

MEDIÇÃO DE DESEMPENHO DE UMA PLATAFORMA IPV6 SOBRE ATM

Nuno Veiga¹, Jorge Sá Silva², Sérgio Duarte³ e Fernando Boavida⁴

Departamento de Engenharia Informática, Universidade de Coimbra - Pólo II

Laboratório de Comunicações e Serviços Telemáticos, 3030 COIMBRA

Tel.: +351-239-790000, Fax: +351-239-701266

¹Escola Superior Tecnologia e Gestão - Instituto Politécnico de Leiria {nveiga@estg.iplei.pt}

²Grupo de Comunicações - Universidade de Trás-os-Montes e Alto Douro {jsilva@utad.pt}

³Instituto Politécnico da Guarda {sduarte@ipg.pt}

⁴Departamento de Engenharia Informática - Universidade de Coimbra {boavida@dei.uc.pt}

Palavras chave: implementação, testes, IPv6 sobre ATM, MARS, *multicast*, FreeBSD.

Resumo

Este artigo apresenta testes e resultados relativos à plataforma prática do Internet Protocol version 6 (IPv6) sobre Asynchronous Transfer Mode (ATM), construída no âmbito do projecto Mars Extensions Development for IPv6 over ATM (MEDIA), tendo em vista a quantificação do seu desempenho.

A plataforma tem como objectivo principal permitir o estudo e a avaliação das tecnologias envolvidas, bem como o desenvolvimento de propostas para o suporte de multicast com QoS em ambientes IPv6 sobre ATM.

No âmbito do projecto MEDIA, está ainda em desenvolvimento um simulador específico para IPv6 sobre ATM – SIANET. Neste contexto a plataforma foi também utilizada para os medir tempos necessários para o simulador e ainda como validação do mesmo.

I. Introdução

O actual crescimento da Internet constitui uma oportunidade para o ATM. A utilização de aplicações multimédia na Internet é generalizada. Oferecendo, além dos serviços de dados, aplicações de voz e vídeo em tempo real, a Internet está a entrar no mercado típico do ATM ao nível dos serviços. Além disso, de modo a melhorar o suporte para as

aplicações multimédia e para os fornecedores de serviços, surgiu na Internet a necessidade de suporte para reserva de recursos e para cobrança.

O rápido crescimento da Internet e a necessidade de suporte de serviços multimédia sensíveis ao atraso levou ao desenvolvimento de uma nova versão do protocolo IP, o IPv6, projectado para ultrapassar as limitações da versão corrente, o IPv4, e para responder às necessidades dos actuais e dos futuros serviços da Internet.

A principal diferença entre o IPv4 e o IPv6 na adaptação a redes ATM, segundo os *overlay-models*, resulta da resolução de endereços da camada 3 para endereços da camada 2 ser efectuada pelo próprio IPv6. No entanto, é assumido pelo IPv6 que a camada inferior oferece um sistema *broadcast connectionless*, necessário ao protocolo de Neighbor Discovery. Como o ATM não oferece de forma nativa este meio é necessário emulá-lo. A solução encontrada resulta na introdução de uma subcamada entre o IPv6 e o ATM que capte todas as mensagens *broadcast* e as envie para um servidor onde será efectuada a sua distribuição a todos os elementos do *cluster*.

Na próxima secção apresenta-se um resumo das propostas para a integração do IPv4 sobre o ATM, bem como uma descrição resumida da solução MARS. Na secção III descrevem-se as especificações existentes para a integração do IPv6 sobre o ATM. O projecto MEDIA é apresentado na secção seguinte que inclui também uma breve descrição do simulador SIANET. Na secção V descreve-se a plataforma prática implementada. A secção VI finaliza o artigo com uma descrição dos testes, medições e cálculos de tempos efectuados.

II. O IPv4 sobre o ATM

A. O IPv4 unicast sobre o ATM

A interacção entre os protocolos IP e ATM é actualmente um ponto central de interesse nas redes. O IP é um protocolo clássico, orientado para dados, não orientado à ligação, que provou a sua grande utilidade sendo a base da Internet. O ATM é um protocolo orientado à ligação, capaz de garantir QoS, e complexo, que tem amadurecido de forma gradual durante a última década. A interacção IP/ATM é muito mais do que um mero encapsulamento de pacotes IP em células ATM, colocando-se muitos problemas arquitecturais e funcionais. A emulação de uma rede IP com mecanismos ATM significa emular uma tecnologia pior com uma tecnologia melhor e mais flexível. A importância da interacção IP/ATM pode comprovar-se pelo número de soluções existentes, desde o *Classical IP over ATM* (CLIP) até ao *LAN Emulation* (LANE), *MultiProtocol Over ATM* (MPOA), *Multicast Address Resolution Server* (MARS), *Next Hop Resolution Protocol* (NHRP).

Os protocolos de *LAN Emulation* [4] inserem-se nos *overlay-models*. O LANE é uma especificação do ATM Forum que oferece uma solução para correr os protocolos de LAN predominantes (principalmente o *Ethernet*) de forma transparente sobre a rede ATM.

O LANE emula uma LAN sobre uma rede ATM, oferecendo à camada de rede um interface de serviços idêntico ao das LANs existentes (como *Ethernet* ou *Token-Ring*,

normas IEEE 802.3 e IEEE 802.5, respectivamente) e envia os dados através da rede ATM usando encapsulamento MAC LAN apropriado. Em suma, o LANE faz uma rede ATM parecer e comportar-se como, por exemplo, uma rede local *Ethernet*.

Outra solução para correr redes IP sobre redes ATM é o *Classical IP over ATM (CLIP)*, especificado pelo grupo IP-Over-ATM do IETF [6] (RFC 2225, antigo RFC 1577). O protocolo CLIP é muito mais simples que o LANE, pois neste último efectuam-se dois processos de conversão e as propriedades ATM aparecem escondidas [11].

No modelo *classical IP*, as máquinas de uma rede estão organizadas em sub-redes, *Logical IP Subnetwork (LIS)*, partilhando um prefixo de endereço IP comum. Para a resolução de endereços IP para endereços ATM é utilizado o protocolo *ATM Address Resolution Protocol (ATMARP)* e a comunicação entre sub-redes faz-se através de *routers*.

O *Next Hop Resolution Protocol (NHRP)* [8] do IETF, define o suporte do IP não apenas sobre o ATM, mas sobre todo o tipo de redes *Non-Broadcast Multiple-Access (NBMA)*, tais como o ATM, *Frame Relay*, X.25.

Em 1995 surgiu o conceito de comutação IP com equipamentos que desempenham o papel ora de *router* ora de comutador, pelo que são designados por *router-comutador*. Várias soluções foram desenvolvidas e comercializadas tais como o *Cell Switch Router (CSR)* da Toshiba, o *IP Switching* da Ipsilon [9], o *Tag Switching* da Cisco [10] e o ARIS da IBM.

Tendo em conta as várias soluções proprietárias atrás mencionadas, criou-se em 1997 o *Multi-Protocol Label Switching (MPLS) Working Group* do IETF, a fim de propor uma norma que integre o melhor de cada solução.

B. O multicast IP sobre ATM

Actualmente, a maioria das aplicações Internet baseiam-se em comunicações ponto-a-ponto. O uso generalizado da Internet está a alterar esta situação, levando a um aumento das necessidades de serviços de difusão e/ou de comunicação entre grupos (*multicasting*).

Não existia originalmente suporte *multicast* no *Classical IP*. Posteriormente introduziu-se o *Multicast Address Resolution Server (MARS)* [1] que agrega um grupo de nós (*cluster*). O MARS permite o mapeamento entre o endereço *multicast* da camada 3 e os correspondentes endereços ATM. Assim, quando o emissor pretender enviar uma mensagem para um endereço IP *multicast* e desconhecer quais os vários endereços ATM correspondentes terá de solicitar ao servidor MARS a resolução.

A especificação do MARS oferece duas opções. Na primeira opção, cada nó que pretende enviar uma mensagem *multicast* estabelece uma ligação ponto-multiponto para todos os nós terminais desse grupo (*VC-mesh*). Na segunda opção, o nó estabelece ligação com um ou mais servidores *multicast (MCS)* que por sua vez mantêm ligações com os nós terminais.

Alguns sistemas poderão beneficiar do método MCS, e.g. quando os elementos do grupo mudam frequentemente ou quando o grupo apresenta um elevado número de elementos.

Uma estrutura do tipo *VC-mesh* poderá ser preferível quando o grupo é pequeno e estável ou o atraso de *re-assembly* e retransmissão dos pacotes AAL5 é inaceitável.

O domínio do MARS, denominado como *Cluster MARS*, é na prática um conjunto de pontos terminais que escolhem usar o mesmo MARS para registarem a sua participação em grupos e receber daquele as suas actualizações.

Em geral, o *Cluster MARS* vai-se sobrepor totalmente a uma *Logical IP Subnet (LIS)* a qual é definida pelo conjunto de nós que usam o mesmo servidor ATMARP.

III. O IPv6 sobre o ATM

Enquanto o IPv4 delegava para um outro módulo a resolução de endereços da camada 3 para a camada 2, e assim apareceram diferentes soluções como o *Address Resolution Protocol (ARP)* ou o ATMARP, o IPv6 assume ele próprio essa função. No entanto o IPv6 requer da camada inferior um meio do tipo *multicast connectionless*. Ora, como o ATM não oferece esta funcionalidade de forma directa, é necessário introduzir uma subcamada entre a camada 3 e a camada 2 que ofereça este ambiente ao IPv6 para assim poder executar o *Neighbor Discovery*, o *Router Discovery*, e o *Address Configuration*.

Dado o servidor MARS apresentar um VC ponto-a-ponto com todos os elementos do *cluster* oferece o sistema ideal para a distribuição das mensagens *broadcast*. Outra possibilidade sugerida é a utilização do *ClusterControlVC* para a distribuição de mensagens de *Neighbor Discovery*. Em [2] e [3] é apresentado um modelo para IPv6 sobre ATM que possibilita o estabelecimento de *shortcuts* em ATM. Este modelo assenta nos servidores MARS para a emulação do meio *broadcast* aos membros pertencentes à mesma ligação.

O IETF, nomeadamente o grupo ION, optou por utilizar o MARS como a estrutura que possibilita oferecer à camada IPv6, através da subcamada IPv6-ATM, o meio *broadcast*. No entanto o MARS apresenta um conjunto significativo de restrições, como por exemplo o não retirar qualquer benefício das tecnologias IPv6 ou do UNIV4.

Praticamente todos os estudos elaborados na integração do IP sobre o ATM não tiram proveito das melhorias introduzidas pelo IPv6, nomeadamente a introdução dos cabeçalhos opcionais, e utilizam versões antigas do *User Network Interface (UNI)*. De facto, com o desenvolvimento do UNIV4 [5] introduziram-se os endereços de grupo, geridos pelo *Interim Local Management Interface (ILMI)*, e o *Leaf Initiated Join (LIJ)*. Enquanto os endereços de grupo oferecerão uma maior simplificação nas tabelas de mapeamento dado possibilitarem relações de um-para-um, o LIJ possibilita aos nós ATM poderem-se eles próprios ligar a um grupo *multicast*.

IV. O projecto MEDIA

O trabalho descrito neste artigo insere-se no projecto *MARS Extensions Developed for IPv6 over ATM (MEDIA)*, projecto PRAXIS/P/EEI/11223/1998, do grupo de Comunicações e Serviços Telemáticos do Departamento de Engenharia Informática da Universidade de Coimbra. O projecto MEDIA pretende estudar extensões aos ambientes

MARS e tirar proveito de algumas funcionalidades introduzidas no IPv6 e das inovações verificadas no ATM.

Nomeadamente, propõe-se a utilização de parâmetros adicionais de QoS, tanto nas mensagens de *Neighbor Discovery (unicast)* como nos pedidos de conversão dos endereços IP *multicast* em endereços ATM [11].

A proposta de novos módulos adicionais para o MARS, em vez da construção de uma nova estrutura, facilita o sucesso do estudo apresentado neste trabalho, dada a aceitação do protocolo MARS.

Em paralelo com este trabalho, no projecto MEDIA, está em desenvolvimento um simulador de IPv6 sobre ATM - SIANET, onde serão exploradas algumas ideias preliminares, realizados testes complementares das soluções implementadas e avaliados mecanismos não implementados.

A. O simulador SIANET

Para complementar os estudos desenvolvidos na plataforma, foi desenvolvido um simulador – SIANET – para redes IPv6 sobre ATM (Figura 1). O SIANET encontra-se implementado de forma modular para facilmente se adaptar a novos módulos e estudos.

O SIANET é uma ferramenta complementar à plataforma experimental em construção. Esta ferramenta de simulação será usada para explorar algumas das propostas que estão a ser avaliadas na plataforma e para estudar e desenvolver propostas adicionais não implementadas.

Os tempos utilizado no simulador foram obtidos na plataforma prática, descrevendo o presente artigo os testes que permitiram a sua medição.

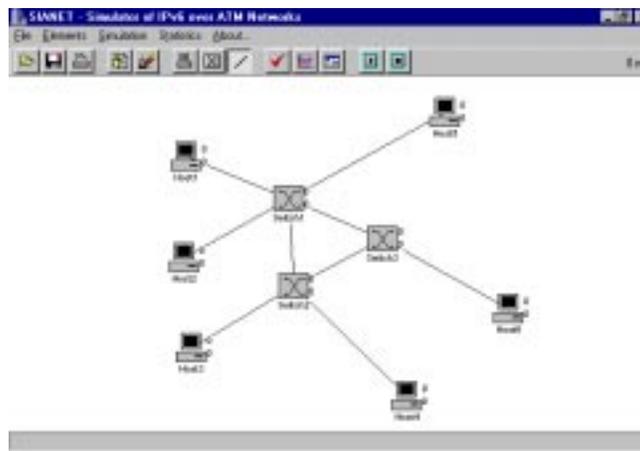


Figura 1 - Janela principal do SIANET.

V. A plataforma prática IPv6 sobre ATM

Está a ser implementado um ambiente experimental constituído por um rede IPv6 sobre ATM. A rede é constituída por um comutador Fore 200ASX e 4 PCs com FreeBSD. Foram implementados os módulos MARS cliente e servidor utilizando estruturas VC-Mesh e MCS. Também foi desenvolvida a nova sub-camada IPv6 sobre ATM que faz a interface entre o IPv6 e o ATM.

A Figura 2 mostra a nova pilha protocolar da plataforma prática.

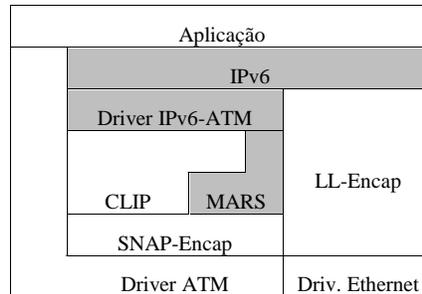


Figura 2 - Pilha protocolar final de um router

O IPv6 vai interagir com uma nova subcamada IPv6-ATM. Esta subcamada oferece ao IPv6 o meio *multicast connectionless* utilizando os serviços MARS.

Os protocolos ARP e ATMARP desaparecem, uma vez que a resolução de endereços passa a ser feita pelo próprio IPv6.

O *device driver* para a placa ATM, no sistema operativo FreeBSD, foi desenvolvido no âmbito de um projecto do Grupo de Comunicações e Serviços Telemáticos.

VI. Testes e resultados

A plataforma prática construída neste trabalho tem como objectivo principal permitir o estudo e a avaliação das tecnologias envolvidas. Neste contexto a plataforma foi também utilizada para medir tempos necessários para o simulador SIANET e ainda como validação do mesmo.

Foram medidos tempos nas várias camadas protocolares (Figura 3): camada IP, sub-camada IPv6 – ATM, camada CLIP, protocolo MARS e rede ATM. Esta secção descreve a obtenção destes tempos.

Tempos	Aplicação	
1	IPv6	
2	Driver IPv6-ATM	
3	CLIP	LL-Encap
4	MARS	
5	SNAP-Encap	
6, 7	Driver ATM	D. Ethernet

Figura 3 – Pilha protocolar de um *router* e tempos medidos. 1- tempos na camada IP; 2- tempos na sub-camada IPv6 – ATM, 3- tempo na camada CLIP; 4- tempos no protocolo MARS; 5- tempo de encapsulamento e desencapsulamento na camada CLIP; 6- tempo de estabelecimento de VC; 7 - tempo na rede ATM.

A. Número de observações e precisão da média

O valor do tempo obtido para cada experiência foi o resultado da média dos tempos obtidos em 10 observações. Os valores de precisão relativa das médias obtidas com as 10 observações para um intervalo de confiança de 95% variaram maioritariamente entre 1% e 4%, i.e., podemos afirmar com 95% de confiança que a média real do tempo para determinada experiência difere da média dos tempos das 10 observações num máximo de 1% a 4%, valor que se considerou satisfatório para os objectivos traçados.

Descreve-se de seguida o método de cálculo da precisão relativa da média das observações efectuado para cada experiência.

Um intervalo com $100(1-\alpha)$ por cento de confiança para a média μ construído através de uma amostra de n observações é dado por:

$$\bar{X} \pm t_{n-1;1-\alpha/2} \sqrt{\frac{S^2_{n-1}}{n}}$$

onde a primeira parcela é a média da amostra e a segunda parcela a precisão absoluta dessa média [7].

Para um intervalo de confiança de $100(1-\alpha) = 95\%$ obtém-se um valor para $1 - \alpha/2$ de 0.975. Com 10 observações ($n=10$), o valor de $t_{9;0.975}$ retira-se de uma tabela de pontos críticos $t_{v;\gamma}$ para a distribuição T-Student com v graus de liberdade, onde γ representa a probabilidade da variável assumir um valor inferior a $t_{v;\gamma}$, sendo de 2,2622.

A precisão relativa da média \bar{X} da amostra obtém-se dividindo a precisão absoluta da média \bar{X} pela própria média.

B. A plataforma

A plataforma utilizada para a medição dos tempos é constituída por três máquinas com velocidades de processamento distintas, equipadas com placas ATM ForeRunner PCA-200E de 155 Mbps, ligadas a um *switch* ATM ForeRunner ASX 200-WG. O *trotski*, com

um processador Pentium MMX a 400 MHz e com 64MB de RAM, é a máquina mais rápida. O *mao* tem um processador Pentium MMX a 233 MHz e 32MB de memória. A máquina mais lenta é o *lenine* com o seu processador 486 DX4 a 100 MHz e 32 MB de RAM.

A Tabela 1 refere os tempos máximos de atraso de mensagens e de estabelecimento de ligação no *switch*.

Features	Specification
Switch Transit Delay	< 10 microseconds
Connection Setup Time	< 10 microseconds, 100 calls / second

Tabela 1 - Tempos de atraso do *switch* ForeRunner ASX 200-WG [manual do *switch* ForeRunner ASX 200-WG].

C. O nbyte benchmark

Para comparar as capacidades de processamento das três máquinas, recorreu-se à execução de um *benchmark*, o *nbyte benchmark*. O *nbyte benchmark* - BYTEmark Native Mode Benchmark ver. 2 (10/95), inclui dez algoritmos diversos: numeric sort - ordenação de inteiros, string sort - ordenação de caracteres, bitfield – manipulação de bits, fp emulation – vírgula flutuante, fourier – coeficientes de Fourier, assignment – operações com matrizes de inteiros, idea - encriptação, huffman – algoritmo de compressão de Huffman, neural net - vírgula flutuante, e lu decomposition – método LU para decomposição de matrizes.

O programa calcula três índices: um de vírgula flutuante, um de inteiros e um de memória. O índice de vírgula flutuante foi desprezado dado que o código da plataforma que se pretende temporizar não faz uso deste tipo de operações, tendo sido considerada apenas a média entre o índice de inteiros e o índice de memória para o cálculo da velocidade relativa das máquinas.

O ambiente base do *benchmark* (índices 1) foi constituído pelo processador AMD K6/233, 512 KB L2-cache utilizando o gcc 2.7.2.3 e o libc-5.4.38. Na Tabela 2 encontram-se os índices obtidos em cada máquina pela execução do *benchmark*.

Máquina	Índice nbench
<i>lenine</i>	0,1385
<i>mao</i>	0,599
<i>trotsky</i>	1,4045

Tabela 2 - Índices obtidos nas máquinas de teste pela execução do *benchmark* nbench.

D. Método de obtenção de tempos

Os tempos que envolvem duas máquinas (tempo de estabelecimento de VC e tempo de envio de mensagens) foram medidos, em ambos os sentidos, entre o *trotsky* e o *mao*, as duas máquinas mais rápidas, e entre o *trotsky* e o *lenine*, a máquina mais rápida e a mais lenta.

Os tempos que envolvem apenas uma máquina foram medidos na máquina mais rápida e na máquina mais lenta, o *trotsky* e o *lenine*.

A medição de cada tempo passou pela identificação dos pontos inicial e final do tempo a medir no código do sistema operativo e do *driver*. Nesses pontos foi colocada uma chamada à função do kernel `microtime()`, que obtém a hora do sistema em microsegundos (μ s). Depois da medição no ponto final, calcula-se a diferença entre os dois instantes medidos e afixa-se no monitor (Figura 4).

```
struct timeval tv_my_time[2];
...
microtime(&tv_my_time[0]);          /* ponto inicial */
...
microtime(&tv_my_time[1]);          /* ponto final */
printf("My time: %ld  $\mu$ s", usec_delay( tv_my_time[1], tv_my_time[0] ));
```

Figura 4 – Código de medição de tempos.

Ao tempo total assim obtido é descontado o tempo t_{mt} que a própria chamada à função `microtime()` consome por si. Este tempo foi calculado com chamadas sucessivas a `microtime()`, sendo de 0,25 e 0,2 μ s nas máquinas com processador Pentium e de 4 μ s na máquina com processador 486.

A lentidão relativa do processador 486 do *lenine* foi um precioso auxiliar para a aplicação da lei de Amdahl (Figura 5) na determinação da distribuição dos tempos entre o equipamento de rede (placa ATM, cablagem e *switch*) e o processamento nas máquinas.

Existe no Pentium a função `rdtsc()`, mais leve e mais precisa que a função `microtime()`, que devolve o tempo em ciclos de relógio do processador, permitindo uma precisão igual ao período de relógio do processador (2,5 nano-segundos no Pentium a 400 MHz do *trotsky*). Esta função não foi utilizada de modo a permitir a utilização do *lenine*, um 486, na medição dos tempos, e porque a precisão da função `microtime()` é suficiente uma vez que os tempos que se pretendem medir são da ordem das dezenas de μ s a milhares de μ s.

E. Lei de Amdahl

Partindo-se do tempo inicial de uma experiência e do tempo final obtido por uma melhoria numa parte dessa experiência, a lei de Amdahl (Figura 5) declara que o melhoramento global obtido (*speedup* S) é inversamente proporcional à fracção de tempo não melhorada (f) somada da fracção de tempo melhorada (1-f) dividida pela melhoria nessa parte (*speedup* local S_l).

$$S = \frac{1}{f + \frac{1-f}{S_l}}$$

Figura 5 – Lei de Amdahl

Nas experiências realizadas, a melhoria introduzida foi a substituição de uma das máquinas envolvidas por outra mais rápida. O melhoramento global S calcula-se dividindo os tempos totais obtidos antes e depois da melhoria. O *speedup* local S_l é a velocidade da máquina nova em relação à inicial. Na Tabela 3 encontram-se as velocidades relativas (*speedup*) de cada máquina em relação às outras duas. O *speedup* da máquina A em relação à máquina B (*speedup*_{A/B}) obtém-se por divisão do índice nbench da máquina A pelo índice da máquina B. Como exemplo pode concluir-se que o *trotsky* é em média 10,14 vezes mais rápido que o *lenine* em operações de inteiros e de memória.

Máquina	Índice nbench	<i>Speedup</i> / <i>lenine</i>	<i>Speedup</i> / <i>mao</i>	<i>Speedup</i> / <i>trotsky</i>
<i>lenine</i>	0,1385	1,00	0,23	0,10
<i>mao</i>	0,599	4,32	1,00	0,43
<i>trotsky</i>	1,4045	10,14	2,34	1,00

Tabela 3 – Velocidades relativas das máquinas, segundo o índice nbench.

F. Exemplo de medição

A título de exemplo, descreve-se nesta secção a obtenção do tempo de envio de uma mensagem na rede ATM.

Para calcular o tempo na rede ATM, recorreu-se ao envio de mensagens MARS de teste de vários tamanhos, do cliente para o servidor e devolução da mesma mensagem pelo servidor (Figura 6).

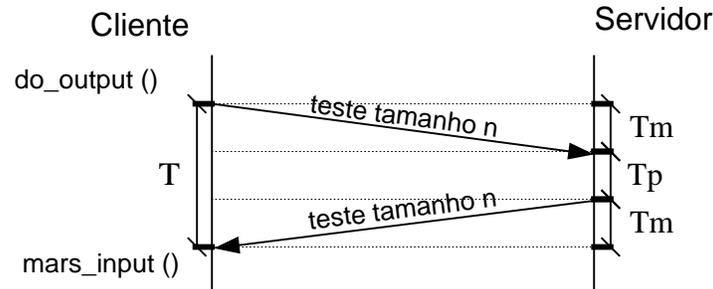


Figura 6 – Troca de mensagens MARS de teste

O tempo total T da experiência (Figura 6) foi medido no cliente sendo delimitado pelo envio da mensagem de teste para o servidor MARS e pela recepção no cliente da respectiva resposta. O ponto inicial de medida foi colocado no ficheiro *driver/mars.c* na chamada à função `do_output()` pela função `send_mars_mesg()` e o ponto final no ficheiro *driver/mars.c*, no início da função `mars_input()`, a função que recebe e processa a resposta. Ao tempo total são descontados os tempos de encapsulamento e desencapsulamento LLC.

Além do tempo total da operação T , foi medido o tempo de processamento do pedido no servidor T_p (Figura 6). T_p é delimitado pela recepção do pedido no servidor no início da função `mars_input()` e pela chamada à função `do_output()` na função `handle_mars_time_test()`, que processa o pedido e envia a resposta para o cliente. Subtraindo ao tempo total da experiência T o tempo T_p no servidor e dividindo por 2 obtemos o tempo de envio de uma mensagem T_m . Assim temos $T_m = (T - T_p)/2$.

Foi feita uma tentativa de medir estes tempos na camada mais baixa de interação com o *hardware* da placa ATM, no ficheiro *driver/fore_atm_200.c*, na função `transmit_pdu()` para o envio de mensagens e na função `pdu_received()` para a recepção, tentativa essa que foi abandonada por estas funções serem chamadas por outros protocolos com uma frequência de cerca de 2 vezes por segundo, e por ser difícil distinguir a este nível a origem e destino das mensagens sem interferir de forma significativa no tempo a medir.

Nas experiências realizadas fez-se variar o tamanho das mensagens de teste enviadas para valores de potências de 2 e valores intermédios desde 32 bytes até 2000 bytes. As mensagens de 32 bytes, mesmo com o cabeçalho de 8 bytes do encapsulamento LLC, são enviadas numa única célula ATM.

Os gráficos da Figura 7 apresentam os tempos T_m de envio de mensagens, em função do tamanho da mensagem, para as experiências com a máquina *trotsky* como servidor, e as máquinas *lenine* e *mao* como clientes.

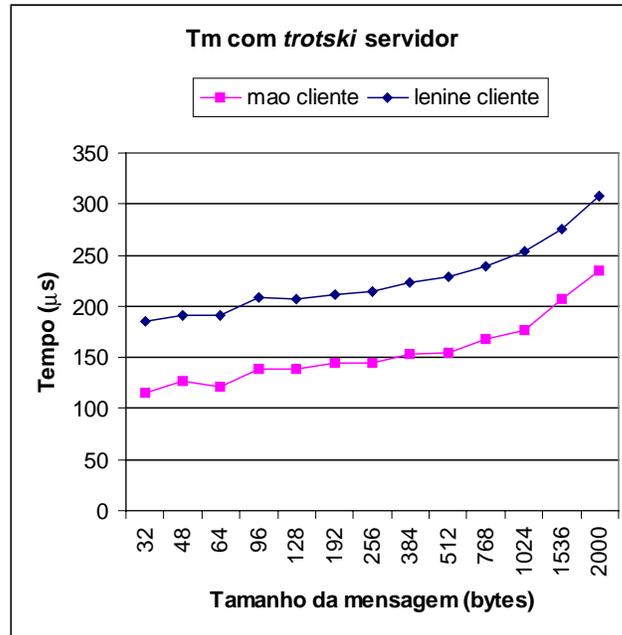


Figura 7 - Tempos T_m de envio das mensagens em função do tamanho das mesmas para as experiências com a máquina *trotsky* como servidor e as máquinas *mao* e *lenine* como clientes.

G. Tabela de síntese

Os tempos utilizados no simulador SIANET são os obtidos na máquina mais rápida, o *trotski* e encontram-se resumidos na Tabela 4.

Tempo	μs
Tempo médio necessário para o tratamento de um bloco de informação pela camada IPv6 no sentido descendente	30
Tempo médio necessário para o tratamento de um pacote pela camada IPv6 no sentido ascendente	4
Tempo médio necessário pela camada IPv6 para consultar as tabelas de conversão de endereços	0.6
Tempo médio necessário pela subcamada IPv6 sobre ATM para tratar um pacote unicast no sentido descendente	0.5
Tempo necessário pela subcamada IPv6 sobre ATM para tratar um pacote <i>multicast</i> no sentido descendente	384
Tempo médio necessário pela subcamada IPv6 sobre ATM para tratar um pacote no sentido ascendente	0.5
Tempo médio necessário para o encapsulamento da informação pela camada CLIP	3
Tempo médio necessário para o desencapsulamento da informação pela camada CLIP	2.2
Tempo médio necessário pelo servidor MARS para responder a um MARS_REQ	189
Tempo médio necessário pelo cliente MARS para tratar um MARS_MULTI	154
Tempo médio necessário pelo servidor MARS para fazer o <i>broadcast</i> da informação recebida	120
Tempo médio para o encaminhamento da informação pelo MCS	120
Tempo médio da passagem das células na camada ATM	48.7

Tabela 4 - Síntese dos tempos utilizados no simulador SIANET.

VII. Conclusões

Este artigo apresenta testes e resultados relativos à plataforma prática do IPv6 sobre ATM, construída no âmbito do projecto MEDIA, tendo em vista a quantificação do seu desempenho.

Os tempos obtidos nesses testes foram ainda utilizados no simulador SIANET para IPv6 sobre ATM, em desenvolvimento no âmbito do mesmo projecto. A plataforma é também utilizada para validação do simulador.

O valor do tempo obtido para cada experiência foi o resultado da média dos tempos obtidos em 10 observações. Os valores de precisão relativa das médias obtidas com as 10 observações para um intervalo de confiança de 95% variaram maioritariamente entre 1% e 4%, valor que se considerou satisfatório para os objectivos traçados.

Os resultados obtidos poderão ainda ser utilizados em estudos comparativos com outras plataformas IPv6 sobre ATM, IPv4 sobre ATM, ou mesmo IPv6 sobre *Ethernet*.

Agradecimentos

O trabalho apresentado neste artigo é parcialmente financiado pela Fundação Ciência e Tecnologia, no âmbito do projecto PRAXIS/P/EEI/11223/1198: *MEDIA – MARS Extensions Developed for IPv6 over ATM*.

Referências

- [1] Armitage, G., "Support for Multicast over UNI 3.1 Networks based ATM Networks", RFC 2022, November 1996
- [2] Armitage, G., Schuler, P., Jork, M. and Harter, G., "IPv6 over Non-Broadcast Multiple Access networks", RFC 2491, January 1999
- [3] Armitage, G., Schuler, P., and Jork, M., "IPv6 over ATM Networks", RFC 2492, January 1999
- [4] ATM Forum Technical Committee, "LAN Emulation over ATM v1.0 Specification", January 1995
- [5] ATM Forum Technical Committee, "Integrated PNNI (I-PNNI) v1.0", Specification. af-96-0987r1, September 1996.
- [6] Laubach, M., "Classical IP and ARP over ATM", RFC 2225, January 1998
- [7] A. Law, W. Kelton, "Simulation modeling and analysis", McGraw-Hill, 1982
- [8] Luciani, J., Katz, D., Piscitello, Cole, B., Doraswamy, N., "NBMA Next Hop Resolution Protocol (NHRP)", RFC 2332, April 1998
- [9] Newman, P., Edwards, W., Hinden, R., Hoffman, E., Ching Liaw, F., Lyon, T., Minshall, G., "Ipsilon's General Switch Management Protocol Specification Version 2.0", RFC 2297, March 1998
- [10] Rekhter, Y., Davie, B., Katz, D., Rosen, E. and Swallow, G., "Cisco System Tag Switching Architecture Overview", RFC 2105, February 1997
- [11] Sá Silva, J., Veiga, N., Duarte, S. e Boavida, F., "O IPv6 Multicast sobre Redes ATM", Encontro de Engenharia Informática 99 (EEI'99), Ordem dos Engenheiros, Dezembro de 1999