverá ser o mínimo: 1.5 segundos. Valor por omissão: 600 segundos.
- **MinRtrAdvInterval (segundos):** tempo mínimo em segundos permitido entre o envio pela interface de *multicast RA*. Deverá ser superior a 3 segundos e inferior a $0.75 * \text{MaxRtvInterval}$. Para as extensões de IPv6 Móvel, o seu valor deverá ser o mínimo: 0.05 segundos. Valor por omissão: $0.33 * \text{MaxRtvAdvInterval}$.
- **AdvHomeAgentFlag on|off:** quando seleccionado, indica que o *router* emissor de RA desempenha também as tarefas de Agente Local. Quando seleccionado, os limites mínimos especificados para o MIPv6 são usados nas variáveis *MinRtrAdvInterval* e *MaxRtrAdvInterval*. Por omissão, o seu valor é *off*.
- **HomeAgentLifetime (segundos):** o tempo, especificado em segundos, em que o *router* oferece serviços de Agente Local. O valor 0 não deverá ser utilizado. O *lifetime máximo* é de 65520 segundos (18.2 horas). Esta opção é ignorada se o *AdvHomeAgentInfo* não estiver seleccionado. Se tanto o *HomeAgentLifetime* e *HomeAgentPreference* estão configurados com os seus valores por omissão, a informação de *Home Agent Option* não irá ser enviada. Valor por omissão: *AdvDefaultLifetime*.
- **HomeAgentPreference:** indica o nível de preferência para o Agente Local que envie estes *Router Advertisements*. Valores superiores a 0 indicam maiores preferências, valores inferior a 0 indicam menores preferências. Esta opção é ignorada se o *AdvHomeAgentInfo* não está seleccionado. Se tanto o *HomeAgentLifetime* e *HomeAgentPreference* estão configurados com os seus valores por omissão, a informação de *Home Agent Option* não irá ser enviada. Valor por omissão: 0.
- **AdvIntervalOpt on|off:** quando seleccionado, a opção *Advertisement Interval* é incluída nos *Router Advertisements*. Se seleccionado, os limites mínimos definidos pelo IPv6 Móvel são usados para o *MinRtvAdvInterval* e *MaxRtvAdvInterval*.

As opções a aplicar aos prefixos de rede (Prefix Specific Options) são:

- **AdvOnLink on|off:** quando seleccionado indica que o prefixo pode ser usado para informação *on-link*. Quando não está activo, os RA não distinguem entre as propriedades *on-link* ou *off-link* do prefixo. Valor por omissão: *on*.
- **AdvAutonomous on|off:** quando activo indica que o prefixo poderá ser utilizado para *autonomous address configuration*, como definido no RFC 2462. Valor por omissão: *on*.
- **AdvRouterAddr on|off:** quando está activo, significa que é enviado o endereço da interface em vez do prefixo de rede, no caso do IPv6 móvel. Se seleccionado, os limites mínimos definidos pelo IPv6 móvel são usados para o *MinRtvAdvInterval* e *MaxRtvAdvInterval*. Valor por omissão: *off*.

5.4.1.2.2. RADVD

O *radvdump* imprime o conteúdo dos RA enviados pelo *radvd*. Possui uma opção “-d” onde se pode definir o nível de *debugging*, através da selecção de um inteiro que poderá tomar valores compreendidos entre 1 e 4 (*quiet* e *very verbose*). O valor por omissão é 0.

5.4.1.2.3. Wireless Tools Package

O *Wireless Extension* é um API genérico que fornece estatísticas e a possibilidade de configuração de interfaces *Wireless LAN*. Esta ferramenta suporta as diferentes tecnologias *Wireless LANs*, independentemente do seu tipo, e permite ainda, alterar automaticamente as aplicações. A *package* inclui as seguintes ferramentas:

- **lwconfig:** manipulação dos parâmetros básicos de *wireless* (*ssid*, *freq*, *channel*, *mode*, *rate*, etc);
- **lwspy:** monitoriza a qualidade de sinal para os APs;
- **lwlist:** lista, entre outros, os APs existentes na área e as suas frequências;
- **lwpriv:** permite a manipulação das *Wireless Extensions* específicas de um driver.

O *software* pode ser obtido de [\[pcmciaacs\]](#), e ser instalado em qualquer directoria. Sugerimos a directoria */usr/local*, devendo ser descompactado e instalado de acordo com as instalações fornecidas.

5.4.1.2.4. WAVEMON

O *wavemon* é um monitor para dispositivos *wireless*. Esta ferramenta é baseada na biblioteca *n-curses* e permite visualizar em modo gráfico (tempo real) em níveis de sinal, estatísticas de pacotes, configuração dos dispositivos e parâmetros de rede. Para instalar, deve-se obter o código fonte [\[WAVEMON\]](#). Deve de seguida ser instalado como normalmente, de acordo com as informações fornecidas.

5.4.1.2.5. TCPDUMP

O *tcpdump* é das ferramentas mais populares para a monitorização de pacotes em interfaces IP. É bastante útil para verificar se a configuração de *routers* ou máquinas se encontra correctamente definida, dado que analisa o encaminhamento de tráfego nas máquinas onde está em execução. Esta ferramenta tem uma dependência com o *libpcap*, sendo necessário instalá-lo antes do *tcpdump*. Os ficheiros requeridos podem ser obtidos em [\[TCPDUMP\]](#).

5.4.1.3. Scripts Adicionais

Para permitir a configuração automática dos diversos elementos, foram criados scripts. Para a configuração de endereçamento, *encaminhamento* e *forwarding* foi elaborado um *script* específico para cada máquina. O *script* é executado aquando do arranque da máquina. A configuração do radvd encontra-se no ficheiro */usr/local/etc/radvd.conf*. Para o MIPL, a configuração é efectuada no ficheiro */etc/sysconfig/network-mip6.conf*.

5.4.1.3.1. Corresponding Node

```
#script NCipv6

#arranque do suporte IPv6
modprobe ipv6

#arranque da interface eth1
ifconfig eth1 up

#eliminar os endereços e a tabela de routing anterior
ip -6 addr flush dev eth1
ip -6 route flush dev eth1

#atribuir um endereço à maquina
ip -6 addr add fec0::1:1:1:1:1/64 dev eth1

#estabelecer o routing para cada uma das subredes
ip -6 route add fec0:0:0:2::/64 via fec0::1:1:1:1:2 dev eth1
ip -6 route add fec0:0:0:3::/64 via fec0::1:1:1:1:3 dev eth1
```

5.4.1.3.2. Home Agent

```
#script R1ipv6

#arranque do suporte IPv6
modprobe ipv6

#arranque da interface eth0 e eth1
ifconfig eth0 up
ifconfig eth1 up

#activar o forwarding no router
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding

#atribuir os endereços às duas interfaces de rede no router
ip -6 addr add fec0::1:1:1:1:2/64 dev eth0
ip -6 addr add fec0::2:1:1:1:1/64 dev eth1

#configuração do routing para cada uma das subredes
ip -6 route add fec0:0:0:3::/64 via fec0::1:1:1:1:3 dev eth0

#inicio do daemon radvd
radvd

#inicio do radvdump
radvdump -d 4

#Ficheiro de configuração radvd.conf

interface eth1
{
```

```

AdvSendAdvert on;
MinRtrAdvInterval 0.5;
MaxRtrAdvInterval 1.5;
AdvIntervalOpt on;

prefix fec0:0:0:2::/64
{
    AdvOnLink on;
    AdvAutonomous on;
};
};

```

```

#output do radvdump

Router advertisement from fe80::2a0:c9ff:fe45:24e4 (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 fec0:0:0:2::/64
    AdvValidLifetime: 2592000
    AdvPreferredLifetime: 604800
    AdvOnLink: on
    AdvAutonomous: on
    AdvRouterAddr: off
AdvSourceLLAddress: 00 A0 C9 45 24 E4
AdvIntervalOpt:
    AdvInterval: 1

```

5.4.1.3.3. Foreign Agent

```

#script R2ipv6

#arranque do suporte IPv6
modprobe ipv6
#arranque das duas interfaces de rede no router
ifconfig eth0 up
ifconfig eth1 up

#activação do forwarding no router
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding

#atribuição de endereços IPv6 às interfaces de rede e ao agente local
ip -6 addr add fec0::1:1:1:3/64 dev eth0
ip -6 addr add fec0::3:1:1:1/64 dev eth1
ip -6 addr add fec0::3:1:1:2/64 dev eth1

#configuração do routing para cada uma das subredes
ip -6 route add fec0:0:0:2::/64 via fec0::1:1:1:2 dev eth0

#inicialização do MIPL
/etc/init.d/mobile-ip6 restart

#inicio do daemon radvd
radvd

#inicio do radvdump
radvdump -d 4

```

```

#Ficheiro de configuração radvd.conf

```

```

interface eth1
{
  AdvSendAdvert on;
  MaxRtrAdvInterval 1.5;
  MinRtrAdvInterval 0.5;
  AdvHomeAgentFlag on;
  AdvHomeAgentInfo on;
  #HomeAgentLifeTime 60000;
  AdvIntervalOpt on;
  prefix fec0:0:0:3::2/64
  {
    AdvOnLink on;
    AdvAutonomous on;
    AdvRouterAddr on;
  };
};

```

```
#output do radvdump
```

```

Router advertisement from fe80::204:75ff:fede:ae59 (hoplimit 255)
Received by interface eth1
# Note: {Min,Max}RtrAdvInterval cannot be obtained with radvdump
AdvCurHopLimit: 64
AdvManagedFlag: off
AdvOtherConfigFlag: off
AdvHomeAgentFlag: on
AdvReachableTime: 0
AdvRetransTimer: 0
Prefix fec0:0:0:3::2/64
  AdvValidLifetime: 2592000
  AdvPreferredLifetime: 604800
  AdvOnLink: on
  AdvAutonomous: on
  AdvRouterAddr: on
AdvSourceLLAddress: 00 04 75 DE AE 59
AdvIntervalOpt:
  AdvInterval: 1

```

```
# MIPL Mobile IPv6 Configuration file
```

```

# Should this node act as a home agent (ha), mobile node (mn) or
# correspondent node (cn). HA and MN both have CN functionality
# embedded. Default: cn.
FUNCTIONALITY=ha

```

```

# In error situations it may be desired to get more detailed
# information what is happening. Increase this value to get more
# messages from the module (default: 0).
DEBUGLEVEL=7

```

```

# Should unicasts to node's site-local address be tunneled when mobile
# node is not in its home network (default: yes).
#TUNNEL_SITELOCAL=yes

```

```

# Minimum number of free tunnel devices kept in cache on MN or HA
# Must be set to at least 1 for MN and HA. To ensure successful
# bindings even during high work loads it could be set to a bigger
# value on the HA.
MIN_TUNNEL_NR=1

```

```

# Maximum number of free tunnel devices kept in cache on MN or HA
# Must be set to at least 1 for MN and HA. To improve performance
# set it higher than MIN_TUNNEL_NR
MAX_TUNNEL_NR=3

```

```
# Device where home address should be assigned to
HOMEDEV=eth1

# Home agent's address for mobile node with prefix length.
HOMEAGENT=fec0::3:1:1:1:2/64
```

5.4.1.3.4. Mobile Node

```
#script Pmipv6
# Neste script não é adicionado o endereço. O MIPL encarrega-se de definir o endereçamento e o # routing

#arranque do suporte IPv6
modprobe ipv6

#inicialização da interface de rede eth0
ifconfig eth0 up

#inicialização da placa wireless
/etc/init.d/pcmcia restart

sleep 5

#indicar ao posto móvel que no inicio deve-se ligar à sua rede local
iwconfig eth0 essid MIPL_local

#inicialização do MIPL
/etc/init.d/mobile-ip6 restart
```

```
# MIPL Mobile IPv6 Configuration file

# Should this node act as a home agent (ha), mobile node (mn) or
# correspondent node (cn). HA and MN both have CN functionality
# embedded. Default: cn.
FUNCTIONALITY=mn

# In error situations it may be desired to get more detailed
# information what is happening. Increase this value to get more
# messages from the module (default: 0).
DEBUGLEVEL=7

# Should unicasts to node's site-local address be tunneled when mobile
# node is not in its home network (default: yes).
#TUNNEL_SITELOCAL=yes

# Minimum number of free tunnel devices kept in cache on MN or HA
# Must be set to at least 1 for MN and HA. To ensure successful
# bindings even during high work loads it could be set to a bigger
# value on the HA.
MIN_TUNNEL_NR=1

# Maximum number of free tunnel devices kept in cache on MN or HA
# Must be set to at least 1 for MN and HA. To improve performance
# set it higher than MIN_TUNNEL_NR
MAX_TUNNEL_NR=3

# Device where home address should be assigned to
HOMEDEV=eth0

# Home address for mobile node with prefix length. Example:
# 3FFE:2620:6:1234:ABCD::1/48 (Don't use the example value!)
HOMEADDRESS=fec0::3:1:1:1:3/64
```

```
# Home agent's address for mobile node with prefix length.
HOMEAGENT=fec0::3:1:1:1:2/64
```

5.4.2. KAME/MIP6

Na pilha KAME (ver secção 5.1.4), o MIPv6 não se encontra, por omissão, activo. É necessário re-configurar o *kernel* e re-compilá-lo. Alguns comandos necessitam também de re-compilação. O ficheiro de configuração do kernel (.config) deve incluir as seguintes opções:

```
options MIP6
options MIP6_DEBUG
options MIP6_ALLOW_COA_FALLBACK
# options MIP6_DRAFT13
pseudo-device hif 1
```

Com a opção MIP6_DEBUG, o *kernel* irá devolver todas as mensagens de compilação. Estas mensagens podem ainda ser activadas e desactivadas durante a execução, utilizando para o efeito o programa *mip6control*.

MIP6_ALLOW_COA_FALLBACK activa a capacidade de recuperação/recomposição de dados (*fallback*) no CoA.

Na especificação do MIPv6, todos os nós IPv6 devem suportar a opção *Home Address Destination*. Caso a ligação não reconheça esta opção, o MN não pode comunicar com esse dispositivo.

Se o MIP6_ALLOW_COA_FALLBACK estiver especificado, o *kernel* irá tentar utilizar o seu HA como endereço de origem sem a opção *Home Address Destination*. Se esta aproximação falhar, o *kernel* irá utilizar o CoA como o seu endereço origem na próxima vez.

5.4.2.1. Aplicações

O KAME fornece três comandos básicos a aplicar ao MIP6: *rtadvd*, *mip6control* e *had*. Para utilizar HAs, é necessário re-compilar o comando *rtadvd* com a opção MIP6 activada. Adicionalmente, tem de se re-compilar o *had*.

Uma outra aplicação essencial é o *mip6control*, que permite controlar as funções do KAME MIP6. Para compilar o *rtadvd*, deve-se, na directoria do *rtadvd* ($\${KAME}/\text{freebsd4}/\text{sbin}/\text{rtadvd}/$ para FreeBSD), adiciona-se (numa nova linha) a opção: CFLAGS+=-DMIP6. De seguida, re-compilar e instalar normalmente.

Do mesmo modo, o pacote *had* não se encontra instalado por omissão. Portanto é necessário, na directoria $\${KAME}/\text{kame}/\text{kame}/\text{had}/$ é necessário proceder à sua instalação.

5.4.2.2. Configuração de um Home Agent

Para configurar um *Home Agent*, é necessário proceder aos seguintes passos:

- designar um endereço Home Agent Subnet Anycast;
- preparar o *rtadvd.conf* para o *Home Agent* ;
- executar o *rtadvd* com a opção *-m*, e executar o *had* .

Adicionalmente, pode-se atribuir ao HA um endereço *Home Agent Subnet Anycast*, que possibilita a realização de DHAAD (*Dynamic Home Agent Address Discovery*). O endereço *anycast* é calculado da seguinte forma: se possuir um prefixo de 64 bits, concatena-se o endereço do seu prefixo e 0xfdfffffffffffffe. Se o prefixo não for de 64 bits, preenche a parte de bits da estação com os bits apropriados, e com o valor de 0xfffffffffffffffffffffe:

- `# ifconfig fxp0 inet6 2001:200:1:2::fdff:ffff:ffff:ffffe anycast alias`

Nota: este endereço tem de ser configurado antes da execução do *had*.

Para gerar as mensagens RA, basta invocar o *rtadvd* com a opção *-m* (de mobilidade):

- `# /usr/local/v6/sbin/rtadvd -m fxp0`

De seguida, é necessário invocar o *had* com o nome do interface onde se pretende activar o DHADD:

- `# /usr/local/v6/sbin/had fxp0`

Após estes passos, pode-se invocar o HA, através do comando *mip6control -g*.

5.4.2.3. Activar o MN

O primeiro passo a executar para activar o papel de MN, é especificar o prefixo da *HN*, através do comando abaixo exemplificado, onde o prefixo indicado deve ser substituído pelo prefixo IPv6 adequado:

- `# mip6control -H2001:200:1:1:: -P64`

De seguida, activar as funções de mobilidade com o comando *mip6control*:

- `# mip6control -m`

Para detectar movimento, um MN necessita de receber pacotes de *Router Advertisement*. Este processo pode ser desencadeado através do comando *rtsol* com as opções *-a -m*, por forma a que o dispositivo detecte rapidamente a sua localização.

5.4.2.4. Configuração de características de segurança

O KAME/MIP6 pode proteger o acto de registo dos CoAs (*Binding Update/Binding*). Por omissão, o KAME/MIP6 utiliza a sub-opção de autenticação definida em [\[KAME1\]](#). Para esta protecção é necessário estabelecer uma *associação de segurança* IPsec entre dispositivos. Portanto, é necessário recorrer ao programa *setkey* para configurar as associações de segurança. Por exemplo, se for necessário proteger o acto de *Binding Update* entre o MN que apresenta o endereço A e o HA cujo endereço é B, configura-se a associação de segurança da seguinte forma:

- `add A B ah 1500 -m transport -A hmac-sha1 "AH SA configuration!";`
- `add B A ah 1600 -m transport -A hmac-sha1 "AH SA configuration!";`

É ainda necessário configurar a política de segurança:

- `spdadd ::0[any] ::0[any] ipv6-opts -P out ipsec ah/transport//require;`

A segurança pode ser inabilitada recorrendo à aplicação *mip6control*. Para incapacitar a autenticação de dados, digite o seguinte:

- `# mip6control -T 0`

6. IPv6 Aplicado a Tecnologias Celulares

Na secção anterior, descrevemos cenários e configurações de IPv6 a aplicar em WLANs. No entanto, o panorama de mobilidade em IPv6 abrange ainda a utilização de IPv6 com tecnologia celular.

Os operadores que baseiam as suas redes em tecnologias celulares tais como *General Packet Radio Service* (GPRS) ou *Universal Mobile Telecommunications System* (UMTS), têm vindo a

apresentar um interesse crescente na utilização de serviços IP e consequentemente, de serviços disponíveis na Internet. No entanto, as diversas limitações do protocolo IP, em particular a limitação relacionada com o seu espaço de endereçamento, fazem com que a sua aplicação em

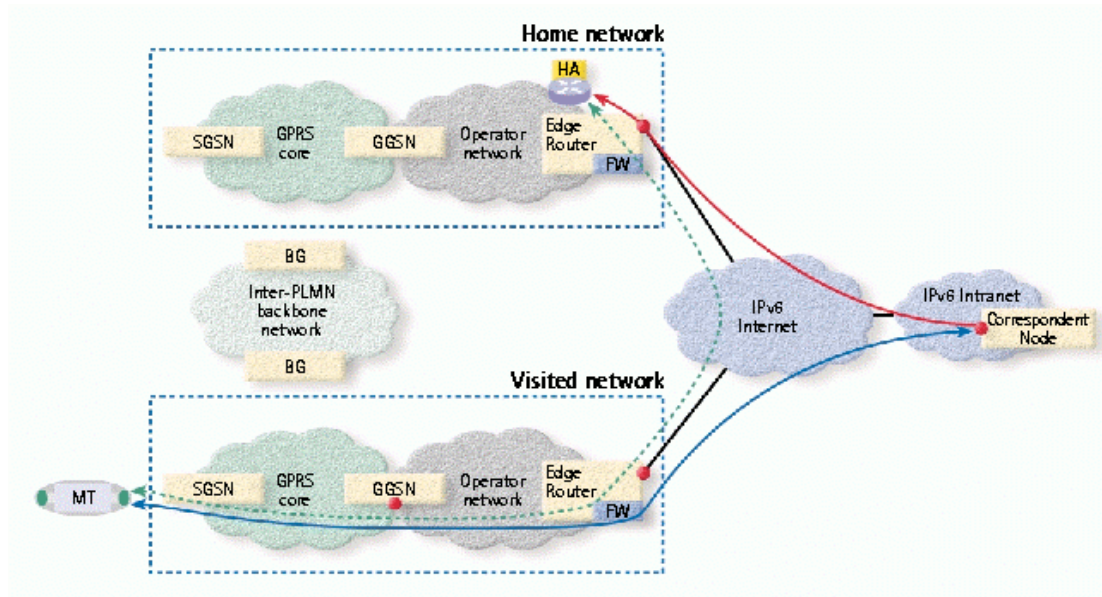


Figura 3: Cenário GPRS IPv6.

redes celulares seja praticamente impossível. Em contraste, o IPv6 é um forte candidato para a integração da tecnologia celular em redes IP.

O vasto espaço de endereçamento do IPv6 é uma característica essencial para a utilização de dispositivos com várias interfaces de rede (*Multiple Access Devices*), ou para convergência de diferentes redes (casa, trabalho, etc.). Adicionalmente, o mecanismo de autoconfiguração IPv6, e a ausência de mecanismos de NATs são dois benefícios para a tecnologia celular, dado que permitem mobilidade e *roaming* a uma escala global. O IETF tem vindo a impulsionar a criação de directivas e mecanismos para permitir a utilização de IPv6 sobre tecnologia celular [RFC3316]. O suporte para IPv6 é actualmente obrigatório a partir do 3GPP release 99, dado que IPv6 é especificado como a única versão do IP capaz de suportar *IP Multimedia System* (IMS). Os elementos mais importantes da arquitectura são o *Mobile Terminal* (MT) e a *Gateway GPRS Support Node* (GGSN), consoante ilustrado na Figura 3⁴.

O MT representa, por exemplo, um telemóvel, que pode ser um dispositivo integrado (com funções de GPRS e IPv6) ou apenas um dispositivo com 2 interfaces separados, um GPRS, outro IPv6, e.g., um computador portátil com placa GPRS e adicionalmente, interface IPv6.

O nodo GGSN tem funções de gestão de mobilidade GPRS e representa ainda o default router para o MT. Entre o MT e o GGSN existe uma ligação dedicada, denominada *Packet Data Protocol (PDP) Context*, criada através de um processo de activação, durante o qual o MT recebe o seu endereço IP e informação adicional necessária ao acesso IP, e.g., nome (DNS). Actualmente, existem 3 tipos de PDP Context: IPv4, IPv6, e Point-to-Point Protocol (PPP).

Um MT pode manter simultaneamente mais do que um *PDP Context*, para o mesmo nodo GGSN, ou para vários. Adicionalmente, os diferentes PDP Context podem ser (ou não) do mesmo tipo.

⁴Figura Original de [NOK1].

A arquitectura IMS é um componente especificado no 3GPP Release 5, para suportar serviços multimedia através de uma infraestrutura SIP. É composta de um conjunto de vários proxies, servidores e registos SIP. Contém ainda *Media Gateways (MGWs)*, elementos capazes de fornecer suporte a redes não IP (e.g., PSTN).

Os MT que suportem IMS utilizam a rede GPRS como uma rede de acesso a IMS. Tal significa que o MT tem de estabelecer um *PDP Context* antes de conseguir utilizar o sistema IMS.

A arquitectura IMS utiliza exclusivamente IPv6, o que significa que o *PDP Context* utilizado é obrigatoriamente do tipo IPv6. Como o terminal móvel utiliza exclusivamente IPv6 para aceder ao IMS, e ainda, como o servidor e proxy SIP IMS suportam exclusivamente IPv6, todo o tráfego IMS é IPv6, mesmo que o MT seja dual, por exemplo., suporte IPv6 e IPv4.

No entanto, e embora existam já especificações onde a utilização de IPv6 é obrigatória, o desenvolvimento de mecanismos IPv6 a nível de tecnologia celular encontra-se em fase embrionária de desenvolvimento. Tal deve-se principalmente ao facto de a utilização de serviços da Internet por parte de operadores celulares ser ainda muito recente, e de o mercado para tal aplicação ser ainda extremamente reduzido. Adicionalmente, existem implicações sérias devido ao facto de o próprio IPv6 não estar ainda solidamente desenvolvido na Internet:

- Redes 2.5G e 3G (Release 4) necessitam de converter o seu *core* para IPv6;
- Coexistência de IPv6 e IPv4 torna-se ainda mais relevante;
- IPv6 necessita de ser configurado não só no equipamento de core, mas também no equipamento terminal.

Adicionalmente, a tecnologia celular levanta os seguintes problemas:

- Privacidade por parte do utilizador (push services, static IP address vulnerability);
- problemas de migração entre redes wired e redes wireless, handoffs.

Na próxima secção, apresentamos o estado de equipamento e software capaz de permitir utilizar IPv6 sobre 2G/3G.

6.1. Suporte IPv6

Nesta secção apresentamos de um modo genérico equipamento e *software* com suporte IPv6. Limitamo-nos ao UE, dado que actualmente não existe equipamento de core com suporte IPv6. No entanto, apresentamos ainda indicações de como estabelecer ligação através de *túneis*. Cenários de transição são abordados na secção 7.

6.2. Symbian 7.0

Existe actualmente um sistema operativo, O Symbian v7.0 [\[SYMBIAN\]](#), que é uma plataforma open-source, com suporte para multimedia (MMS), texto (SMS) e email (POP/IMAP4/SMTP). O Symbian apresenta uma *pilha dupla* IPv4/IPv6, browsing com XML; Java; WAP. Tem ainda possibilidade de segurança e capacidade de utilização de Bluetooth ou IrDA. Dados sobre a corrente utilização do Symbian são:

- 1.23m telefones com Symbian enviados para quatro operadores no Japão e em território coberto por GSM/GRPS;
- 3.91 milhões de telemóveis com suporte Symbian distribuídos entre Janeiro-Setembro 2002;
- acordo de integração de tecnologia assinado com o operador NTT DoCoMo;

Dispositivos capazes de utilizar o Symbian são exemplos de UE com *pilha dupla*. Alguns dos dispositivos celulares que já integram o Symbian são:

- Nokia 7770 [\[NOKIA\]](#);
- Nokia 3600;
- Motorola A920;
- NTT DoCoMo F2102V
- Sony Ericsson P800
- Panasonic X700;
- Motorola A1000;
- Siemens SX1.

No entanto, e na prática, o *core* da rede continua a não suportar IPv6 nativo, sendo necessário utilizar mecanismos de transição para obter conectividade IPv6. Os cenários de transição são abordados na secção 7. Passamos de seguida a apresentar um cenário alternativo para testar IPv6 num ambiente GPRS.

6.3. Testes

Para testar IPv6/GPRS utilizando Linux, é necessário utilizar um PC e um telefone (ou interface) com capacidade GPRS. A configuração base em Linux pode ser obtida em [\[UK6X\]](#).

Adicionalmente, o operador UK6X [\[UK6X\]](#) apresenta uma configuração IPv6, onde é utilizado como UE um portátil ligado, através de IrDA a um telefone GPRS. Este funciona como um modem e a ligação é estabelecida através de túneis bidireccionais IPv6-em-IPv4 entre o PC e o *router* APN. O PC obtém conectividade IPv6 através da rede GPRS. A UK6X afirma ter efectuado testes com sucesso utilizando Linux e FreeBSD. O telemóvel utilizado foi um *Motorola Timeport*.

Além da configuração, a UK6X permite a qualquer utilizador efectuar testes através do seguinte modo:

- pedir um IP à 6UKX;
- pedir registo ao operador GPRS local para gprs.sixwinit.org APN. Este registo permite ao operador criar um túnel IPv6-em-IPv4 do *router* deles para o utilizador;
- configurar o dispositivo GPRS para utilizar o APN gprs.sixwinit.org;
- configurar o túnel IPv6-em-IPv4 no PC.

A configuração a colocar no dispositivo GPRS é:

```
APN: gprs.sixwinit.org
UserID:6WINIT
Password: XXXXXXXX
CID=2
```

Para activar PPP entre o PC e telefone deve-se efectuar o seguinte:

- instalar o pppd, adicionando o ficheiro [\[PPP1\]](#) e utilizar os scripts de configuração fornecidos [\[ppp2\]](#);
- iniciar a ligação PPP com o comando: `pppd call sixwinit`.

Do lado do computador, é necessário configurar o túnel, consoante apresentado no script:

```
#!/bin/sh
### Configuration ###
#Configure your Local IPv6 address below
REP="2001:618:5:2::?:/126"
### End Configuration ###
#tunnel configuration
ifconfig sit0 up
ifconfig sit0 tunnel ::192.168.1.6
ifconfig sit1 up
ifconfig sit1 add $REP
route -A inet6 add 0::0/0 gw ::192.168.1.6
```

Na próxima secção abordamos cenários de transição, incluindo exemplos de utilização com GPRS.

7. Cenários de Transição

Ao longo das secções anteriores foram apresentados cenários básicos de mobilidade IPv6. No entanto, os cenários apresentados diziam apenas respeito a IPv6, não tendo sido contemplados cenários onde o IPv6 tenha de coexistir com o IPv4, situação que representa a maioria dos cenários actuais.

Apesar dos protocolos IPv4 e IPv6 não serem conceptualmente diferentes, eles não são compatíveis entre si. A grande questão que se coloca ao novo protocolo está relacionada com a forma como se irá processar a integração de serviços IPv6 em redes IPv4, as quais podem incluir terminais IPv6 que necessitam de interagir com serviços IPv4. O futuro do IPv6 estará fortemente dependente da habilidade de o integrar nas redes IPv4 existentes sem que existam situações significativas que provoquem a inoperabilidade dos serviços existentes [Waddington 2002]. Se por um lado os utilizadores irão beneficiar das novas potencialidades introduzidas pelo IPv6, por outro, lado é importante que tenham a percepção que os serviços suportados pelo novo protocolo não são piores que os serviços suportados pelo seu antecessor.

Atendendo à importância do protocolo IP no funcionamento da Internet, o IETF criou um grupo de *ngtrans* [\[NGTRANS\]](#) cuja missão principal é estudar as melhores formas de integrar e de efectuar a transição entre os dois protocolos. Após vários anos, foram especificadas várias técnicas [\[RFC2893\]](#) que se destinam a ser usadas em sistemas terminais (*hosts*) e/ou *routers*, as quais são denominadas de mecanismos de transição.

Nesta secção, apresentamos os mecanismos de transição existentes, e exemplificamos a sua aplicação em alguns cenários que utilizam tecnologia GPRS.

7.1. Classificação dos Mecanismos de Transição

Os mecanismos de transição propostos pelo IETF podem ser classificados em três tipos:

- Pilha Dupla (*Dual Stack*);
- Mecanismos de Tradução (*Translation Mechanisms*);
- Mecanismos de Túnel (*Tunneling Mechanisms*).

Na Tabela 1 são apresentados os mecanismos de transição propostos pelo IETF considerados relevantes para serem usados nos cenários de transição IPv4-IPv6 em redes GPRS, indicando a

que tipo cada um pertence. Na coluna *Conectividade* é apresentado para cada um dos mecanismos de transição o tipo de conectividade que oferecem. Por exemplo, o mecanismo de transição pilha dupla oferece conectividade entre terminais IPv4 sobre redes IPv4 e entre terminais IPv6 sobre redes IPv6. A coluna *Localização* indica em que tipo de elemento de rede, terminais (T) e/ou nós intermédios (NI), o mecanismo é implementado.

Tabela 4: Mecanismos de Transição Aplicados a GPRS

Nome	Conectividade	Tipo	Localização
pilha dupla (<i>Dual Stack</i>)	IPv4 – IPv4 sobre IPv4 IPv6 – IPv6 sobre IPv6	dual stack	Terminais (T) ou Nós Intermédios (NI)
NAT-PT (Network Address Translator – Protocol Translator)	IPv4 – IPv6 ; IPv6 – IPv4	Tradutor	NI
TRT (Transport Relay Transport)	IPv6 – IPv4	Tradutor	NI
6to4	IPv6 – IPv6 sobre IPv4	Túnel	Entre dois NI
Túneis configurados	IPv4 – IPv4 sobre IPv6 IPv6 – IPv6 sobre IPv4	Túnel	Entre dois NI; Entre o NI e T; Entre dois T

7.1.1. Pilha Dupla (Dual Stack)

O mecanismo de transição *pilha dupla* implica, tal como o nome indica, a presença das duas pilhas protocolares, uma para cada versão do protocolo IP, na mesma interface de rede. As duas pilhas protocolares funcionam em paralelo, cabendo à aplicação a decisão de qual das duas é usada. Desta forma, os dispositivos de rede têm a capacidade de receber e de enviar pacotes em qualquer uma das versões do protocolo IP. Um dispositivo que suporte *pilha dupla* pode comunicar com nós que só suportem uma das versões do protocolo IP.

As aplicações que usam IPv4 continuam a funcionar como antes. No entanto, para que haja comunicação entre aplicações IPv4 e aplicações IPv6, é necessário que pelo menos uma delas suporte as duas versões do protocolo IP. Este mecanismo pode ser implementado tanto em sistemas terminais como em nós intermédios. O mecanismo de transição *pilha dupla* apresenta as seguintes desvantagens:

- a escassez dos endereços IPv4 é uma das principais motivações para o aparecimento do IPv6, pelo que não faz sentido o uso de mecanismos de transição que exijam a atribuição de endereços IPv4;
- com a presença de dois protocolos de rede, existirão no mínimo dois protocolos de encaminhamento com o aumento de complexidade que daí advém;
- este tipo de solução não estimula a transição para redes IPv6 nativas;
- não resolve problemas de interoperabilidade entre as aplicações que suportam versões diferentes do protocolo IP.

Este mecanismo tem como campo de aplicação as redes que são controladas por uma única organização. A tarefa de actualização é muito simples e não envolve, na maioria dos casos, despesas com novo *hardware*. Neste momento, a maioria dos sistemas operativos já dispõe de suporte para *pilha dupla*.

7.1.2. Mecanismos de Tradução

O problema da comunicação entre duas estações (ou duas aplicações) que suportem versões diferentes do protocolo IP não fica resolvido com o uso de *pilha dupla*, sendo necessário recorrer a um qualquer mecanismo de tradução. A tradução refere-se à conversão directa entre protocolos e pode ocorrer em diversas camadas da pilha protocolar. A operação de tradução pode ser realizada por equipamentos terminais ou por nodos intermédios.

Os mecanismos de tradução existentes são do tipo *Best-Effort*, ou seja, sempre que não exista uma equivalência directa entre os campos dos dois protocolos, a informação respeitante a esses campos perde-se durante a operação de tradução. Por exemplo, na conversão do cabeçalho de um pacote IPv6 para IPv4 perde-se a informação transportada em alguns cabeçalhos de extensão.

Os mecanismos de tradução podem ser divididos em dois tipos: *stateless* e *stateful*. Num mecanismo de tradução do tipo *stateless*, a operação de tradução não mantém nenhuma relação entre a tradução de diferentes pacotes pertencentes à mesma comunicação (por exemplo, à mesma sessão TCP). Assim, diferentes pacotes de uma sessão podem ser traduzidos por diferentes elementos de rede que implementem o mesmo mecanismo de tradução. No caso dos mecanismos de tradução *stateful*, todos os pacotes de uma sessão têm de ser traduzidos pelo mesmo elemento de rede que implementa o mecanismo de tradução.

7.1.2.1. SIIT - Stateless IP/ICMP Translation Algorithm

Não sendo por si só um mecanismo de transição, alguns mecanismos de tradução usam o algoritmo definido pelo SIIT [\[RFC2765\]](#) para traduzir os cabeçalhos dos pacotes IPv6/IPv4 em IPv4/IPv6, assim como as mensagens dos protocolos ICMPv6/ICMPv4 em ICMPv4/ICMPv6. O SIIT não pode ser usado para traduzir pacotes que usem endereços *Multicast*. Trata-se de um algoritmo de tradução do tipo *Best-Effort*. Durante a tradução de pacotes IPv6 para IPv4, o SIIT ignora os cabeçalhos de extensão do IPv6 dos tipos *Routing Header*, *Hop-by-Hop* e *Destination Options*, uma vez que não existe equivalência directa com nenhum dos campos do cabeçalho IPv4. Na tradução de IPv4 para IPv6, é ignorado o campo *Options* do cabeçalho IPv4.

O algoritmo não impõe nenhum mecanismo de atribuição de endereços IPv4 em endereços IPv6 (ou vice versa), nem nenhuma restrição ao modo como os pacotes são encaminhados de/para esse endereço. Estas operações são realizadas pelos mecanismos de transição que usam o SIIT.

No entanto, o SIIT propõe uma correspondência baseada no uso dos endereços IPv6 IPv4-*mapped* e dos IPv6 IPv4 traduzidos. Se forem usados estes dois tipos de endereços, quando um nó IPv6 pretende comunicar com um nó IPv4, o nó IPv6 é caracterizado por um endereço IPv6 IPv4 traduzido (::ffff:0:V4ADDR), enquanto que ao nó IPv4 é atribuído um endereço IPv6 IPv4-*mapped* (::ffff:V4ADDR). A operação de tradução dos endereços IPv6 em IPv4 é realizada pelo SIIT removendo os prefixos ::ffff:0:0:0/96 e ::ffff:0:0:0/96 aos endereços IPv6. De forma inversa, a operação de tradução dos endereços IPv4 em endereços IPv6 é realizada pelo SIIT acrescentando os respectivos prefixos.

Num cenário real de uma comunicação ponto-a-ponto na Internet, pode existir a necessidade de efectuar várias traduções à medida que o pacote atravessa redes com protocolos IP diferentes. Durante a operação de tradução, o mecanismo que implementa este algoritmo tem que processar, para posteriormente poder traduzir, cada um dos campos do cabeçalho IP. Estas operações implicam uma redução de desempenho da rede, razão pela qual, devemos reduzir ao mínimo o número de traduções que o pacote está sujeito ao longo do percurso. Numa situação em que é necessário fazer várias traduções, é preferível usar mecanismos de túnel no núcleo da rede e mecanismos de tradução junto dos terminais.

7.1.2.2. Tradução de IPv4 para IPv6

Na operação de tradução, o cabeçalho do pacote original (IPv4) é removido e substituído por um cabeçalho do tipo IPv6. Tanto o cabeçalho do protocolo de transporte como o seu campo de dados são mantidos inalterados durante a tradução, excepto nos pacotes ICMPv4.

A operação de fragmentação é uma das diferenças que existe entre os protocolos IPv4 e IPv6. Enquanto que no IPv4 todos os nós de um percurso podem realizar fragmentação, no IPv6 esta operação só pode ser realizada na origem. Assim, a execução do algoritmo *Path MTU Discovery* é obrigatória no protocolo IPv6 e opcional no IPv4.

Na execução do *Path MTU Discovery*, o nó IPv4 envia um pacote com o tamanho igual ao MTU usado na ligação e com o bit DF (*Don't Fragment*) activo. Caso o tamanho do pacote exceda o MTU de alguma das ligações da rede IPv4, o pacote é eliminado pelo *Router* dessa ligação que envia um pacote ICMPv4 *Packet Too Big* à origem. Do lado IPv6, quando o tamanho do pacote traduzido (IPv6) excede o MTU de uma dada ligação, é enviado um ICMPv6 *Packet Too Big* que depois de traduzido é entregue à origem. Através deste mecanismo, a origem pode ajustar o tamanho dos pacotes (IPv4) que envia para que nunca excedam o MTU das ligações IPv4 e IPv6 usadas. Assim o *Path MTU Discovery* pode ser usado entre a origem e o destino quando existe um tradutor de IPv4 para IPv6 no caminho usado. Neste caso, o tradutor só vai incluir nos pacotes traduzidos um cabeçalho de fragmentação se o pacote IPv4 já tiver sido fragmentado na origem.

Quando a origem não executa o mecanismo *Path MTU Discovery*, (ou seja, envia os pacotes com o bit DF inactivo) é o tradutor o responsável por garantir que o tamanho do pacote depois de traduzido não excede o MTU de nenhuma das ligações IPv6 usadas. Para cumprir este objectivo, o tradutor fragmenta os pacotes para que o tamanho máximo seja igual ao MTU (1280 bytes) mínimo definido no protocolo IPv6.

7.1.2.3. Tradução de IPv6 para IPv4

Na operação de tradução, o cabeçalho do pacote original (IPv6) é removido e substituído por um do tipo IPv4. Tanto o cabeçalho do protocolo de transporte como o seu campo de dados são mantidos inalterados durante a tradução, excepto nos pacotes ICMPv6.

Existem diferenças entre os protocolos IPv4 e IPv6 em relação ao valor mínimo do MTU adoptado. O IPv6 usa o MTU mínimo de 1280 bytes, enquanto que o IPv4 usa o valor de 68 bytes. Por conseguinte, durante a execução do *Path MTU Discovery*, quando existe um tradutor de IPv6 para IPv4 entre a origem e o destino, existe a possibilidade da origem (IPv6) receber um pacote ICMP *Packet Too Big* enviado por um *Router* IPv4 quando o tamanho dos pacotes que está a enviar é igual ao valor mínimo do MTU definido pelo IPv6 (1280 bytes). Quando esta situação ocorre, o protocolo IPv6 impõe que o nó IPv6 origem use o MTU mínimo de 1280 bytes mas que inclua um cabeçalho de extensão do tipo *Fragment Header* em todos os pacotes. Assim, só se pode garantir que não existe fragmentação entre a origem e o destino quando o MTU das ligações IPv4 for maior ou igual a 1280 bytes. Quando o MTU do caminho for menor que o MTU mínimo definido para o IPv6, a origem vai usar o MTU de 1280 bytes e os *Routers* IPv4 ficam encarregues de fragmentar o pacote depois de traduzido sempre que seja necessário.

7.1.3. NAT-PT

O conceito de NAT existe já há algum tempo nas redes IPv4, sendo a sua principal função a de “mascarar” uma rede com m endereços privados em n endereços públicos, em que normalmente $m > n$. O NAT do IPv4 e o mecanismo de transição NAT-PT [RFC2767] diferem em algumas características. No primeiro, o pacote traduzido usa a mesma versão do protocolo IP que o pacote original. No segundo, o pacote traduzido usa uma versão do protocolo IP diferente da que é usada pelo pacote original. O NAT-PT é um mecanismo de transição em que a tradução é

realizada ao nível do protocolo IP. Na operação de tradução é usado algoritmo SIIT para efectuar a tradução dos cabeçalhos das duas versões do protocolo IP.

Existem duas variantes do mecanismo NAT-PT [Figura 4], i) *Traditional NAT-PT* e ii) *Bi-Directional NAT-PT*.

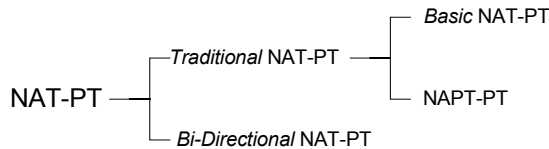


Figura 4 – Classificação dos mecanismos de tradução NAT-PT.

Na versão *Traditional NAT-PT*, apenas os terminais IPv6 podem iniciar as comunicações com os terminais IPv4. Esta variante do NAT-PT pode, por sua vez, ser dividida em duas sub-variantes: o *Basic NAT-PT* e o *NAPT-PT*.

Existem protocolos da camada de aplicação, por exemplo o DNS e o FTP, que colocam endereços IP no campo de dados. Uma vez que a tradução efectuada pelo NAT-PT, em qualquer uma das suas variantes, é realizada ao nível da camada de rede, este tipo de aplicações não funciona quando é usado este mecanismo de transição. Nestas situações, é necessário o uso de *Applications Level Gateways* (ALGs) em conjunto com o NAT-PT. Os ALGs têm a função de realizar a tradução dos endereços que são transportados no campo de dados. Devido às diferenças entre os protocolos que usam o campo de dados para transportar endereços, é necessário o uso de ALGs diferentes para cada um deles.

NAT-PT RegularNo mecanismo NAT-PT Regular (*Basic NAT-PT*) existe um bloco de endereços IPv4 a atribuir aos terminais IPv6 à medida que estes vão iniciando as sessões com os diferentes terminais IPv4. Podemos estabelecer tantas sessões, com pares origem destino diferentes, quantos os endereços IPv4 que constituem o bloco de endereços.

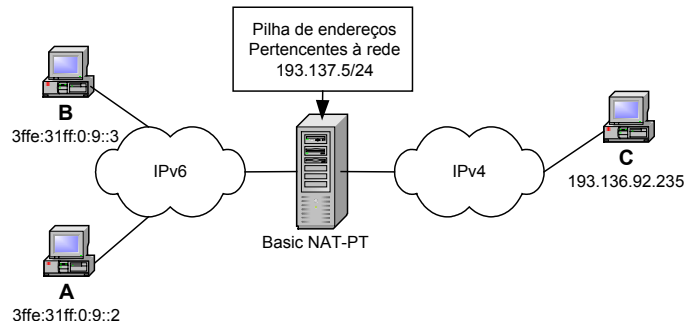


Figura 5: NAT-PT Regular.

A rede da Figura 5 exemplifica o funcionamento do NAT-PT Regular. Neste exemplo, o tradutor dispõe de uma pilha de endereços IPv4 que usam o prefixo 193.137.5.0/24. Assim a rede IPv6 é “vista” pelos nós IPv4 como se fosse uma rede IPv4 com o prefixo 193.137.5.0/24.

A rede IPv4 é “vista” pelos nós IPv6 como se fosse uma rede IPv6 (virtual) com o prefixo 3ffe:31ff:0:100::/96. Quando um nó IPv6 pretende comunicar com um nó ao qual esteja atribuído o endereço 193.136.92.235, usa o endereço destino 3ffe:31ff:0:100::193.136.92.235.

De acordo com a configuração adoptada no tradutor, é necessário que os protocolos de encaminhamento tanto da rede IPv4 como da rede IPv6 garantam que os pacotes cujo endereço destino pertença aos prefixos 193.137.5.0/24 (do lado IPv4) e 3ffe:31ff:0:100::/96 (do lado IPv6) sejam encaminhados por este tradutor. Os pacotes enviados pelo terminal A (IPv6) com destino ao terminal C (IPv4) usam os seguintes endereços:

Endereço IP origem:	3ffe:31ff:0:9::2
Endereço IP destino:	3ffe:31ff:0:100::193.136.92.235 (endereço virtual do terminal C)

No caso de ser o primeiro pacote da comunicação entre estes nós, o NAT-PT selecciona um dos endereços IPv4 da sua pilha de endereços que esteja disponível (por exemplo, o endereço 193.137.5.223). A partir deste momento, os parâmetros referentes à tradução são guardados para serem utilizados na tradução dos pacotes seguintes que pertençam ao mesmo par origem/destino.

Depois de ter sido atribuído o endereço IPv4 193.137.5.223 ao terminal A, os pacotes enviados com destino ao terminal C usam os endereços seguintes:

Endereço IP origem:	193.137.5.223 (endereço virtual atribuído ao terminal A)
Endereço IP destino:	193.136.92.235

Uma vez que os endereços IPv4 são atribuídos dinamicamente aos terminais IPv6, o mecanismo de tradução NAT-PT do tipo NAT-PT Regular é *Stateful*. Por este motivo, é necessário que todos os pacotes pertencentes a uma dada sessão sejam traduzidos por um único dispositivo *Basic* NAT-PT.

7.1.3.1. NAPT-PT

O mecanismo de tradução *Network Address Port Translation – Protocol Translation* (NAPT-PT) para além de traduzir o protocolo IP e o ICMP, também traduz o número dos portos dos protocolos de transporte. O NAPT-PT pode ser usado por vários nós IPv6 para comunicarem com nós IPv4 recorrendo a um único endereço IPv4.

No NAT-PT, quando se esgota a pilha de endereços IPv4, deixa de ser possível servir mais nós IPv6. Através da tradução dos números dos portos dos protocolos TCP e UDP, o NAPT-PT permite manter simultaneamente 2^{16} sessões TCP e 2^{16} sessões UDP usando um único endereço IPv4. Desta forma, com o NAPT-PT é possível aumentar o número de nós IPv6 que podem ser servidos simultaneamente sem ser necessário aumentar o número de endereços IPv4. Durante a tradução dos pacotes, os números dos portos de origem são modificados pelo tradutor. Os novos números para o porto origem são determinados pelo estado do tradutor, por exemplo, pelas sessões activas. Por este motivo, os pacotes pertencentes à mesma sessão devem ser traduzidos por um único dispositivo de tradução, uma vez que o NAPT-PT é um mecanismo do tipo *Stateful*.

A Figura 6 ilustra uma rede onde é usado o NAPT-PT para permitir a comunicação entre os nós das redes IPv6 e IPv4. À interface IPv4 do NAPT-PT foi atribuído o endereço 193.137.5.1, enquanto que a interface IPv6 usa o endereço 3ffe:31ff:0:9::1. O NAPT-PT foi configurado para usar o prefixo de tradução 3ffe:31ff:0:100::/64. Tal como no « NAT-PT Regular, é necessário que os protocolos de encaminhamento da rede IPv6 garantam que os pacotes cujo endereço destino pertença ao prefixo 3ffe:31ff:0:100::/96 sejam encaminhados por este tradutor. No entanto, o NAPT-PT não requer o mesmo tipo de requisito do lado da rede IPv4.

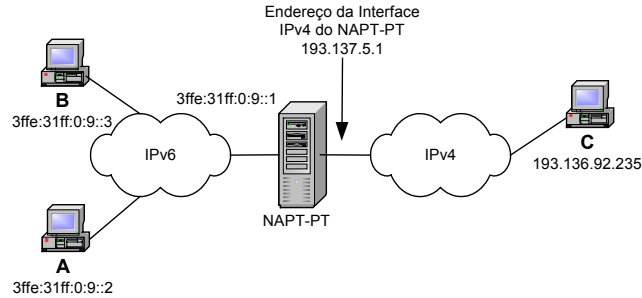


Figura 6: Exemplo de Utilização de NAPT-PT.

Supondo que o nó A (IPv6) pretende iniciar uma sessão *Telnet*⁵ com o nó C (IPv4) iria criar um pacote com os parâmetros seguintes:

Endereço IP origem:	3ffe:31ff:0:9::2
Endereço IP destino:	3ffe:31ff:0:100::193.136.92.23
Porto TCP origem:	2000
Porto TCP destino:	23

O endereço IP de destino é formado pelo prefixo de tradução (3ffe:31ff:0:100::/64) e pelo endereço IPv4 (136.92.235) do nó C.

O pacote enviado pelo nó A com destino ao nó C vai ser encaminhado para o NAPT-PT, onde é traduzido. Após a tradução, o pacote expedido pela interface IPv4 do NAPT-PT tem os parâmetros seguintes:

Endereço IP origem:	193.137.5.1
Endereço IP destino:	193.136.92.235
Porto TCP origem:	1500
Porto TCP destino:	23

O endereço destino do pacote IPv4 traduzido, corresponde aos últimos 32 bits do endereço de destino do pacote IPv6 recebido pelo NAPT-PT, enquanto que o endereço origem é dado pelo endereço IPv4 do tradutor. O tráfego recebido pelo NAPT-PT com IP origem igual a 193.136.92.235, e cujo porto origem seja o 23, é reconhecido como fazendo parte da mesma sessão. Depois de traduzido, o pacote é entregue ao nó A.

7.1.3.2. NAT-PT Bi-Direccional

Na versão do NAT-PT *Bi-Direccional*, as sessões podem ser iniciadas tanto pelos terminais IPv4 como pelos terminais IPv6. Na tradução, a correspondência dos endereços das duas pilhas pode ser dinâmica ou estática. No caso em que a correspondência entre os endereços das duas famílias é dinâmica, toda a informação que o tradutor necessita para efectuar a tradução dos endereços está embecida nos endereços dos pacotes que vão ser convertidos. Os terminais IPv6 usam endereços IPv6 IPv4 traduzidos. Estes endereços são caracterizados pelo prefixo

⁵ por omissão, as sessões de *telnet* usam o porto destino 23 do protocolo TCP.

::ffff:0:0:0/96 seguido pelo endereço IPv4 atribuído ao nó IPv6. Aos nós IPv4 é atribuído do lado IPv6 um endereço IPv6 do tipo IPv6 IPv4-*mapped* [RFC 2373]. Estes endereços usam o prefixo ::ffff:0:0:0/96, seguido do endereço IPv4 atribuído ao nó (IPv4).

Devido ao uso dos endereços IPv6 IPv4-*mapped* e IPv6 IPv4 traduzidos, o processo de tradução entre os endereços das duas versões do protocolo IP é realizado à custa da inserção e da remoção de prefixos. Para os pacotes enviados pelo terminal IPv6, o tradutor remove os prefixos ::ffff:0:0:0/96 e ::ffff:0:0:0/96 dos endereços origem e destino, respectivamente. No sentido inverso, o tradutor adiciona os prefixos ::ffff:0:0:0/96 e ::ffff:0:0:0/96 aos endereços origem e destino.

Na rede da Figura 7, é usado o NAT-PT Bi-direccional na comunicação entre os nós IPv6 e os IPv4. A rede IPv4 usa o prefixo 193.136.92.0/24, enquanto que, a rede IPv6 é caracterizada pelo prefixo ::ffff:0:0:0/96.

Tal como no *Basic* NAT-PT, os protocolos de encaminhamento deverão garantir o encaminhamento pelo tradutor para o prefixo 193.137.5.0/24 (na rede IPv4) e para o prefixo ::ffff:0:0:0/96 (na rede IPv6).

Os pacotes enviados pelo nó A (IPv6) para o nó C (IPv4) usam os endereços seguintes:

Endereço IP origem:	::ffff:0:193.137.5.12
Endereço IP destino:	::ffff:193.136.92.235

Depois de traduzido para IPv4 o pacote é caracterizado pelos endereços:

Endereço IP origem:	193.137.5.12
Endereço IP destino:	193.136.92.235

No sentido inverso, os pacotes enviados pelo nó C com destino ao nó A, usam os endereços:

Endereço IP origem:	193.136.92.235
Endereço IP destino:	193.137.5.12

Após ter ocorrido a tradução de IPv4 para IPv6 são usados os endereços:

Endereço IP origem:	::ffff:193.136.92.235
Endereço IP destino:	::ffff:0:193.137.5.12

Dado que a informação usada na tradução dos endereços é transportada no esquema de endereçamento usado, não é necessário que os pacotes pertencentes à mesma sessão sejam todos traduzidos por um único tradutor. Por conseguinte, o NAT-PT Bi-direccional é do tipo *Stateless*.

Tal como foi inicialmente mencionado, a correspondência entre os endereços das duas famílias do protocolo IP pode ser estabelecida estaticamente em alternativa ao método dinâmico. A informação respeitante à correspondência entre os endereços IPv4 e IPv6 é guardada pelo NAT-PT numa tabela que é consultada sempre que é necessário traduzir um endereço. Esta tabela é configurada manualmente e é local a cada tradutor.

Para ilustrar o caso de correspondência estática, considere-se o cenário da Figura 9 com duas redes, uma IPv4 e outra IPv6. Na comunicação entre os nós das duas redes, é usado o tradutor *Bi-Directional* NAT-PT. No tradutor foi manualmente configurada a correspondência entre os endereços IPv4 e IPv6 de acordo com o conteúdo da Tabela 5.

Tabela 5: Correspondência Entre Endereços IPv4 e IPv6

Endereço IPv6:	Endereço IPv4:
3ffe:31ff:0:9::1 (endereço real)	193.137.100.1 (endereço virtual)
3ffe:31ff:0:9::2 (endereço real)	193.137.100.2 (endereço virtual)
3ffe:31ff:0:100::235 (endereço virtual)	193.136.92.235 (endereço real)

A tabela foi especificada de modo que a rede IPv6 seja “vista” pela rede IPv4 como se fosse uma rede IPv4 com o prefixo 193.137.100.0/24 e, a rede IPv4 seja “vista” pela rede IPv6 como se fosse uma rede IPv6 com o prefixo 3ffe:31ff:0:100/64. Tal como nos casos anteriores, assume-se que os protocolos de encaminhamento respeitam esta configuração.

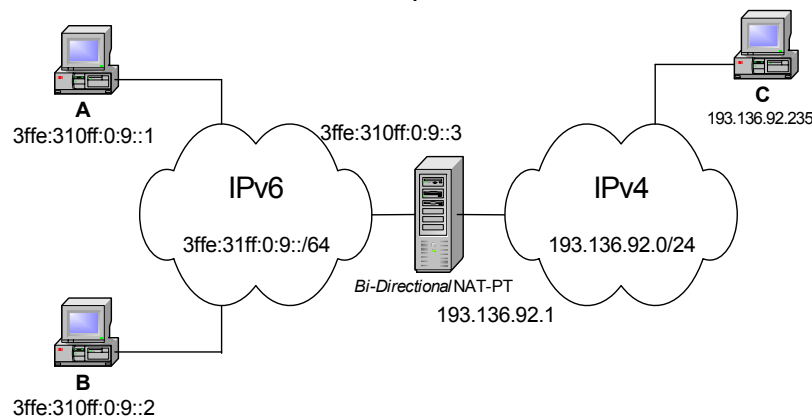


Figura 7: NAT-PT Bi-direccional.

Quando o nó A (IPv6) pretende comunicar com o nó C (IPv4), no pacote IPv6 são usados os endereços seguintes:

Endereço IP origem:	3ffe:31ff:0:9::1
Endereço IP destino:	3ffe:31ff:0:100::235

Depois de traduzido, são usados os endereços:

Endereço IP origem:	193.137.100.1
Endereço IP destino:	193.136.92.235

Na resposta o nó C usa os endereços:

Endereço IP origem:	193.136.92.235
---------------------	----------------

Endereço IP destino:	193.137.100.1
----------------------	---------------

O pacote que foi enviado pelo nó C com destino ao nó A, depois de traduzido usa os endereços:

Endereço IP origem:	3ffe:31ff:0:100::235
Endereço IP destino:	3ffe:31ff:0:9::1

O NAT-PT Bi-direccional que usa a correspondência estática entre os endereços IPv6 e IPv4 é do tipo *Stateless*, desde que se garanta que os vários tradutores usam as mesmas correspondências entre endereços.

7.1.4. TRT - Transport Relay Translator

O TRT [\[RFC3142\]](#) é um mecanismo de transição que realiza a tradução ao nível da camada de transporte. Este mecanismo é usado para realizar a tradução entre sessões sobre IPv6 e sessões sobre IPv4 mas tem a limitação de permitir apenas que as sessões sejam iniciadas pelos nós IPv6.

O tradutor deve estar localizado na fronteira que separa as redes IPv6 e IPv4. Não é necessário realizar nenhuma alteração nos nós terminais que pretendam comunicar entre si através deste mecanismo.

Na comunicação entre dois nós são estabelecidas duas sessões, uma entre o nó origem e o tradutor, e outra entre o tradutor e o nó de destino. O funcionamento do TRT é similar ao de um *Proxy*, uma vez que a sessão iniciada pelo nó IPv6 é terminada no tradutor, e a sessão iniciada no tradutor é terminada no nó IPv4.

É usado um prefixo IPv6 de 64 bits (pr efixo de tradução) através do qual os nós IPv6 identificam os nós IPv4, ou seja as redes IPv4 são “vistas” pelos nós IPv6 como se fossem uma rede IPv6 cujo prefixo coincide com o prefixo de tradução. Quando um nó IPv6 pretende comunicar com um nó IPv4, o endereço de destino é constituído pelo prefixo de tradução e pelo endereço IPv4 do nó destino. Por exemplo, caso fosse usado o prefixo de tradução 3ffe:3103:0:108::/64 e ao nó IPv4 estivesse atribuído o endereço 193.136.92.219, então o endereço de destino seria dado por 3ffe:3103:0:108::193.136.92.219. Note-se que o encaminhamento na rede IPv6 deverá ser tal que todos os pacotes cujo endereço de destino coincida com o prefixo de tradução sejam encaminhados para o TRT. De forma a ilustrar o funcionamento deste mecanismo de tradução, considere-se o cenário ilustrado na Figura 8, formado por duas redes, uma IPv4 com o prefixo 193.136.92.0/24 e outra IPv6 com o prefixo 3ffe:3103:0:100::/64. Na fronteira destas redes encontra-se o TRT que usa o prefixo de tradução 3ffe:3103:0:108::/64.

Considere-se que o nó A (IPv6) pretende estabelecer uma sessão TCP com o nó B (IPv4) do porto origem 4035 (número variável atribuído pelo sistema operativo) para o porto destino 22 (porto normalizado do serviço SSH). O nó A estabelece uma sessão para o endereço destino 3ffe:3103:0:108::193.136.92.235. O estabelecimento desta sessão desencadeia o estabelecimento de uma sessão IPv4 do TRT para o nó B. Esta sessão é caracterizada pelo tuplo constituído pelo endereço origem 193.136.92.135, (que corresponde ao endereço da interface IPv4 do TRT), pelo endereço destino 193.136.92.235 (determinado pelos últimos 32 bits do endereço destino da sessão IPv6), pelo porto origem 3025 (número variável atribuído pelo sistema) e pelo porto destino 22 mantido da sessão IPv6.

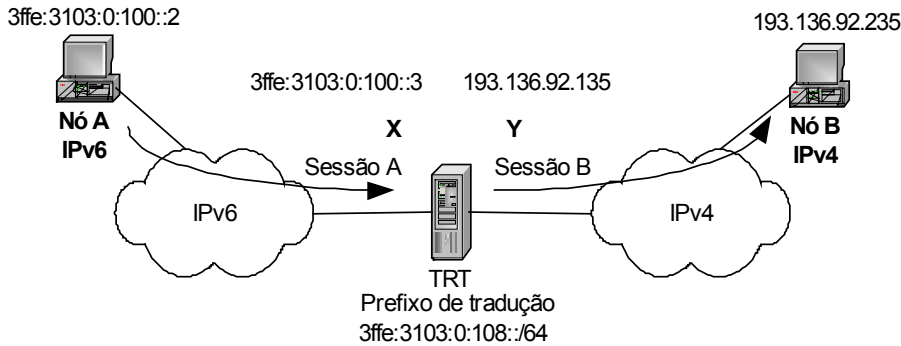


Figura 8: Exemplo de Utilização do Mecanismo TRT.

Dado que são usadas duas sessões independentes entre a origem e o destino, nos cenários de transição onde é usado este mecanismo de tradução não se colocam os problemas decorrentes do uso de MTUs diferentes entre as redes IPv4 e as redes IPv6, como acontece quando a tradução é feita ao nível do protocolo IP.

O [RFC 3142] (que especifica este mecanismo) refere ainda que no mesmo *Site* podem ser usados vários mecanismos TRT, para resolver possíveis problemas de escalabilidade. A cada um dos tradutores é atribuído um prefixo de tradução diferente. Através da escolha de qual dos prefixos de tradução usado, o DNS pode realizar a operação de balanceamento de carga entre os vários tradutores.

De acordo com o funcionamento do TRT, os pacotes pertencentes a uma dada sessão devem ser traduzidos por um único tradutor (por causa da escolha do número de porto origem). Assim, estamos na presença de um mecanismo de transição do tipo *Stateful*.

7.2. Túneis

Os túneis são mecanismos de encapsulamento usados para suportar as comunicações entre redes do mesmo protocolo de rede que estejam fisicamente interligadas apenas por redes da outra versão do protocolo IP. Neste tipo de mecanismos, aos pacotes de uma dada versão do protocolo IP é-lhes adicionado um novo cabeçalho pertencente a outra versão do protocolo IP. À operação de adição do novo cabeçalho é dado o nome de encapsulamento, sendo a operação inversa denominada por desencapsulamento. A primeira operação é realizada no início do túnel, sendo a segunda realizada no fim do túnel. Do ponto de vista lógico, um túnel é definido pela associação entre os endereços IP de início e de final de túnel.

No contexto dos cenários de transição IPv4/IPv6, podem existir dois tipos de túneis:

- **túneis IPv6 sobre IPv4.** Estes serão os túneis mais usados durante a fase inicial do período de transição, pois as infra-estruturas de rede usam maioritariamente o protocolo IPv4. Neste tipo de túneis, aos pacotes IPv6 é-lhes adicionado um cabeçalho IPv4 [Figura 11], com o objectivo de se poderem usar as redes IPv4 para os transmitir **túneis IPv4 sobre IPv6.** Neste tipo de túneis, aos pacotes IPv4 é adicionado um cabeçalho IPv6 de forma a poderem ser encaminhados através das infra-estruturas IPv6. Este tipo de túneis será usado numa fase avançada do período de transição, quando o protocolo IPv6 estiver maioritariamente implementado nas redes de trânsito.

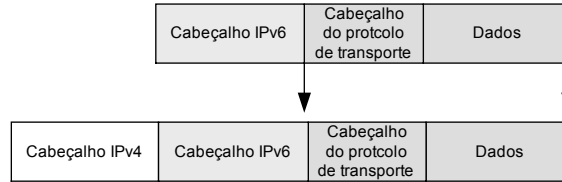


Figura 9: Encapsulamento IPv6 em IPv4.

Ainda no contexto dos cenários de transição, os túneis podem ter origem e destino em nós intermédios (*Routers*) ou em sistemas terminais. Os túneis podem ainda ser classificados de acordo com a forma usada para determinar o endereço de fim de túnel:

- **túneis automáticos.** Os sistemas que fazem o encapsulamento determinam automaticamente o endereço de fim de túnel. Neste tipo de túneis, são usados esquemas de endereçamento nos quais é inserida a informação respeitante ao endereço de fim de túnel;
- **túneis configurados.** Os sistemas que fazem o encapsulamento usam informação que lhes foi previamente configurada para determinar o endereço de fim de túnel.

Na maioria dos casos, a comunicação entre dois nós recorre apenas a um túnel. No entanto, em algumas situações é necessário recorrer a mais do que um túnel. Os túneis podem existir de uma forma hierárquica (túnel dentro de um túnel) ou sequencial (concatenação de túneis).

Estamos na presença de túneis hierárquicos, quando os túneis destinados à transição IPv4/IPv6 coexistem com outros túneis com funções de segurança ou de diferenciação de serviço. Por exemplo, o tráfego IPv6 pode ser encapsulado em IPv4 (para poder usar a infra-estrutura de encaminhamento IPv4), sendo de novo encapsulado em IPv4 pelo protocolo IPsec [RFC 1825] (por questões de segurança). Os túneis são sequenciais quando ao longo de um caminho são usados vários túneis, sem existirem túneis sobre túneis. Por exemplo, pode ser usado um túnel desde o sistema terminal até ao seu *Router*, e a partir daí outro túnel até ao destino final.

7.2.1. Túneis Baseados em Endereços IPv6-IPv4 Compatíveis

Os endereços IPv6 IPv4 compatíveis (::V4ADDR) são constituídos pelo prefixo ::/96, e por um endereço IPv4 nos últimos 32 bits. Este tipo de endereços pode ser usado no estabelecimento de túneis automáticos [RFC 1933] IPv6 sobre IPv4. Os nós que implementam os extremos do túnel têm suporte de *pilha dupla*. Estes nós usam endereços IPv6 IPv4 compatíveis, nos quais os últimos 32 bits são determinados pelos endereços IPv4 que lhes foram atribuídos.

Embora conceptualmente simples, o uso de endereços IPv6 IPv4 compatíveis no estabelecimento de túneis apresenta as seguintes limitações:

- os endereços IPv6 IPv4 compatíveis só usam 32 dos 128 bits disponíveis no esquema de endereçamento IPv6;
- a organização hierárquica de endereçamento IPv6 é ignorada;
- é necessário atribuir aos dispositivos que implementam os extremos do túnel um endereço IPv4, o que faz com que este mecanismo de transição não resolva o problema da exaustão do espaço de endereçamento do IPv4.

7.2.2. 6to4

No mecanismo de transição 6to4 [RFC3056], é atribuído aos nós participantes um endereço

6to4, cuja estrutura se apresenta na Figura 10.. O campo V4ADDR do endereço é de extrema importância uma vez que permite obter o endereço IPv4 do fim do túnel.

3 bits	13 bits	32 bits	16 bits	64 bits
FP 001	-TLA ID 0x002 ⁶	V4ADDR	SLA	Identificador da interface
2002		V4ADDR	SLA	Identificador da interface

Legenda:

FP -	Format Prefix
TLA ID -	Top Level Aggregation Identifier
V4ADDR -	Endereço IPv4 do fim do túnel.
SLA ID -	Site Level Aggregation Identifier.
Interface ID -	Link Level Host Identifier.

Figura 10: Formato de Endereços IPv4.

O mecanismo de transição 6to4 é implementado à custa das seguintes entidades:

- **terminal 6to4.** Trata-se de um terminal com suporte IPv6 ao qual foi atribuído, por configuração manual ou automática, um endereço do tipo 6to4;
- **6to4 Router** . Este *Router* dispõe de um endereço IPv4 global e de um endereço IPv6 6to4. No endereço 6to4 é usado no campo V4ADDR o endereço IPv4 que lhe foi atribuído. É este o dispositivo que se encarrega das operações de encapsulamento e de descapsulamento;
- **Site 6to4.** Rede que usa um prefixo 2002::V4ADDR::/48. O campo V4ADDR é determinado pelo endereço IPv4 global atribuído à interface IPv4 do 6to4 *Router*;
- **6to4 Relay Router** . *Router* que suporta encapsulamento 6to4 e que é usado por nós IPv6 6to4 para comunicarem com nós IPv6 nativos e vice-versa;

A rede da Figura 11 é composta por dois *sites* 6to4 que usam os prefixos 2002:c189:0501::/48⁷ e 2002:c189:0502::/48, por uma rede IPv6 com o prefixo 3ffe:3102:ffff:0:9::/64 e por uma rede IPv4 que é usada para ligar as outras redes;

Os *Routers* 6to4 A e B anunciam os prefixos 2002:c189:0501::/48 e 2002:c189:0501::/48 para o interior dos seus sites e o 6to4 *Relay Router* D anuncia o prefixo 3ffe:3102:ffff:0:9::/64 para a rede IPv6. Estes anúncios são usados pelos terminais 6to4 e IPv6 na auto-configuração de endereços e na configuração do *Default Gateway*.

⁶ Atribuído pela IANA

⁷ C189:0501 corresponde à representação hexadecimal do endereço IPv4 193.137.5.1.

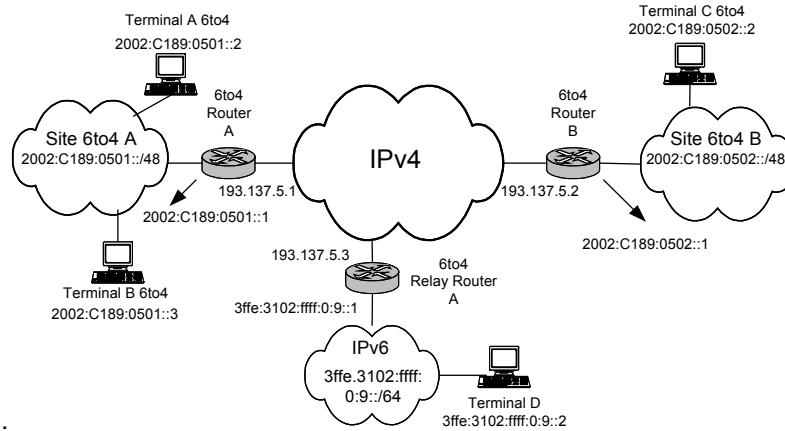


Figura 11: Exemplo de cenário 6to4.

Nas comunicações entre os nós que pertencem ao mesmo *site* 6to4, não é necessário recorrer a nenhum mecanismo de transição.

Quando o terminal A pretende comunicar com o terminal C, o terminal A envia um pacote IPv6, para o 6to4 Router A, com os endereços IP origem 2002:c189:0501::2 e IP destino 2002:c189:0502::2.

O router 6to4 A ao analisar o endereço de destino, determina que se trata de um endereço IPv6 6to4. Desta forma, o endereço do fim do túnel pode ser determinado à custa do endereço IPv6 destino. Depois de determinado o endereço de fim de túnel, que neste caso corresponde a 193.137.5.2 (c189:0501), é adicionado ao pacote IPv6 um cabeçalho IPv4 com o endereço IP origem 193.137.5.1 (endereço da interface IPv4 do Router A) e o endereço IP de destino 193.137.5.2 (endereço do Router B).

No outro extremo do túnel, o pacote é desencapsulado e é entregue ao nó C. No sentido inverso, o processo envolvido na entrega dos pacotes enviados pelo terminal C com destino ao terminal A é semelhante ao que foi descrito no caso em que o terminal A envia um pacote ao terminal C.

Na situação em que o terminal A que pertence a um Site 6to4, pretender comunicar com o terminal D que pertence a uma rede IPv6 nativa, o terminal A envia ao Router A um pacote IPv6 cujo endereço origem corresponde a 2002:c189:0501::2 e com o endereço de destino 3ffe:3102:fff:9::2. Nesta situação, o RouterA não consegue determinar o endereço do outro extremo do túnel à custa do endereço destino. Neste caso, é necessário configurar um túnel que é usado para enviar todos os pacotes IPv6 cujos endereços não obedecem ao esquema de endereçamento definido pelo 6to4. Por conseguinte, no Router A foi necessário configurar um túnel para o router de relay D. Desta forma, quando o pacote IPv6 enviado pelo terminal A com destino a D chega ao Router A é encapsulado e enviado pelo túnel definido pelos endereços 193.137.5.1 e 193.137.5.3. No sentido inverso, quando o terminal D pretende comunicar com terminal A, envia um pacote com os endereços origem 3ffe:3102:fff:9::2 e destino 2002:c189:0501::2. Este pacote é enviado em primeiro lugar para o Router D, que à custa do endereço destino consegue determinar qual o endereço IPv4 do outro extremo do túnel.

7.3. Cenários de Transição Aplicados a Redes GPRS

Nesta secção são identificados alguns cenários de transição [\[RFC3574\]](#) e para cada um deles é proposta uma solução baseada nos mecanismos de transição descritos anteriormente, que possibilita que um terminal GPRS comunique com uma estação, independentemente da versão

do protocolo IP em uso.

Foram identificados os seguintes cenários:

- **cenário 1** O UE (*User Equipment*) suporta IPv4 e comunica com estações IPv4 e estações IPv6. É usada uma rede IPv6 para ligar a rede do operador móvel com as redes IPv4 e IPv6;
- **cenário 2** - O UE (*User Equipment*) tem suporte *pilha dupla* e comunica com estações IPv4 e estações IPv6;
- **cenário 3** – O UE (*User Equipment*) suporta IPv6 e comunica com estações IPv4 e estações IPv6. É usada uma rede IPv4 para ligar a rede do operador móvel com as redes IPv4 e IPv6.

À excepção do cenário 2, no qual os equipamentos móveis (UE) suportam os dois protocolos, nenhum dos cenários considera o uso do mecanismo de transição no UE. Por um lado, o mecanismo de transição introduz um aumento da complexidade (do software e do hardware) do UE. Por outro, implica um aumento da necessidade do processamento no UE, que se irá reflectir no aumento do consumo de energia deste equipamento.

7.3.1. Cenário 1

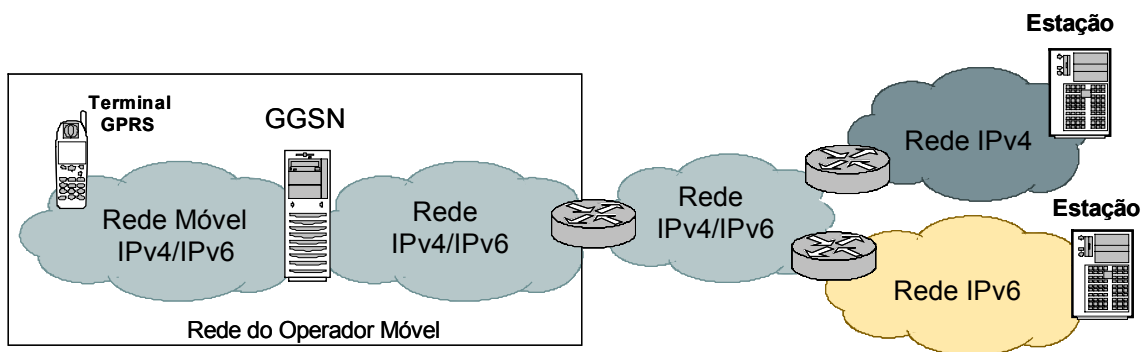


Figura 12: Cenário de Transição 1.

No cenário ilustrado na Figura 12, considera-se que a rede de transito é uma rede IPv6, enquanto que na rede do operador é usado o protocolo IPv4.

Uma vez que neste cenário os extremos da comunicação (UE e Estação IPv4) usam o mesma versão do protocolo IP, é suficiente o uso de um túnel IPv4 sobre IPv6 entre o *Gateway* do operador móvel e o *Gateway* da rede da Estação IPv4. Só se considera o uso de túneis IPv4 sobre IPv4 manualmente configurados, já que o IETF não propôs (até à data) nenhum mecanismo que implemente túneis automáticos IPv4 sobre IPv6.

Este cenário também considera o caso em que o terminal GPRS comunica com uma estação IPv6. Para que tal seja possível, é necessário usar um tradutor no *Gateway* da rede do operador móvel. O mecanismo de tradução TRT não é considerado, uma vez que não permite que a ligação seja iniciada pelo terminal GPRS (IPv4). Desta forma, apenas se considera o uso do mecanismo NAT-PT Bi-direcional.

O serviço de DNS pode ser garantido se forem considerados dois servidores de DNS, um no “mundo IPv4” e outro no “mundo IPv6”, tendo que ser garantida a coerência entre os endereços usados pelo tradutor e os endereços guardados nos servidores DNS.

7.3.2. Cenário 2

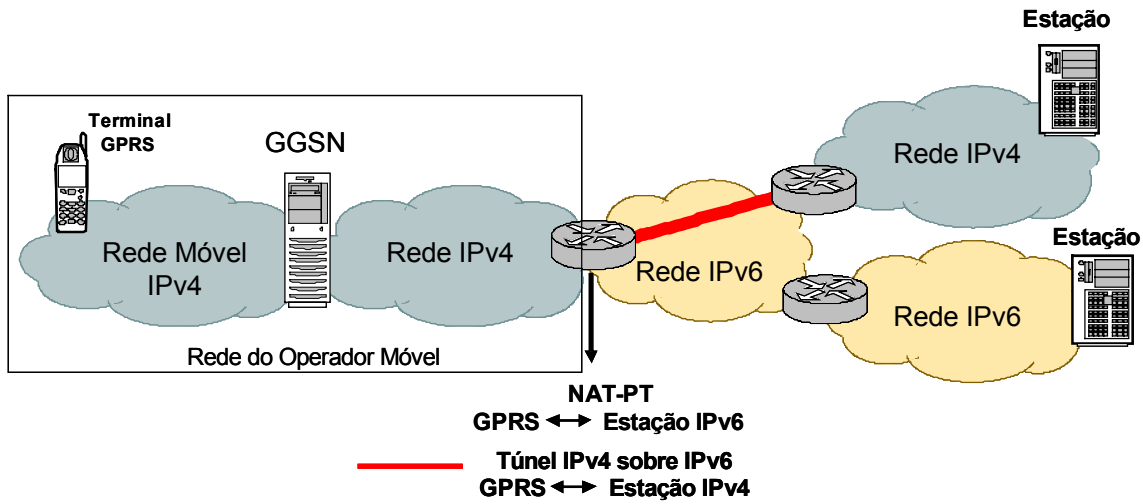


Figura 13: Cenário de transição 2.

No segundo cenário (ver Figura 13), considera-se que as redes do operador móvel assim como a rede de transito suportam as duas versões do protocolo IP. Dado que o UE suporta IPv4 e IPv6 pode comunicar com estações IPv4 e com estações IPv6 sem necessitar de recorrer a outros mecanismos de transição, uma vez que na comunicação com as estações IPv4 é usado o protocolo IPv4, enquanto que na comunicação com as estações IPv6 é usado o protocolo IPv6.

Neste cenário, o UE decide qual das pilhas protocolares deve usar mediante o tipo do endereço devolvido pelo servidor DNS. Caso seja devolvido um endereço IPv6, o UE usa a pilha IPv6, caso contrário é usada a pilha protocolar IPv4.

Apesar de ser conceptualmente simples, esta solução para além de se basear no uso de endereços IPv4 (recurso escasso) introduz complexidade adicional nas redes e nos equipamentos que necessitam de suportar as duas versões do protocolo IP. Note-se que de acordo com esta solução, as aplicações instaladas no UE devem suportar as duas versões do protocolo IP, para que não haja a necessidade de existirem aplicações duplicadas, uma para cada uma das versões do protocolo IP.

Este cenário não resolve os problemas decorrentes do uso do protocolo IPv4 nas redes móveis, uma vez que o problema da escassez do espaço de endereçamento do protocolo IPv4 foi remediado pelos operadores através do uso de endereços privados.

7.3.3. Cenário 3

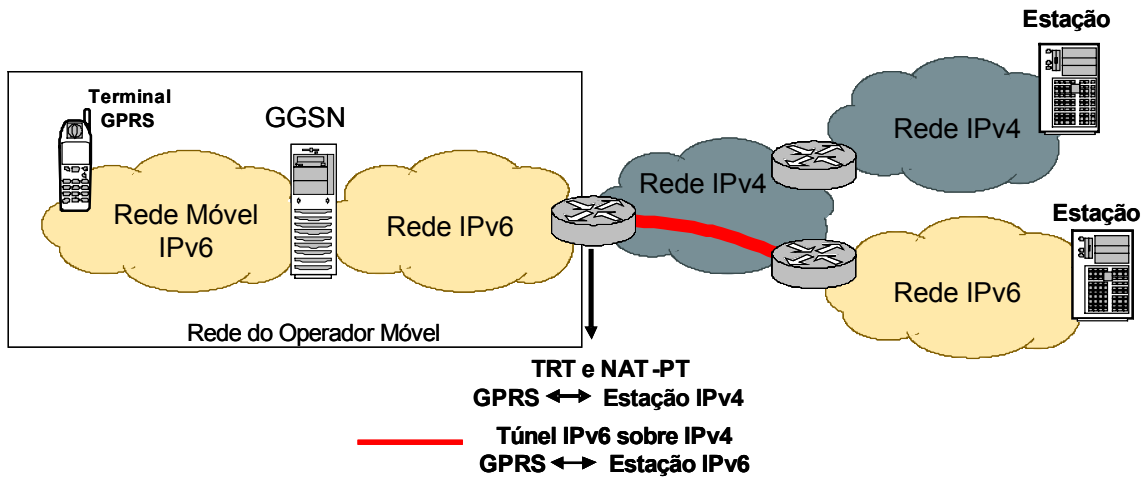


Figura 14: Cenário de transição 3.

O terceiro cenário encontra-se ilustrado na Figura 14. Para que seja possível a comunicação entre o terminal GPRS e a estação IPv4 é necessário usar um tradutor no *Gateway* da rede do operador móvel. Este tradutor é usado para traduzir os pacotes IPv4 em pacotes IPv6, e vice-versa. O tradutor pode ser implementado pelos mecanismos de transição NAT-PT ou TRT. Note-se que quando é usado o TRT as comunicações só podem ser iniciadas pela estação que usa IPv6 (neste caso o UE).

A comunicação entre o terminal GPRS e um estação IPv6, garante-se através do uso de um túnel IPv6 sobre IPv4 entre o *Gateway* do operador móvel e o *Gateway* da rede da Estação IPv4, uma vez que os extremos da comunicação (UE e Estação IPv4) usam o mesma versão do protocolo IP. Os túneis a usar podem ser túneis IPv6 sobre IPv4 configurados, túneis automáticos baseados em endereços IPv6 IPv4 compatíveis ou túneis 6to4. Note-se que o uso de túneis IPv6 sobre IPv4 automáticos torna a solução deste cenário mais escalável do ponto de vista da administração dos túneis. No entanto os mecanismos através dos quais são implementados os túneis automáticos usam esquemas de endereçamento IPv6 próprios a cada um destes mecanismos, pelo que as redes IPv6 que comunicam através deste mecanismo também devem usar esquemas de endereçamento compatíveis com estes mecanismos. Os túneis baseados em endereços IPv6 IPv4 compatíveis (tal como o nome sugere) usam endereços IPv6 IPv4 compatíveis (::IPv4). Por sua vez, o mecanismo 6to4 usa endereços com a forma 2002::V4ADDR::/48. Devido à escassez dos endereços IPv4, o mecanismo 6to4 é o melhor dos dois mecanismos através dos quais é possível implementar túneis IPv6 sobre IPv4, já que o mecanismo 6to4 necessita apenas de um endereço IPv4 global.

Relativamente ao serviço de DNS, considerado de fundamental importância no funcionamento de uma rede IP, é necessário articular este serviço com a presença dos mecanismos de transição. Por conseguinte, é determinante que exista coerência entre os endereços usados pelo tradutor e os endereços guardados nos servidores DNS.

8. Resumo e Conclusões

Neste documento, apresentámos uma análise do estado actual de especificação e de desenvolvimento de mobilidade IPv6. Começámos por introduzir noções e conceitos

relacionados com mobilidade IP em geral, e particularizámos para mobilidade IPv6.

Com o intuito de ajudar à futura criação de bancadas heterogéneas de testes de mobilidade IPv6, descrevemos modelos genéricos de utilização de IPv6 em WLANs e em GPRS/UMTS. Apresentámos diferentes tipos de *software* e de *hardware* passível de utilização para a criação dos cenários referidos, e respectiva configuração.

Adicionalmente, foram ainda descritos os diversos mecanismos de transição IPv4/IPv6, incluindo configurações e exemplos de aplicação à tecnologia GPRS, dado que estas redes não suportam ainda IPv6 nativo. Deste levantamento e análise, retirámos várias conclusões. A primeira é de que IPv6 pode ser facilmente aplicado à tecnologia WLAN, para criação de cenários básicos. É, no entanto, necessário investigar os problemas relacionados com o desempenho das comunicações (*handovers*) quando um dispositivo móvel muda de uma rede para outra.

Outra conclusão é de que embora os dispositivos terminais GPRS possam utilizar IPv6, não existe suporte nativo no *core*. Consequentemente, é necessário utilizar túneis para estabelecer conectividade IPv6.

Os problemas detectados devem-se maioritariamente ao facto de a introdução de tecnologias *wireless na* Internet ser extremamente recente e ainda devido ao facto de o IPv6 ser encarado ainda com desconfiança.

Finalmente, é absolutamente necessário compreender o funcionamento de mobilidade IPv6 em cenários utilizando tecnologias heterogéneas, para providenciar uma migração transparente. Tal processo requiere testes adicionais, para os quais este documento pretende contribuir com especificações básicas, e processos a seguir.

9. Contacto dos Autores

Jorge Sá Silva

DEI – Universidade de Coimbra
Pinhal de Marrocos, 3030 Coimbra
E-mail: sasilva@dei.uc.pt

Tiago Camilo

DEI – Universidade de Coimbra
Pinhal de Marrocos, 3030 Coimbra
E-mail: tandre@ipg.pt

Sérgio Duarte

IPG – Instituto Politécnico da Guarda
Av. Dr. Francisco Sá Carneiro, 50 6300 - 559 Guarda
E-mail: sduarte@ipg.pt

Rui Mendes

DEI – Universidade de Coimbra
Pinhal de Marrocos, 3030 Coimbra

E-mail: rmendes@student.dei.uc.pt

André Costa

DEI – Universidade de Coimbra
Pinhal de Marrocos, 3030 Coimbra
E-mail: adcosta@student.dei.uc.pt

Luís Miguel Lopes de Oliveira

DEI - Instituto Politécnico de Tomar
Quinta do Contador - Estrada da Serra
2300 Tomar

Instituto de Telecomunicações - Pólo de Aveiro
Campus de Santiago
Universidade de Aveiro
3800 Aveiro
E-mail: loliveira@ipt.pt ; loliveira@av.it.pt

António Amaral

Instituto de Telecomunicações - Pólo de Aveiro
Campus Universitário 3810-193 AVEIRO - PORTUGAL
Telefone: 234 – 377900
E-mail: aamaral@av.it.pt

Rute Sofia

IIS, Universität der Bundeswehr München
Munich, Alemanha
E-mail: sofia@informatik.unibw-muenchen.de

10. Referências

- [6BONE] 6BONE
<http://www.6bone.net/>
- [cards] *Placas Wireless PCMCIA*
http://www.linux-wlan.org/docs/wlan_adapters.html
- [cards1] *Placas Wireless pcmcia, drivers para Linux*
<ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/>
- [CIDR] V. Fuller, T. Li, J. Yu, K. Varadhan. *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. IETF RFC 1519, Setembro 1993.

- [DEE99] S. Deering, W. Fenner, B. Haberman. *Multicast Listener Discovery (MLD) for IPv6*. IETF RFC 2710, Outubro 1999.
- [DHCP] R. Droms. *Dynamic Host Configuration Protocol*. Outubro 1993.
- [E2E] J. H. Saltzer, D. P. Reed, D. D. Clark. *End-to-end arguments in system design*. *ACM Transactions on Computer Systems*. Novembro 1984.
- [ERICSSON] Ericsson, *Implementação MIPv6*.
<ftp://ftp.kame.net/pub/kame/contrib/mip6/ericsson>
- [gprsip] Howto para Configuração GPRS.
<http://turtiainen.dna.fi/GPRS-HOWTO>
- [IKE] D. Harkins, D. Carrel. *The Internet Key Exchange (IKE)*. IETF RFC2409, Novembro 1998.
- [IPSec] S. Kent, R. Atkinson. *Security Architecture for the Internet Protocol*. IETF RFC 2401, Novembro 1998.
- [IPv4] Darpa Internet Program, *Internet Protocol Specification*. IETF RFC 759, Setembro 1981.
- [IPv6] S. Deering, R. Hinden. *Internet Protocol, version 6 (IPv6) Specification*. IETF RFC 2460. Dezembro 1998.
- [KAME] *KAME Project*, Janeiro 2004.
<http://www.kame.net>
- [kame1] D. Johnson, C. Perkins, *Mobility Support in IPv6*, IETF Draft (Work in Progress). Novembro 2001.
- [LANC1] Universidade de Lancaster, *Implementação MIPv6*.
<http://www.cs-ipv6.lancs.ac.uk/ipv6/>
- [LANCASTER] *LANCASTER MOBILE IPv6 PACKAGE, 1998*.
<http://www.cs-ipv6.lancs.ac.uk/ipv6/MobileIP>
- [MIP4] IETF, *Charter do Grupo de Mobilidade IPv4 (mip4)*.
<http://www.ietf.org/html.charters/mip4-charter.html>
- [MIP6] IETF, *Charter do Grupo de Mobilidade IPv6 (mip6)*.
<http://www.ietf.org/html.charters/mip6-charter.html>
- [MIPL] *GO-CORE Project, MIPL Mobile IPv6 for Linux*, Janeiro 2004.
<http://www.mipl.mediapoli.com/>
- [MIPv6] D. Johnson, C. Perkins, J. Arkko. *Mobility Support in IPv6*. Internet Draft (Work in Progress). Junho 2003
- [NAT] K. Egevang, P. Francis. *The IP Network Address Translator (NAT)*. IETF RFC 1631, Maio 1994.
- [ND] T. Narten, E. Nordmark, W. Simpson. *Neighbor Discovery for IP Version 6 (IPv6)*. IETF RFC 2461, Dezembro 1998.
- [ngtrans] IETF, *Charter do Grupo NGTRANS*.
<http://www.ietf.org/html.charters/ngtrans-charter.html>
- [NOK1] Nokia, *Introducing Mobile IPv6 in 2G and 3G Mobile Networks*.

- [Nokia] Nokia, *Nokia 7700*
<http://www.nokia.com/phones/7700>
- [pcmciaacs] *pcmcia-cs Wireless Tools*
http://pcmcia-cs.sourceforge.net/ftp/contrib/wireless_tools.25.tar.gz
- [PER96] C. Perkins. *IP Mobility Support*. IETF RFC 2002, Outubro 1996
- [ppp1] UK6XS, Utilização de pppd com GPRS.
<http://www.uk6x.com/networkservices/gprsconf/gprs.tar.gz>
- [ppp2] UK6XS, *Scripts de configuração para ppp/gprs*.
<http://www.uk6x.com/networkservices/gprsconf/>
- [QoS] P. Ferguson, G. Huston. *Quality of Service: Delivering QoS in the Internet and the Corporate Network*. Wiley Computer Books, New York, NY, 1998.
- [radvd] *Código fonte RADVD*.
<http://v6web.litech.org/radvd/dist/radvd-0.7.2.tar.gz>,
- [RFC1256] S. Deering (Editor). *ICMP Router Discovery Messages*. IETF RFC 1256, Setembro 1991.
- [RFC1918] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear. *Address Allocation for Private Internets*. IETF RFC 1918, Fevereiro 1996.
- [RFC2462] S. Thomson, S. Narten. *IPv6 Stateless Address Autoconfiguration*. Dezembro 1998.
- [RFC2765] E. Nordmark, *Stateless IP/ICMP Translation Algorithm (SIIT)*. Fevereiro 2000.
- [RFC2767] K. Tsuchiya, H. Higuchi, Y. Atarashi. *Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)*. Fevereiro 2000.
- [RFC2893] R. Gilligan, E. Nordmark, *Transition Mechanisms for IPv6 Hosts and Routers*. Agosto 2000.
- [RFC3056] B. Carpenter. *Connection of IPv6 Domains via IPv4 Clouds*. Fevereiro 2001
- [RFC3142] J. Hagino. *An IPv6-to-IPv4 Transport Relay Translator*. Junho 2001.
- [RFC3220] C. Perkins, *IP Mobility Support for IPv4*. IETF RFC 3220, Janeiro 2002.
- [RFC3314] M. Wasserman (Editor). *Recommendations for IPv6 in Third Generation Partnership Project (3GPP) Standards*. IETF RFC 3314. Setembro 2002.
- [RFC3316] J. Arkko, G. Kuijpers, H. Soliman, J. Loughney, J. Wiljakka. *Internet Protocol Version 6 (IPv6) for Some Second and Third Generation Cellular Hosts*. Abril 2003.
- [RFC3574] J. Soininen, *Transition Scenarios for 3GPP Networks*. IETF RFC 3574. Agosto 2003.
- [Symbian] *Symbian OS*
<http://www.symbian.com/technology/symbos-v7s-det.html>
- [tcpdump] *Ficheiros tcpdump*.
<http://site.n.ml.org/download/f81d965c6368a7ec9d09fd57d7460319/libpcap/libpcap-0.7.2.tar.gz>
<http://site.n.ml.org/download/09b813e103539331a9f6cbf4ca1195a4/tcpdump/tcpdump-.7.2.tar.gz>
- [UK6X] UK6X homepage.
<http://www.uk6x.com/>
- [UK6Xgprs] <http://www.uk6x.com/networkservices/gprs.html>
- [USAGI] *Projecto USAGI*;
<http://www.linux-ipv6.org/>
- [wavemon] *Código fonte WAVEMON*
<http://www.wavemage.com/wavemon-current.tar.gz>

[WIDE] Wide project, 2000.
<http://www.wide.ad.jp/>