

Construção de uma plataforma SSM para ambientes IPv6

António Trindade, Jorge Sá Silva, Jorge Granjal e Fernando Boavida
trindade@student.dei.uc.pt,
{sasilva,jgranjal,boavida}@dei.uc.pt

Palavras chave: *Multicasting, IPv6, Plataforma, SSM*

Resumo

Este artigo apresenta a plataforma de multicasting para ambientes IPv6 desenvolvida no Laboratório de Comunicações e Telemática da Universidade de Coimbra. Depois da constituição de uma ilha IPv6, e da consecutiva ligação ao 6BONE, foi implementado o Source Specific Multicast (SSM), e foram desenvolvidas aplicações que permitiram avaliar estes ambientes.

O SSM foi proposto para solucionar um conjunto significativo de restrições que os actuais protocolos de encaminhamento multiponto apresentam, principalmente na interligação de diferentes domínios.

1 Introdução

Quando em 1991 surgiram os primeiros *Internet Service Providers* comerciais, o número de utilizadores começou a duplicar duas vezes em cada ano. Foi no início dos anos 90, altura em que a Internet se tornou objecto de consumo, que se começaram a adivinhar os problemas que iriam surgir alguns anos mais tarde. Os endereços estavam a ser atribuídos a um ritmo incomportável. As previsões mais optimistas apontavam o ano 2005 como o ano em que seria atribuído o último endereço IPv4 [RFC0791].

Para travar esse crescimento foram introduzidas várias soluções baseadas no *Network Address Translation* (NAT, especificado em [RFC3022]) e no *Classless Inter-Domain Routing* (CIDR) [RFC1519]. Enquanto as técnicas baseadas no NAT possibilitavam a coexistência de políticas de endereçamento privado com os endereços públicos disponíveis, o CIDR acabava com a divisão em classes. A divisão entre endereço de rede e endereço do *host* seria completamente flexível. Além do mais, o CIDR permitia que várias redes

pudessem ser anunciadas pelo router externo como uma só, o que reduziria o tamanho das tabelas de encaminhamento.

O IPv6 ([RFC2460]) foi desenvolvido tendo como principal objectivo ultrapassar a escassez de endereços apresentada pelo IPv4. No entanto, com o objectivo de dotar a Internet das funcionalidades exigidas pelos serviços actuais, e de oferecer a flexibilidade para as exigências futuras, foram ainda propostas outras funcionalidades no IPv6.

O tamanho dos endereços acabou por ser fixado em 128 *bits*. Este número é, segundo muitos, um compromisso entre a capacidade de expansão futura e a economia de espaço na memória dos equipamentos que vão manipular estes endereços, bem como no *overhead* do próprio protocolo.

Até hoje, o *multicast* não mereceu grande atenção comercial, principalmente devido à dificuldade de configuração dos equipamentos e à relativa falta de aplicações economicamente interessantes. Contudo, nos dias de hoje, com o crescimento da Internet e com a respectiva adesão por todos os sectores sociais, o cenário alterou-se. A distribuição de sinais de vídeo, o tele-trabalho corporativo, o ensino à distância e as simulações distribuídas são apenas alguns exemplos de aplicações que não podem ser implementadas pelos tradicionais métodos baseados em ligações ponto-por-ponto. Para comunicações com um número muito elevado de participantes a emulação de sistemas multiponto por um conjunto de várias ligações ponto-por-ponto obrigaria a um desperdício inoportável de largura de banda e de processamento.

Embora a dificuldade de configuração dos equipamentos seja outra das razões para o atraso verificado no aparecimento dos sistemas multiponto nativos, os respectivos protocolos de encaminhamento também apresentam níveis de complexidade elevada. Estas dificuldades acentuam-se para grupos *multicast* com participantes em vários domínios administrativos e quando as estações terminais apresentam comportamentos dinâmicos, isto é, quando podem entrar ou sair de um grupo *multicast* durante a sessão. Este tipo de actividade obriga a refazer toda a árvore de distribuição *multicast*. Por contraste, o encaminhamento *unicast* é praticamente estático uma vez que apenas ocorrem alterações quando uma ligação física falha ou é substituída.

O SSM ([Thaler02]) foi desenvolvido para solucionar a complexidade presentes nos actuais protocolos de encaminhamento *multicast*. Segundo este protocolo, sempre que uma estação terminal se inscreve num determinado grupo *multicast*, para além de especificar o respectivo endereço como acontecia com os anteriores protocolos, deverá especificar o(s) endereço(s) da(s) estação(ões) emissora(s) da(s) qual(is) aceita receber informação. Este procedimento apresenta duas vantagens significativas: para além de facilitar toda a construção da árvore de distribuição *multicast* uma vez que as mensagens de controlo são directamente encaminhadas para os nós emissores, possibilita ainda a implementação de mecanismos de filtragem. Esta última propriedade irá ser importante pois permitirá eliminar o tradicional modelo *any-to-any*, que possibilita que uma estação ao inscrever-se num grupo *multicast* esteja sujeita a receber informação de

fontes não desejáveis.

A simplificação dos mecanismos de encaminhamento permitirão ultrapassar a difusão dos nós com funções de *rendez-vous point*. A difusão da sua localização requer complexos protocolos adicionais na interligação de grupos *multicast* distribuídos por diferentes domínios administrativos.

Os endereços reservados para SSM são um subconjunto dos endereços *multicast*. No caso do IPv4, o intervalo é de 232.0.0.0 a 233.255.255.255. O bloco 232.0.0.0/8 é utilizado para endereços reservados pela IANA, enquanto que a utilização do bloco 233.0.0.0/8 é livre. Em IPv6, o bloco reservado pela IANA é o `ff20::/12` e o de uso geral é o `ff30::/12`.

Tal como para qualquer ligação *multicast* os membros de um qualquer grupo deverão efectuar a subscrição através dos protocolos *Internet Group Management Protocol* (IGMP). No entanto, estas funcionalidades só estão presentes na versão 3 do IGMP ([RFC3376]) para o IPv4, ou através do *Multicast Listener Discovery* (MLD) versão 2 — [Vida02] — para o IPv6. O IGMPv3 já atingiu o estado de "Draft Standard" do IETF, tendo sido publicado no RFC3376. O MLDv2 ainda se encontra sob a forma de *Internet-Draft*.

O IPv4 não inclui de forma nativa o suporte *multicast*. Com a importância crescente dos sistemas *multicast* estas funcionalidades já se encontram presentes no IPv6. Aliás, as funcionalidades *multicast* são utilizadas pelos mecanismos internos do IPv6, como acontece nos processos de auto-configuração de endereços e na resolução de endereços IPv6 em endereços *hardware*, substituindo o uso do *broadcast*, que não existe em IPv6.

No âmbito do processo de avaliação dos mecanismos SSM em várias plataformas e da proposta de novos modelos para comunicações avançadas, o Laboratório de Comunicações e Telemática está a desenvolver um conjunto de projectos integrados.

Os projectos em curso recorrem, essencialmente, a duas metodologias complementares: a simulação e a prototipagem. Através da simulação possibilita-se o estudo de cenários impossíveis de avaliar em plataformas reais, devido ou à sua complexidade ou a questões de natureza financeira.

A plataforma real, que se encontra em implementação, possibilita a avaliação dos novos modelos propostos e o contacto real com os protocolos e os problemas associados. Esta plataforma apresenta um conjunto de sub-redes, em ambientes *wireless* e *wired*, que por sua vez ainda se encontram ligadas pelo 6BONE a outros parceiros nacionais e internacionais.

A secção seguinte apresenta, em detalhe, uma destas sub-redes e descreve o processo de construção do ambiente de estudo. Por forma a avaliar a plataforma em causa, é descrito na secção 4 o desenvolvimento de um Chat com suporte SSM. A secção final apresenta algumas conclusões e descreve o trabalho futuro.

2 Construção de uma plataforma experimental

A plataforma IPv6 apresentada neste trabalho é composta por quatro computadores baseados em processadores Intel (figura 1), nas quais foi inicialmente instalado o sistema operativo FreeBSD, versão 4.7. Uma destas máquinas, com um processador Intel Pentium III a 733MHz, com 256MB de RAM e dois interfaces de rede, desempenha o papel de router IPv6 do Departamento de Engenharia Informática para o exterior (6BONE). Das restantes máquinas, com processadores Intel Celeron a 1.7GHz e com 512 MB de RAM, duas desempenham o papel de sistemas terminais e outra de router interno. As placas de rede escolhidas foram 3COM 3c905cx.

No início deste trabalho foi ainda testada a utilização de um router Cisco 2501, a correr uma versão com suporte para IPv6 experimental do sistema IOS 11.0. Dado que não foi possível instalar neste *router* uma versão posterior do IOS, com suporte para os protocolos que seriam utilizados na plataforma, devido a limitações de *hardware* do próprio equipamento, este foi abandonado pouco tempo depois.

Posteriormente, foi adicionada à plataforma um quinto computador com o Windows XP Pro que permitiu proceder à análise das capacidades deste sistema operativo para ambientes IPv6.

Dado que o SSM é um protocolo muito recente e para o qual não existe suporte em nenhuma das versões oficiais do FreeBSD ([FBSDB]), foi utilizada uma versão em desenvolvimento do *kernel*, preparada no âmbito do projecto KAME ([KAME]). De facto, um dos principais objectivos para a criação deste ambiente é poder contribuir no processo de avaliação do SSM, para o qual ainda existem muito poucas plataformas experimentais, principalmente para ambientes IPv6.

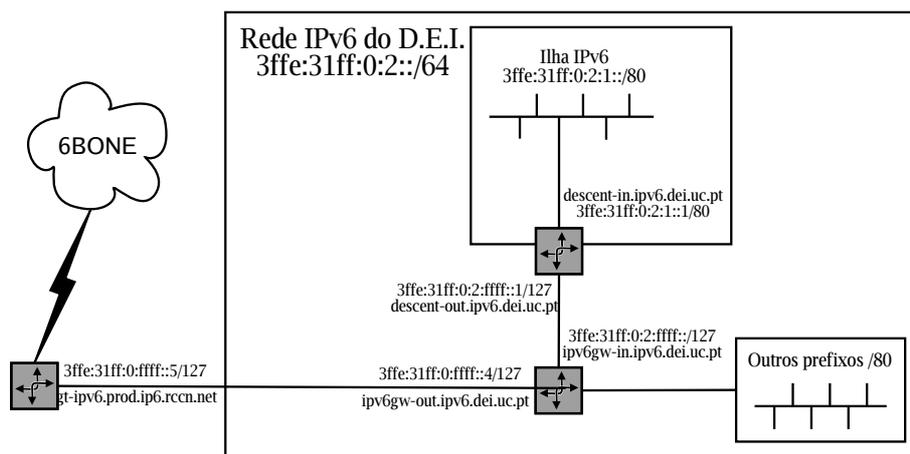


Figura 1: Esquema da rede de teste

O projecto KAME é um projecto conjunto de várias empresas japonesas que visa criar uma plataforma de referência para IPv6, analogamente ao que foi feito aquando do

aparecimento do IPv4 no sistema 4.3BSD, no final da década de 1970. Este projecto actualmente desenvolve a pilha protocolar de IPv6 para FreeBSD 4.8 e 5.0, NetBSD 1.6.1, OpenBSD 3.3 e Darwin (sistema operativo que serve como base ao MacOS/X).

A tabela 1, mostrada a seguir, resume a configuração dos computadores utilizados na plataforma, bem como os serviços de rede disponibilizados.

Nome da máquina	Sistema Operativo	Endereço IPv4	Endereço IPv6	Serviços
ipv6-gw-out.ipv6.dei.uc.pt	FreeBSD 4.8	193.136.212.222	3ffe:31ff:0:ffff::5/127	Router IPv6, SSH, SNMP
ipv6-gw-in.ipv6.dei.uc.pt	FreeBSD 4.8	10.3.0.56	3ffe:31ff:0:2:ffff::1/127	(IPv4), DNS e <i>firewall</i>
descent-in.ipv6.dei.uc.pt	FreeBSD 4.8	10.3.0.127	3ffe:31ff:0:2:1::1/80	Router IPv6, SSH e FTP
descent-out.ipv6.dei.uc.pt	FreeBSD 4.8	10.3.0.127	3ffe:31ff:0:2:ffff::1/127	
unreal.ipv6.dei.uc.pt	FreeBSD 4.8	10.3.0.127	3ffe:31ff:0:2:1::2/80	HTTP, FTP, SSH
hexen.ipv6.dei.uc.pt	Windows XP Pro	10.3.0.57	3ffe:31ff:0:2:1::3/80	FTP, SSH
doom.ipv6.dei.uc.pt	FreeBSD 4.8	10.3.0.55	3ffe:31ff:0:2:1::4/80	FTP, SSH

Tabela 1: Sumário dos endereços e serviços da plataforma

No decorrer da preparação do ambiente de trabalho foi necessário mudar posteriormente para a versão 4.8 do FreeBSD, pois essa é, neste momento, a versão de referência do projecto KAME. A versão oficial da pilha em todos estes sistemas é também desenvolvida no âmbito deste projecto, embora muito mais antiga e não contendo suporte para protocolos experimentais, como é o caso do SSM.

Os computadores foram ainda configurados com endereços IPv4, atribuídos pela administração da rede do D.E.I, apenas para suportar a utilização de aplicações que ainda não funcionam com IPv6, tal como X Windows e SNMP.

Foram instalados e configurados alguns serviços de rede com suporte para IPv6, nomeadamente o DNS, o FTP, o SSH e o SMTP. Este último, embora configurado, destina-se apenas ao processamento de correio interno a cada uma das máquinas. Posteriormente, através de um processo que se prevê de relativa simplicidade, poderá ser configurado para aceitar correio externo.

O servidor de DNS encontra-se a correr no *router* IPv6 do DEI. Este servidor contém os mapas de todas as redes IPv6 do DEI, bem como o registo IPv4 de todos os computadores da plataforma.

Na plataforma existe ainda um computador com o servidor HTTP Apache, versão 2.0.45, configurado para responder tanto por IPv4 como por IPv6. Este computador é também responsável por colectar, através de SNMP (em IPv4, pois as ferramentas utilizadas para este efeito não suportam ainda IPv6), e armazenar estatísticas de tráfego do router do D.E.I.

O prefixo IPv6 atribuído ao D.E.I. é 3ffe:31ff:0:2::/64. Este prefixo teve de ser dividido em vários sub-prefixos todos com 80 bits, de modo a comportar os diferentes projectos IPv6 no âmbito do Laboratório de Comunicações e Telemática. Destes prefixos, apenas o prefixo 3ffe:31ff:0:2:ffff::/80 foi reservado para uso exclusivo do administrador do router IPv6. Mas o uso de prefixos de 80 bits trouxe alguns problemas, que serão descritos mais à frente.

Posteriormente, um destes prefixos foi ainda sub-dividido em prefixos de 127 bits para

ligações ponto-a-ponto entre os diferentes routers das novas sub-redes que possam vir a ser preparadas no futuro. Para este trabalho foi utilizado o prefixo `3ffe:31ff:0:2:1::/80`. Os restantes prefixos (`3ffe:31ff:0:2:0::/80` a `3ffe:31ff:0:2:ffff::/80`) serão atribuídos a essas sub-redes.

O router IPv6 do DEI é o router de fronteira do departamento. É este sistema que gere a ligação com o 6BONE, a rede de testes IPv6 mundial. Como o *backbone* da Universidade de Coimbra ainda não suporta IPv6, a ligação ao 6BONE é feita através de um túnel *6over4*, onde os pacotes IPv6 são encapsulados em pacotes IPv4 e enviados através da rede como se realmente tratassem de pacotes IPv4. Chegados à outra extremidade do túnel, são desencapsulados e possivelmente enviados através de um *backbone* nativo IPv6 ou através de um outro túnel *6over4*.

Embora seja lançada semanalmente uma nova versão do *kit* KAME, optou-se por efectuar a sua transferência através do protocolo CVSUP (*CVS Update Protocol*).

```
00100 allow ipv6 from any to any via lo0
03700 allow tcp from any to any established
03800 allow ipv6 from any to any frag
03900 allow tcp from any to any 25 setup
04000 allow tcp from any to any 22 setup
04100 allow tcp from any to any 20-21 setup
04200 allow ipv6 from 3ffe:31ff:0:2::/64 to any
04300 allow ipv6 from 3ffe:31ff:0:ffff::5 to any
04400 allow tcp from any to 3ffe:31ff:0:ffff::5 53 setup
04500 allow udp from any to 3ffe:31ff:0:ffff::5 53
04600 allow udp from 3ffe:31ff:0:ffff::5 53 to any
04700 allow tcp from any to any 80 setup
04800 deny log tcp from any to any in recv gif0 setup
04900 allow tcp from any to any setup
05000 allow udp from any 53 to 3ffe:31ff:0:ffff::5
05100 allow udp from 3ffe:31ff:0:ffff::5 to any 53
05200 allow udp from any 123 to 3ffe:31ff:0:ffff::5
05300 allow udp from 3ffe:31ff:0:ffff::5 to any 123
05400 allow ipv6-icmp from any to any icmp type 1
05500 allow ipv6-icmp from any to any icmp type 2,135,136
65535 deny ipv6 from any to any
```

Figura 2: Regras do *firewall* IPv6 do D.E.I.

O *firewall* encontra-se implementado recorrendo ao módulo IP6FW. Actualmente, existem dois *firewalls* diferentes para o FreeBSD 4, o IPFW, e o IPFilter. Este último, é compatível com várias arquitecturas Unix, nomeadamente SunOS, IRIX e os derivados do BSD.

No presente momento, as regras existentes limitam as ligações de fora aos serviços DNS, SSH, SMTP, FTP e HTTP. Qualquer outra tentativa de ligação é negada pelo *firewall* (ver figura 2). Foram omitidas desta lista as regras referentes a protecções contra *spoofing*, bem como regras que impedem o uso de mapeamento *6to4* e transmissão de endereços *site-local* (*unicast* e *multicast*) para o exterior.

Foi também instalado num dos computadores o servidor HTTP Apache, na sua versão de desenvolvimento 2.0.45, mais tarde mudado para a versão 2.0.46, devido a problemas de segurança surgidos na versão anterior. A opção por uma versão em desenvolvimento foi devida a apenas esta conter suporte nativo para IPv6. A actual versão de produção - 1.3.27 - não apresenta esse nível de suporte. De qualquer forma, o Apache Group já não acrescenta novas funcionalidades nesta família de versões. Este desenvolvimento é apenas realizado nas versões 2.0.x. No entanto, dos estudos elaborados no decorrer deste projecto, esta versão do Apache revelou-se bastante estável para ser usada em servidores de produção.

O servidor foi configurado para responder tanto a pedidos IPv4 como IPv6. No entanto, de fora da rede departamental, este serviço está apenas acessível por IPv6 devido às restrições impostas pelo *firewall* IPv4 do Departamento.

Este computador foi ainda preparado para colectar estatísticas de tráfego no router de saída, através do protocolo SNMP, utilizando o programa MRTG. Foram efectuadas várias medições do tempo de ida e volta (medido com o utilitário `ping6`) entre este computador e alguns pré-seleccionados, situados em Lisboa, Guarda, e também na Alemanha, Japão e Estados Unidos da América.

A figura 3 apresenta os resultados de uma experiência efectuada entre a plataforma IPv6 e uma máquina situada no Japão (`www.kame.net`).

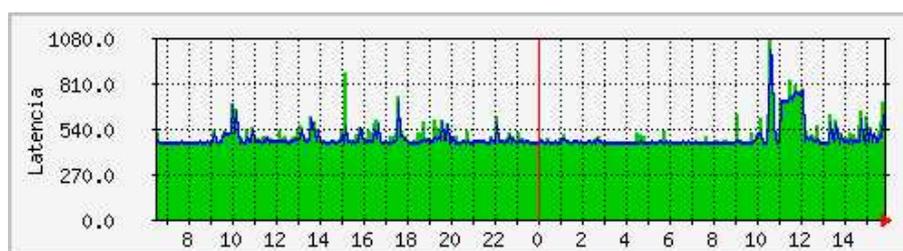


Figura 3: Medição do tempo de ida e volta (`ping6`) medido com o MRTG

Os dois routers da plataforma foram configurados para que a informação de encaminhamento fosse comunicada através do protocolo RIPv6, que resulta da adaptação do conhecido protocolo RIP (*Routing Information Protocol*) ao IPv6.

O IPv6 é um dos protocolos implementados no daemon MRT (*Multi-threaded Routing Toolkit*) e a sua escolha para este projecto resultou da simplicidade da sua sintaxe de configuração, que é em tudo semelhante à sintaxe utilizada pelos routers Cisco. No

entanto, para além do RIP, o MRT inclui ainda suporte para *Open Shortest Path First* (OSPF), o *Border Gateway Protocol* (BGP4+), e ainda o IS-IS.

A figura seguinte apresenta parte do ficheiro geral de configuração do FreeBSD, no que respeita à configuração IPv6, do router de fronteira do D.E.I:

```
gateway_enable="YES"
gif_interfaces="gif0"
gifconfig_gif0="193.136.212.221 193.136.2.246"
hostname="ipv6-gw.dei.uc.pt"
ipv6_firewall_type="simple"
ifconfig_xl1="inet 193.136.212.221 netmask 255.255.255.224"
ipv6_enable="YES"
ipv6_ifconfig_xl0="3ffe:31ff:0:2:ffff::0/127"
ipv6_ifconfig_gif0="3ffe:31ff:0:ffff::5/127"
ipv6_defaultrouter="3ffe:31ff:0:ffff::4"
ipv6_gateway_enable="YES"
router_enable="YES"
router="/usr/local/sbin/mrtd"
```

Figura 4: Ficheiro de configuração do router de saída

O interface de rede `gif0` é o interface virtual utilizado pelo sistema para estabelecer o túnel *6over4*, e o interface `xl1` é o interno.

3 Instalação do módulo SSM

Como foi referido acima, foi utilizado o sistema operativo FreeBSD, versão 4.7, que mais tarde foi actualizado para a versão 4.8.

Tanto o processo de instalação do próprio FreeBSD como o de actualização por CVS e *upgrade* encontram-se bem documentados em [FBSD], pelo que não serão descritos aqui. Também se lá pode encontrar informação suficientemente detalhada sobre a configuração básica IPv6 e ligação ao 6BONE.

O *kit* KAME é composto pelos ficheiros em linguagem C que constituem a implementação das pilhas protocolares IPv6 e IPv4, bem como os utilitários de configuração dos sistema e diagnóstico que delas dependem. Podem-se obter versões *snapshot*, todas as segundas-feiras, através do servidor FTP do projecto KAME (`ftp.kame.net`, acessível por IPv4 e IPv6). Este *kernel* contém suporte para protocolos ainda experimentais, incluindo alguns ainda não completamente especificados, como é o caso do SSM. Além do FTP, é possível obter também o código fonte através de CVS anónimo ou CVSUP, sendo este método o aconselhado em FreeBSD. Estes últimos permitem

obter sempre as últimas alterações a qualquer dia, sem ter que se esperar pelo lançamento do *snapshot* seguinte.

A instalação do módulo **SSM** não se apresenta trivial. Primeiro é necessário efectuar uma cópia de segurança do directório `/usr/include`. Se já existir uma versão mais antiga do *kit* KAME instalado, deve-se também apagar o directório `/usr/local/v6`.

Depois de ter o código fonte no disco, deve-se, no directório base do *kit* (mostrado nos exemplos seguintes como `~/kame`), deve-se executar o comando “`make TARGET=freebsd4 prepare`”. A palavra a *bold* deverá ser substituída pela correspondente ao sistema operativo que se está a utilizar (neste caso é mesmo `freebsd4`). Este comando cria *links* simbólicos para ficheiros na árvore do código fonte do próprio *kernel* do FreeBSD, localizado em `/usr/src/sys`.

De seguida deve-se editar o ficheiro de configuração do *kernel*, que está no directório `~/kame/freebsd4/sys/i386/conf`. Pode-se copiar o ficheiro `GENERIC.KAME` para um outro e editar este último (usualmente, utiliza-se como nome deste ficheiro o nome do computador, mas todo em maiúsculas). Visto isto, copiou-se o primeiro ficheiro para um outro chamado `IPV6-GW`. Para que houvesse no *kernel* suporte para **SSM**, túneis *6over4* e *firewall* foram acrescentadas as seguintes linhas:

```
options MLDV2 # SSM em IPv6
options IPV6FIREWALL
options IPV6FIREWALL_VERBOSE
pseudo-device gif
```

O próximo passo consiste em criar os *makefiles* para a compilação do *kernel* e instalá-lo com os seguintes comandos:

```
~/kame/freebsd4/sys/i386/conf$ config IPV6-GW
~/kame/freebsd4/sys/i386/conf$ cd ../../compile/IPV6-GW
~/kame/freebsd4/sys/compile/IPV6-GW$ make depend
~/kame/freebsd4/sys/compile/IPV6-GW$ make install
```

De seguida, compilam-se e instalam-se as ferramentas de configuração e diagnóstico:

```
~/kame/freebsd4$ make
~/kame/freebsd4$ make install
```

Depois disto ter-se-á as ferramentas instaladas no directório `/usr/local/v6`. Antes fazer *reboot* para que o novo *kernel* entre em funcionamento, dever-se-á editar o ficheiro

`/etc/rc`, por forma a que os directórios `/usr/local/v6/bin` e `/usr/local/v6/sbin` apareçam na variável `PATH` antes dos `/bin` e `/sbin`, respectivamente. Só assim se garante que, no arranque do sistema, sejam utilizadas as ferramentas de configuração (`ifconfig`, `route`, etc...) que são fornecidas com o *kit* KAME e não as que são distribuídas com o sistema.

Depois do *reboot* foram instalados os serviços acima especificados. FTP e SSH são distribuídos com o próprio sistema operativo, pelo que foi apenas necessário activá-los no ficheiro `/etc/rc.conf`. Os outros serviços (BIND 9, Apache 2.0.46, MRT, MRTG e SNMP) foram instalados utilizando os *ports* do FreeBSD. Os directórios onde estão estes serviços são, respectivamente: `/usr/ports/net/bind9`, `/usr/ports/www/apache2`, `/usr/ports/net/mrtg`, `/usr/ports/net/mrt` e `/usr/ports/net/net-snmp4`. A instalação destas aplicações é feita como todos os outros *ports*, isto é, utilizando o comando `make install`. Os únicos serviços que necessitam de algumas alterações à configuração são o BIND, no qual foi preciso configurar os mapas de DNS (figura 5, o Apache, no qual foi mudado o directório base, além de outras configurações, e ainda o SNMP que foi configurado por forma a apenas poder ser acedido a partir da rede da plataforma.

```

$TTL 43200
$ORIGIN ipv6.dei.uc.pt.
@      IN      SOA      ipv6.dei.uc.pt. hostmaster.ipv6-gw-out.ipv6.dei.uc.pt. (
                2003060501      ; Serial (yymmddxx)
                10800           ; Refresh 3 hours
                3600            ; Retry 1 hour
                3600000         ; Expire 1000 hours
                86400 )         ; Minimum 24 hours

                                IN      NS      ipv6-gw-out.ipv6.dei.uc.pt.

ipv6-gw-out      IN      AAAA     3ffe:31ff:0000:ffff:0000:0000:0000:0005
ipv6-gw-out      IN      A       193.136.212.221
ipv6-gw-in       IN      AAAA     3ffe:31ff:0000:0002:ffff:0000:0000:0000
descent-out      IN      AAAA     3ffe:31ff:0000:0002:ffff:0000:0000:0001

```

Figura 5: Extracto do mapa directo de DNS

4 Desenvolvimento de aplicações e avaliação

Dada a natureza experimental do SSM, que se encontra mais acentuada para os ambientes IPv6, foi necessário iniciar o desenvolvimento de aplicações de teste.

Mesmo para o *multicasting* em IPv6 sem suporte SSM, o conjunto de aplicações de teste é limitado, principalmente se for pretendido a avaliação de cenários envolvendo vários domínios administrativos. Assim, procedeu-se ao desenvolvimento de uma aplicação de Chat *multicast* com suporte SSM (mais precisamente SFM—*Source Filtered Multicast*). Como ponto de partida foi utilizado um programa de exemplo que

é fornecido com o *kit* KAME, chamado *mchat*. Como se pode facilmente depreender do nome, esta é uma aplicação de *chat* que comunica através de *multicast* (mas sem suporte SSM).

No decorrer deste projecto foram feitas alterações por forma adicionar à aplicação capacidades SSM. Este trabalho levantou diversas dificuldades, principalmente devido à falta de documentação e à imaturidade da implementação. O *interface* de programação IPv6 unicast esteja descrito em [RFC2133] e a *interface* de programação *multicast* encontra-se documentada em *Internet-draft* ([Thaler02]), apenas com descrições informativas, pelo que poderão existir diferenças significativas entre o que está escrito nestes documentos e o que se encontra de facto implementado no sistema operativo. Por outro lado, como o protocolo MLDv2 ([Vida02]) ainda não atingiu o estágio de *Standard*, é extremamente difícil encontrar implementações isentas de erros. Mesmo os responsáveis do projecto KAME apenas deram como terminado o trabalho de implementação do SSM em Fevereiro deste ano. Eles próprios, depois de vários contactos, admitiram que é necessário um processo mais intenso de *debugging*.

Algumas das dificuldades foram ultrapassadas pela análise do código fonte do próprio kernel e das bibliotecas do *kit* KAME. Outras dificuldades ainda persistem devido a possíveis *bugs* no código da implementação ou em funcionalidades ainda não implementadas no próprio sistema.

Embora o MLDv2 preveja a utilização de filtros *multicast* com vários endereços (esta possibilidade encontra-se descrita na especificação e é intitulada SFM (*Source Filtered Multicast*)), não foi possível conseguir que a aplicação funcionasse de acordo com o previsto, apesar da depuração do código para determinar se o problema estava na aplicação ou no próprio *kernel*.

Um dos objectivos do presente projecto foi avaliar o suporte IPv6 do Windows XP. Este foi o primeiro sistema Windows com suporte nativo para IPv6. No entanto, ainda não é uma implementação com qualidade comercial.

O Microsoft Windows 2003 Server, e o Service Pack 1 para o Windows XP serão os primeiros produtos desta empresa com suporte IPv6 suportado comercialmente. Na versão testada, a configuração da pilha protocolar é feita inteiramente na linha de comando, utilizando um utilitário chamado `ipv6.exe`. Sempre que o sistema arranca é necessário proceder à reconfiguração completa, pois o sistema não permite armazenar os parâmetros de configuração.

Segundo a própria Microsoft o suporte disponibilizado é apenas uma implementação de teste e não serve para uso de produção. Quanto ao suporte SSM em IPv6, este está completamente ausente da implementação testada. No entanto, o sistema suporta a resolução de nomes que devolvam endereços IPv6, embora a consulta ao DNS seja feita por IPv4. O Internet Explorer funcionou também por IPv6, não tendo sido observadas diferenças no funcionamento em IPv4 ou em IPv6.

Foi ainda instalada em vários computadores da plataforma a aplicação VIC, com *patches*

que lhe permitem suportar sessões SSM. Isto permitiu fazer várias experiências com uma aplicação de video-conferência com suporte SSM. Estas experiências foram apenas feitas na rede IPv6, sem o recurso a *routers*; os testes com *routers* poderão constituir trabalho futuro. Ao contrário do *chat*, esta aplicação apenas suporta SSM (e não SFM) e o seu comportamento foi o esperado não havendo problemas de maior a apontar.

5 Conclusões

Neste artigo foi descrita a implementação de uma plataforma IPv6 *multicast* para ambientes SSM, desenvolvida no âmbito de um projecto do Laboratório de Comunicações e Telemática da Universidade de Coimbra.

A plataforma encontra-se já ligada a outra plataforma IPv6 *wireless* com suporte *multicast*. No entanto, como trabalho futuro, pretende-se estender o ambiente SSM da plataforma descrita no artigo aos ambientes *wireless* e avaliar vários modelos para IP Móvel com multiponto, desenvolvidos no âmbito de outros projectos.

Os problemas encontrados no desenvolvimento do *chat* não foram resolvidos. Tal como admitem os próprios responsáveis do projecto KAME, é necessário bastante mais *debugging* do código antes de se poder considerar uma implementação estável. Como o funcionamento da aplicação está de acordo com o descrito na especificação, apenas se pode supor que a implementação ainda não suporta completamente as funcionalidades requeridas.

Referências

- [RFC2460] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, IETF, December 1998.
- [RFC1519] V. Fuller, T. Li, J. Yu, and K. Varadhan, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*, RFC 1519, IETF, September 1993.
- [RFC2133] R. Gilligan, S. Thomson, J. Bound, and W. Stevens, *Basic Socket Interface Extensions for IPv6*, RFC 2133, IETF, April 1997.
- [KAME] *The KAME Project*, <http://www.kame.net/>.
- [RFC0791] J. Postel, *Internet Protocol*, RFC 0791, IETF, September 1981.
- [FBSD] The FreeBSD Documentation Project, *The FreeBSD Handbook*, 2nd ed., no. 1-57176-303-1, WindRiver, Novembro 2002.

- [RFC3022] P. Srisuresh and K. Egevang, *Traditional IP Network Address Translator (Traditional NAT)*, RFC 3022, IETF, January 2001.
- [RFC3376] A. Thyagarajan, B. Cain, B. Fenner, I. Kouvelas, and S. Deering, *Internet Group Management Protocol, Version 3*, RFC 3376, IETF, October 2002.
- [Thaler02] Dave Thaler, Bill Fenner, and Bob Quinn, *Socket Interface Extensions for Multicast Source Filters*, DRAFT-IETF-MAGMA-MSF-API-03.TXT, IETF, June 2002.
- [Vida02] R. Vida, L. Costa, S. Fdida, S. Deering, B. Fenner, I. Kouvelas, and B. Haberman, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*, DRAFT-VIDA-MLD-v2-07.TXT, IETF, June 2002.