MÁRCIO GASPAR ANTÓNIO

ESTUDO DO IMPACTO DO TAMANHO MÁXIMO DA CARGA DA TRAMA ETHERNET NO PERFIL DO TRÁFEGO IPV6 NA INTERNET

Orientador: Nuno Manuel Garcia dos Santos

Universidade Lusófona de Humanidades e Tecnologias

Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação

Lisboa

2013

MÁRCIO GASPAR ANTÓNIO

ESTUDO DO IMPACTO DO TAMANHO MÁXIMO DA CARGA DA TRAMA ETHERNET NO PERFIL DO TRÁFEGO IPV6 NA INTERNET

Dissertação apresentada para a obtenção do Grau de Mestre no Curso de Mestrado em Engenharia Informática e Sistemas de Informação, conferido pela Universidade Lusófona de Humanidades e Tecnologias de Lisboa.

Orientador: Prof. Doutor Nuno Manuel Garcia dos Santos

Universidade Lusófona de Humanidades e Tecnologias

Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação

Lisboa

2013

Dedicatória

Dedico este trabalho

à minha família em especial a minha mãe.

Agradecimento

Primeiramente agradeço à Deus todo - poderoso que me tem ajudado em todos os momentos da minha vida.

Ao Doutor Nuno Garcia pela presença constante, pelo incentivo, companheirismo e gentileza que me ajudaram e contribuiram para a metodologia científica.

A minha gratidão aos meus colegas de Mestrado que ingressarmos juntos em 2010, pelas constantes trocas e demonstrações de amizade em especial ao Ricardo Barbosa e Pedro Cunha pela troca de ideias durante os dois anos e também por estarem sempre comigo.

A minha família maravilhosa, à Serafina Francisco Gaspar e aos meus irmãos… sem palavras para expressar todos os meus agradecimentos.

Márcio António

Resumo

A transição entre a versão 4 para a versão 6 do Internet Protocol (IP) vem ocorrendo na comunidade da Internet. No entanto, a estrutura interna dos protocolos IPv4 e IPv6, em detalhe no tamanho dos seus cabeçalhos, pode provocar alterações no perfil tráfego da rede.

Este trabalho estuda as mudanças nas características de tráfego de rede, avaliando o que mudaria se o tráfego gerado fosse apenas IPv6 em vez de IPv4. Este artigo estende-se uma pesquisa anterior, abordando novas questões, mas usando os registos de dados reais disponíveis publicamente.

É adotada uma metodologia de engenharia reversa nos pacotes IPv4 capturados, permitindo assim inferir qual a carga original no computador tributário e em seguida reencapsular essa carga em novos pacotes usando restrições de encapsulamento IPv6.

Conclui-se que, na transição de IPv4 para IPv6, haverá um aumento no número de pacotes transmitidos na Internet.

Palavra-chave: análise de tráfego, redes de computadores, Ethernet, IPv4, IPv6.

Abstract

The transition between the version 4 to the version 6 of the Internet Protocol (IP) has been taking place in the Internet community. Yet, the internal structure of the IPv4 and IPv6 protocols, in detail in the size of its headers, may cause changes in the network traffic profile.

This paper studies the changes in the network traffic characteristics, by assessing what would change if the generated traffic was only IPv6 instead of IPv4. This paper extends a previous research, addressing new questions but using publicly available data traces.

The adopted methodology reverse engineers the captured IPv4 packets, and re-encapsulates the inferred original payload of the packets by using IPv6 encapsulation constraints.

Conclusion points out that, in the transition from IPv4 to IPv6, there will be an increase in the number of packets transmitted in the Internet.

Keyword: traffic analysis, computer networks, Ethernet, IPv4, IPv6.

Siglas e Acrónimos

µs- Microsegundo

ALGs- Application Layer Gateways

COST- European Cooperation in Science and Technology

CPU- Central Processing Unit

DNS- Domain Name System

DoS- Denial of Service

FDDI- Fiber Distributed Data Interface

GB- GigaByte

Http- Hypertext Transfer Protocol

IANA- Internet Assigned Numbers Authority

ICMP- Internet Control Message Protocol

ICOIN- International Conference on Information Networking

IEEE- Institute of Electrical and Electronics Engineers

IP- Internet Protocol

IPv4- Internet Protocol Version 4

IPv6- Internet Protocol Version 6

IPX- Internetwork Packet Exchange

ISATAP- Site Automatic Tunnel Adressing Protocol

ISP- Internet Service Provider

KB- KiloByte

Km- Kilometro

LAN- Local Area Networks

LDF- Lengh of Data Field

LLC- Logical Link Control

MAC- Medium Access Control

MB- MegaByte

Mbps- Megabit por segundo

ms- Milissegundo

MTU- Maximum Transmission Unit

NAT- Network Address Translation

NAT-PT- Network Address Translation – Protocol Translation

OSI- Open Systems Interconnection

PAD- Packet Aggregation and De-aggregation

PARC- Palo Alto Research Center

PDU- Protocol Data Unit

RFC- Request for Comments

SFD- Start of Frame Delemiter

TCP- Transmission Contro Protocol

UDP- User Datagram Protocol

UTP- Unshielded Twisted Pair

WAN- Wide Area Network

WIDE- Wideley Integrated Distributed Environment

Índice

Dedicatória i

Agradecimento ii

Resumo iii

Abstract iv

Siglas e Acrónimos v

Índice vii

Índice de Figuras ix

Índice de Tabelas xi

1 Introdução 1

1.1 Enquadramento 1

1.2 Definição do Problema 2

1.3 Motivação, objectivo e metodologia da solução 3

1.4 Organização da Dissertação 5

2 Estudo do Estado da Arte 6

2.1 Introdução 6

2.2 Estado da Arte 6

2.3 O padrão Ethernet 7

2.3.1 História 9

2.3.2 A trama Ethernet 11

2.3.3 O tamanho máximo de uma carga Ethernet 12

2.4 Mecanismo de Transição IPv4 Versus IPv6 14

2.4.1 Pilha Dupla (*Dual-Stack*) 16

2.4.2 Tunelamento (Encapsulation ou Tunneling) 17

2.4.3 Tradução (Translation) 19

3 DADOS EM ANÁLISE 21

3.1 Introdução 21

3.2 Dados IPv4 21

3.3 Conclusão 29

4 MÉTODOS EXPERIMENTAIS E RESULTADOS 30

4.1 Introdução **30**
4.2 Arquitectura da aplicação **30**
4.3 Métodos Experimentais **32**
4.4 Resultados **34**
4.5 Conclusão **37**

5 Conclusões **38**

Referências **41**

APÊNDICE I ..... **I**

Índice de Figuras

Fig. 2.1. Rascunho do esquema para a primeira rede Ethernet [6] 10

Fig. 2.2 Formato da trama Ethernet. 11

Fig. 2.3 Mecanismo de transição pilha dupla [11]. 16

Fig. 3.1 Gráfico de distribuição cumulativa de tamanho de pacotes [18]. 22

Fig. 3.2 Protocolo de distribuição 1 [18]. 23

Fig. 3.3 Gráfico de distribuição cumulativa de tamanho de pacotes [19]. 23

Fig. 3.4 Protocolo de distribuição 2 [19]. 24

Fig. 3.5 Gráfico de distribuição de tamanho de pacotes [20]. 24

Fig. 3.6 Protocolo de distribuição3 [20]. 25

Fig. 3.7 Gráfico de distribuição de tamanho de pacotes [21] 25

Fig. 3.8 Protocolo de distribuição 4 [21]. 26

Fig. 3.9 Gráfico de distribuição de tamanho de pacotes [22]. 26

Fig. 3.10 Protocolo de distribuição 5 [22]. 27

Fig. 3.11 Gráfico de distribuição de tamanho de pacotes [23] 27

Fig. 3.12 Protocolo de distribuição [23] 28

Fig. 3.13 Exemplo da linha de comando Wireshark para conversão dos ficheiros 29

“.dump”.

Fig. 4.1 Segmento de código da classe PacoteIP responsável pela formação da chave de pesquisa. 31

Fig. 4.2 Segmento de código da classe MapaPacotes responsável pela adição do pacote ao TreeMap. 32

Fig. 4.3 Variação do rácio do número de pacotes produzidos sobre o número de pacotes originais pela mudança para o protocolo IPv6 face ao limiar de agregação. 35

Fig. 4.4 Variação do rácio do número de bytes transmitidos sobre o número de bytes originais pela mudança para o protocolo IPv6 face ao limiar de agregação. 37

Índice de Tabelas

Tabela 3.1: Descrição dos ficheiros de dados. 22

Tabela 4.1 Variação do rácio do número de pacotes produzidos sobre o número de pacotes originais pela mudança para o protocolo IPv6 face ao limiar de agregação. 36

Tabela 4.2 Variação do rácio do número de pacotes produzidos sobre o número de pacotes originais pela mudança para o protocolo IPv6 face ao limiar de agregação. 36

Tabela I.1 Resultados do processamento considerando tempo de agregação de 0 µs. I

Tabela I.2 Resultados do processamento considerando tempo de agregação de 300 µs. I

Tabela I.3 Resultados do processamento considerando tempo de agregação de 500 µs. II

Tabela I.4 Resultados do processamento considerando tempo de agregação de 700 µs. II

Tabela I.5 Resultados do processamento considerando tempo de agregação de 1000 µs. II

1 Introdução

1.1 Enquadramento

É um facto conhecido que o limite do tamanho da carga de uma trama Ethernet condiciona o processo de formação dos pacotes IP numa máquina geradora de tráfego, em primeiro lugar porque este tipo de máquinas estão tipicamente situadas em redes de acesso, as quais implementam na sua vasta maioria, redes locais usando o padrão Ethernet. O padrão Ethernet [1] estipula que a trama Ethernet pode transportar no máximo 1500 bytes (B) de carga, *i.e.* e por exemplo, no *payload* de uma trama Ethernet podemos encapsular um pacote IP com um máximo de 1500 bytes de tamanho.

Este pacote IP é por sua vez composto por um cabeçalho e por mais um *payload*, o qual poderá conter um segmento de outros protocolos, como por exemplo, o protocolo TCP. No entanto, o processo de encapsulação dos dados gerados pela camada de aplicação está sujeito às limitações impostas pelas camadas mais baixas, e em particular, pelas limitações impostas pela camada de dados, definidas pela tecnologia usada, sendo esta genericamente o padrão Ethernet. Isto resume-se em que, independentemente do tipo de conteúdo gerado pela aplicação, este conteúdo será fraccionado por forma a que o pacote resultante possa ser acomodado dentro de uma trama Ethernet, *i.e.*, o pacote IP terá um tamanho máximo de 1500B. Este pressuposto inicial é confirmado pela observação dos dados capturados em vários pontos da rede, como por exemplo os dados publicados em [2].

Uma vez que o objectivo do trabalho é concluir sobre as mudanças no perfil de tráfego aquando da transição do IPv4 para IPv6, assumem-se alguns pressupostos. Em primeiro lugar, assume-se que as aplicações nas camadas mais acima e nas camadas mais abaixo da rede não sofrerão mudanças significativas, *i.e.*, que as aplicações que geram tráfego não mudarão na sua natureza, nos próximos tempos. Esta suposição é razoável, uma vez que apesar de haver uma permanente mudança nas aplicações que comunicam sobre a Internet, não são previsíveis mudanças radicais no tipo de aplicações que geram tráfego. Outra suposição é a de que os computadores que geram

tráfego continuarão a estar conectados sobre Ethernet, cujo padrão impõe um limite para o tamanho da carga transportada pela trama.

Este trabalho pretende ser uma contribuição ao conhecimento, no contexto do curso de Engenharia Informática e Sistemas de Informação, sobre o impacto do tamanho máximo da carga da trama Ethernet no perfil do tráfego IPv6.

Neste trabalho, usámos um conjunto de registos de tráfego real, disponibilizados publicamente, e aplicámos nestes registos um procedimento descrito em [2]. Seguidamente inferimos o rácio de criação dos pacotes nas várias máquinas tributárias, e refabricamos os pacotes com base nos cenários experimentais. Por fim, concluímos sobre as variações do perfil de tráfego.

1.2 Definição do Problema

O problema de investigação estudado nesta dissertação está descrito na questão principal e na hipótese principal, como seguem:

Questão principal

A mudança da versão 4 para a versão 6 do protocolo IP provoca alteração no perfil do tráfego de rede na Internet?

Hipótese principal

Se num ficheiro de tráfego real de rede, capturado num ponto significativo da estrutura, como por exemplo, num ponto de agregação de um *Internet Service Provider* (ISP), dois ou mais pacotes consecutivos têm as mesmas características, *i.e.*, têm os mesmos endereços IP de origem e de destino, e os mesmos portos de origem e de destino, e ainda o mesmo protocolo, e se os seus tempos de registo de captura estão suficientemente próximos, *i.e.*, se os dois pacotes foram capturados num intervalo de tempo pequeno, por exemplo 1 ms, então provavelmente estes pacotes foram gerados pela mesma aplicação e pelo mesmo evento na mesma máquina, ou com mais detalhe, a aplicação gerou um

conteúdo cujo tamanho excedia as especificações das camadas inferiores, e isso originou a divisão do conteúdo em mais do que um pacote; logo, é possível replicar o processo de encapsulação seguindo as diferentes restrições impostas pelo IPv4 e pelo IPv6, sendo depois observáveis as variações nos perfis de tráfego gerado.

1.3 Motivação, objectivo e metodologia da solução

O trabalho publicado em [2] e que recebeu um prémio ao melhor artigo científico da conferência sugere que a transição da versão 4 para a versão 6 do protocolo IP resultará num aumento do número de pacotes transmitidos sobre a rede. A verificar-se, este problema é importante porque se sabe que uma das principais limitações das redes actuais não é tanto o problema da largura de banda disponível mas sim o problema da velocidade de resposta dos *routers* e de outros equipamentos activos de rede [3]. No entanto, as conclusões de Garcia, Freire e Monteiro foram tiradas apenas sobre um conjunto limitado de dados, e como se verá adiante, não tiveram estudos consequentes. A motivação deste trabalho de mestrado é assim validar ou refutar as conclusões dos autores de [2], usando um outro conjunto de dados.

O objectivo desta dissertação é concluir sobre a pergunta de investigação, concretamente, "a transição da versão 4 para a versão 6 do protocolo IP provoca alguma alteração no perfil de tráfego de rede?". O objectivo passa ainda por estender e questionar os resultados da investigação de [2], assim como concluir sobre as mudanças no perfil de tráfego aquando da transição do IPv4 para IPv6.

Este estudo foi realizado desenvolvendo tarefas de emulação sobre tráfego real de rede, *i.e.*, foram recolhidos da Internet um conjunto de ficheiros que contêm informações sobre os pacotes de dados transmitidos numa ligação inter-continental entre o Japão e os Estados Unidos, foram inferidas quais as cargas originais geradas pela aplicação do utilizador, e posteriormente foram replicadas as acções de encapsulamento dessas cargas com diferentes restrições quanto à versão do pacote IP.

A escolha adequada dos ficheiros de tráfego, adiante designada como *traces* é importante porque deve ser o mais ergódica possível, *i.e.*, deve ser representativa de uma utilização variada das várias aplicações que comunicam sobre a Internet. Foi feita uma pesquisa dos vários ficheiros de traces que existem publicados na Internet, e usada a informação compilada pelo grupo de trabalho da Acção COST IC 0703 – "Traffic Monitoring and Analysis", e seleccionados os *traces* de dados anteriormente referidos, capturados num ponto de agregação de uma ligação entre o Japão e os Estados Unidos. Decorrente do tipo e da natureza desta ligação, presume-se que está salvaguardada a representatividade da pluralidade de aplicações e de utilizadores. Esta garantia é ainda mais reforçada dado o elevado número de pacotes capturados e presentes em cada ficheiro.

A investigação deste tema não tem recebido atenção da comunidade científica, como se pode verificar adiante no capítulo dedicado ao Estado da Arte.

A pesquisa suporte deste trabalho foi feito em:

. Livros
. E- Livros
. Revistas científicas
. Web - Sites

De uma maneira mais sistematizada, as tarefas realizadas no trabalho de dissertação compreendem os seguintes passos:

- realizar uma pesquisa em diferentes fontes bibliográficas sobre as mudanças no perfil do tráfego IPv6 na internet;
- analisar os protocolos relacionados com o problema;
- descrever um algoritmo que soluciona o problema;
- utilizar a ferramenta Wireshark para a conversão dos pacotes capturados;
- Construir uma ferramenta em Java que aplique o algoritmo de resolução do problema;
- analisar os resultados;
- tirar conclusões.

1.4 Organização da Dissertação

O presente trabalho está composto por 5 capítulos, organizado da seguinte forma: este parágrafo concluí a introdução, na qual descrevemos o problema estudado, a sua relevância, e ainda os procedimentos experimentais usados. Os restantes capítulos apresentam os seguintes temas:

Capítulo 2: Aborda o estado da arte descrevendo os primeiros estudos relacionando com o tema, bem como a generalidade do tema;

Capítulo 3: Apresenta os registos de tráfego real e o seu enquadramento, bem como a justificação sobre a adequabilidade destes registos para o estudo;

Capítulo 4: Descreve a arquitectura da aplicação construída; apresenta com detalhe os métodos experimentais e quais os pressupostos que serviram de suporte a cada um dos cenários; apresenta ainda os resultados obtidos;

Capítulo 5: Concluí esta dissertação apresentando as conclusões relevantes da experiência.

2 Estudo do Estado da Arte

2.1 Introdução

Este capítulo apresenta a revisão possível do estado da arte quanto ao problema principal de investigação. Começa por apresentar as conclusões da investigação mais recente nesta área e discute em seguida os conceitos relativos ao padrão Ethernet, e finalmente, apresenta vários mecanismos de transição entre IPv4 e IPv6.

A discussão do padrão Ethernet e dos mecanismos de transição entre IPv4 e IPv6 não são centrais para o problema de investigação, mas servem para reforçar a ideia de que a existência de ambos os protocolos na rede implica um aumento do número de pacotes transmitidos, e consequentemente, uma degradação no desempenho desta mesma rede.

2.2 Estado da Arte

Depois de feita uma pesquisa nos motores de busca académicos, como por exemplo, o *Google Scholar* e o IEEE Xplore, conclui-se que os efeitos de uma eventual mudança do tamanho do *payload* da trama Ethernet nos perfis de tráfego de rede é um assunto que não tem recebido atenção dos investigadores.

Tanto quanto é o melhor conhecimento do autor, o primeiro estudo publicado sobre este tema foi apresentado na International Conference on Information Networking 2008 (ICOIN 2008) [2] tendo recebido o *Best Paper Award*. Neste artigo, foi avaliado o impacto do limite *de facto* de 1500 bytes no perfil do tráfego IPv6.

Os autores de [4] experimentaram a transferência de ficheiros usando diferentes limites para a carga das tramas Ethernet, e concluíram que o desempenho da rede melhora significativamente

quando tramas Jumboframes são usadas. Os autores de [4] não focam especificamente a questão da transição de IPv4 para IPv6, embora concordem com os resultados publicados em [2].

Os autores em [2] descrevem o processo de recuperação do tamanho do *payload* original na camada de aplicação e questionam o efeito da aplicação de um cabeçalho IPv6 em vez do cabeçalho IPv4. Concluem que se fosse mantida a limitação dos 1500 bytes disponíveis para o *payload* da camada física, o número de pacotes gerados poderia ser superior ao que actualmente é, podendo ir até 45% de aumento do número de pacotes transmitidos na rede. Para tal, os autores utilizaram um conjunto de dados recolhidos em vários pontos nos Estados Unidos, configurando diferentes tipos de utilização da Internet, com predominância para a utilização académica, já que os pontos de recolha eram na sua maior parte associados a universidades.

Um dos objectivos da corrente dissertação é estender e questionar os resultados da investigação de [2], usando outras fontes de tráfego mais recentes e recolhidas noutro contexto.

Depois de pesquisas aprofundadas nas bibliotecas de artigos científicos na Internet, não foi possível encontrar trabalhos que se relacionassem com o tema proposto para este estudo, além dos já anteriormente referidos.

2.3 O padrão Ethernet

A Ethernet foi criada nos anos 70 pela Xerox e consiste num protocolo de rede para redes locais (LAN), sendo um dos tipos de rede mais utilizado, cuja função é baseada na ideia de criar pontos de rede enviando mensagens entre si. Estes pontos de rede estão assentes em topologias de rede que podem ser de vários tipos:

- Ponto a ponto – entre dois pontos distintos;
- Anel – os vários dispositivos estão dispostos numa rede em forma de anel;
- Estrela – os dispositivos tem de estar ligados obrigatoriamente a um posto central;
- Barramento – todos os dispositivos estão ligados a um barramento;

- Árvore – os dispositivos estão ligados de uma forma física que se assemelha a uma árvore;
- Malha – os dispositivos estão ligados de tal forma que um dispositivo pode ter mais do que uma ligação, formando estas ligações uma malha.

As ligações podem ser feitas em três tipos de cabos, o cabo coaxial, o cabo de par entrançado, por exemplo o Unshielded Twisted Pair (UTP) e a fibra óptica, e dependendo dos tipos de cablagem e da versão do padrão Ethernet usada, podem permitir as seguintes velocidades de comunicação:

- 10Mbit/s – 10Mbit Ethernet;
- 100Mbit/s – Fast Ethernet;
- 1000Mbit/s – Gigabit Ethernet;
- 10Gbit/s – 10Gigabit Ethernet.

Outras velocidades estão em estudo, como forma de alargar o protocolo Ethernet, bem assim como a Ethernet para redes sem fios, baseada na família de padrões 802.11, disponibiliza uma outra gama de velocidades. No entanto estas famílias estão fora do âmbito deste estudo, logo não serão apresentadas.

A Ethernet é uma tecnologia de interligação para redes locais (Local Area Networks – LAN) e é baseada no envio de tramas ou quadros (pedaços de informação de tamanho variável encapsulados numa estrutura lógica bem definida). Esta tecnologia define a cablagem e os sinais eléctricos para a camada física, como também o formato de pacotes e de protocolos para a camada de controlo de acesso ao meio (Medium Access Control - MAC) do modelo OSI (Open Systems Interconnection) [5].

Para outro tipo de redes, a Carrier Ethernet está a ser cada vez mais usada em redes de operadoras e metade das empresas da América do Norte estão a implementar agora o serviço de Carrier Ethernet [23].

O modelo OSI prevê sete camadas na estrutura da comunicação de dados entre dois computadores, dispondo-se estas entre a camada física, directamente relacionada com o meio de comunicação (cabo de cobre, fibra óptica, *etc*.) e a camada de aplicação directamente relacionada com o software que o utilizador está a operar. A lista completa das camadas do modelo OSI é: camada física, camada de ligação de dados (ou camada de dados), camada de rede, camada de transporte, camada de sessão, camada de apresentação e a camada de aplicação. No contexto deste estudo algumas das camadas intermédias podem ser ignoradas, até porque estas não estão sempre activas, em particular, a camada de sessão e a camada de apresentação, responsáveis pelo estabelecimento de uma sessão de comunicação entre duas aplicações, e responsável pela formatação dos dados destinados a serem apresentados na aplicação de destino.

O padrão Ethernet mapeia-se na camada de ligação de dados, também designada por camada 2. Esta camada divide-se ainda em duas sub-camadas, a LLC e MAC, isto é, a *Logical Link Control* e a *Media Access Control*. A sub-camada LLC encarrega-se de fornecer mecanismos de multiplexagem e de controlo de fluxo que permite que vários protocolos de rede (por exemplo IP ou IPX) sejam transportados pelo mesmo meio da rede. A sub-camada MAC gere o acesso a um canal de comunicação (à camada física) e gere também o endereçamento neste canal possibilitando a conexão de diversos computadores numa rede. O endereçamento é realizado pelo endereço MAC (também chamado endereço físico), que consiste num número único atribuído a cada dispositivo de rede pelo próprio fabricante, possibilitando o envio de pacotes para um destino na mesma rede. A sub-camada MAC actua como interface entre a sub-camada LLC e a camada física.

2.3.1 História

A Ethernet original foi inventada em meados da década de 1970 por Bob Metcalfe e David Boggs. A figura 2.1 mostra o desenho esquemático de Metcalfe para a Ethernet.

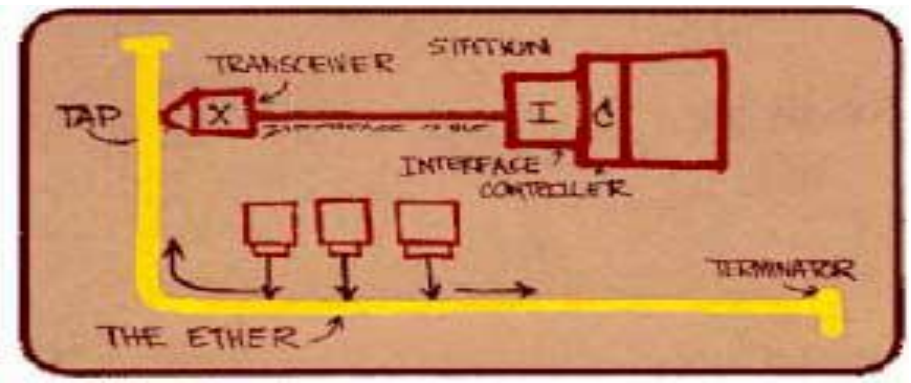

Fig. 2.1. Rascunho do esquema para a primeira rede Ethernet [6]

Na figura nota-se que a LAN Ethernet original usava um barramento coaxial para interconectar os nós.

As topologias de barramento da Ethernet persistiram durante toda a década de 1980 e até a metade de 1990, com uma topologia de barramento a Ethernet é uma transmissão LAN de broadcast, todos os quadros transmitidos movem-se para, e são processados por, todos os adaptadores conectados ao barramento.

A falta de normalização dificultava o progresso das pesquisas e a venda de equipamentos. Assim, com o objectivo de tentar resolver este problema foi atribuída ao IEEE, em 1980, a responsabilidade de conceber e gerir a normalização da Ethernet – IEEE 802.3. Desde que foi regulamentada pelo IEEE as suas especificações foram totalmente disponibilizadas. Este aspecto conjugado com a facilidade de operação e com a sua robustez resultou no grande sucesso da Ethernet [7].

A Ethernet original de Metcalfe e Boggs executava a 2,94 Mbps e interligava até 256 hospedeiros a distâncias de até 1,5 km. Metcalfe e Boggs conseguiram que a maioria dos pesquisadores do PARC da xerox se comunicassem por meio dos seus computadores.

2.3.2 A trama Ethernet

Designa-se por trama ou quadro o conjunto de bits enviados por uma dada estação. Ao nível da subcamada MAC (Medium Acess Control), a trama é vista como um conjunto de campos. O protocolo IEEE 802.3 especifica o formato da trama como especifica a figura 2.2.

Bytes	7	1	2 ou 6	2 ou 6	2	0-1500	0-46	4
	Preâmbulo	SFD	Endereço de destino	Endereço de origem	LDF	Dados	PAD	Verificação

Fig. 2.2- formato da trama Ethernet.

Seguidamente descreve-se o significado de cada um dos campos:

- Preâmbulo (Preamble): Sequência de uns e zeros consecutivos que permitem ao relógio do receptor sincronizar com o do emissor.

- SFD (Start of Frame Delemiter): um Byte (10101011) que identifica o início da trama.

- Endereços: Identificam o destinatário (Destination Address) e a origem (Source Address) da trama. Esses campos têm um comprimento de 2 ou 6 bytes.

- LDF (Lengh of Data Field): Indica o número de bytes que ocupa o campo de dados (DATA).

- Dados (DATA): campo de dados, o seu comprimento pode variar entre 0 e 1500 bytes.

- PAD: Campo de comprimento variável que só é inserido nas tramas mais pequenas de maneira a terem um comprimento mínimo de 64 bytes (comprimento mínimo da trama Ethernet). Assim, qualquer trama com comprimento inferior a 64 bytes terá um campo PAD.

- Verificação (Checksum): Campo que contem uma soma de verificação e que permite a detecção de erros.

Existem outros formatos de tramas, como por exemplo, a Jumboframe [8], ou ainda outras tramas de outras versões elaboradas sobre o padrão Ethernet, mas precisamente por não serem padronizadas, são usadas apenas em contextos muito particulares, e regra geral, não se usam nas redes locais, pelo que estão fora do âmbito deste estudo.

2.3.3 O tamanho máximo de uma carga Ethernet

Como pode ser visto pela análise da figura 2.2, o tamanho máximo dos dados transportáveis por uma trama Ethernet é de 1500 bytes. Isto significa que, independentemente da aplicação que estiver a ser usada, os dados a serem transmitidos serão partidos em blocos de 1500 bytes no máximo para poderem ser encapsulados numa trama Ethernet. Num exemplo trivial, consideremos uma aplicação de email com um anexo. Suponhamos que esse anexo é o artigo [2]. Ora, o email em questão pode facilmente chegar ao tamanho 570.000 bytes, uma vez que o artigo tem um tamanho de 569.293 bytes. Note-se ainda que muitos clientes de email permitem a adição de anexos até vários MB, por exemplo 25MB para o cliente GMail.

No exemplo acima, o cliente de email iria gerar 380 tramas Ethernet para enviar o artigo, isto é, além de a rede ter que comutar 380 tramas, a Internet teria que encaminhar os 380 pacotes resultantes desta acção. Para conteúdos mais pesados, como o caso limite do GMail, seriam geradas quase 17,500 tramas.

Não tendo chegado ao estatuto de padrão, existe ainda a opção Ethernet Jumboframe [8] já referida anteriormente. Nesta configuração, o tamanho do campo de dados pode chegar aos 8KB, isto é, 8192 bytes.

As Jumboframes têm o potencial de reduzir o esforço geral da rede e os ciclos de CPU dos equipamentos activos. Um trabalho recente demonstrou também o efeito positivo que as Jumboframes podem ter no desempenho global do TCP [4]. A presença de tramas grandes podem ter um efeito adverso sobre a latência de rede, especialmente sobre as ligações de baixa

largura de banda. O tamanho da trama usada por uma ligação ponto-a-ponto é tipicamente limitada pelo menor denominador comum. Por exemplo, as redes 802.5 Token Ring podem usar tramas de 4464 bytes e as redes FDDI pode usar tramas de 4352 bytes. Tecnologias mais recentes, como a 802.11 podem usar tramas de 7.935 bytes. O padrão Ethernet IEEE apenas admite tramas com um *payload* máximo de 1500 bytes.

O uso de 9000 bytes como tamanho preferido para Jumboframes surgiu a partir de discussões com a equipe de engenharia conjunta da Internet2 [9] e as redes do governo federal dos EUA. Pesquisas anteriores descobriram que as Jumboframes superam significativamente o desempenho do padrão de 1500 bytes em ambientes WAN [24], [25].

O eventual uso de tramas grandes nas redes locais tributárias do tráfego Internet, implica que os pacotes IP que são encaminhados pelos *routers* depois que estes saem das suas redes locais, têm que ser manipuláveis por toda a estrutura. Alguns *routers* podem implementar fragmentação de pacotes IP como resposta a este problema, mas isso implica que o *router* do outro lado seja capaz de realizar a desfragmentação dos pacotes recebidos.

Quando um equipamento de rede precisa de transmitir dados e isso implica um processo de encapsulação, esse equipamento tem em consideração aquilo que se designa por MTU (Maximum Transmission Unit), que é o valor máximo do unidade protocolar de dados (*Protocol Data Unit*, *PDU*) reportada ou definida pela e para a rede. Tipicamente em redes locais, e como consequência da limitação do tamanho do *payload* da trama Ethernet, o MTU é definido como sendo 1500 bytes.

Assim, se um equipamento de rede recebe um PDU que é maior do que o MTU definido para a rede do seu interface de saída, o equipamento precisa de efectuar a fragmentação desse PDU. A fragmentação de pacotes IP consiste que um datagrama seja dividido em pedaços, pequeno o suficiente para poder ser transmitido por uma conexão com o MTU menor que o datagrama original. Esta fragmentação acontece na camada IP (camada 3 do modelo OSI) e usa o parâmetro MTU da interface de rede que irá enviar o pacote pela conexão. O processo de fragmentação

marca os fragmentos do pacote original para que a camada IP do destinatário possa montar os pacotes recebidos, reconstituindo o datagrama original.

A princípio, um pacote é encaminhado e entregue com o mesmo tamanho que foi gerado na origem (fragmentação local). Mas, como a rota até o destino é uma escolha do *router*, um pacote pode seguir por uma rede que necessite de mais fragmentação. A fragmentação que ocorre durante a transmissão do pacote é invisível para o módulo IP do computador que enviou o pacote. Caso um pacote seja fragmentado, transmitido e remontado entre dois *routers* o módulo IP não será informado disto.

Assim, havendo necessidade de fragmentação entre dois *routers*, estes ficam obrigados a remontar os pacotes antes de entregá-los ao destino. Esta fragmentação é chamada de fragmentação transparente.

A fragmentação de pacotes que acontece no módulo IP é chamada de não-transparente. Isso significa que o IP pode enviar tanto fragmentos de pacotes como pode enviar pacotes sem fragmentação. Na prática, isso significa também que cada *router* receberá e transmitirá tantos pacotes completos, adequados ao tamanho da MTU, como fragmentos de pacotes ajustados a MTU.

2.4 Mecanismo de Transição IPv4 Versus IPv6

A transição entre a versão 4 para a versão 6 do Internet Protocolo (IP) vem ocorrendo na comunidade da Internet. A versão IPv6 implementa características que possibilitam por exemplo a ligação de um maior número de dispositivos. Com a implementação do IPv6 é preciso garantir que sistemas IPv6 possam enviar, rotear e receber datagramas IPv4 e vice versa.

Esta comunicação deverá ser conquistada através de mecanismos de transição, o IPv6 introduz melhoramentos significativos entre outros aspectos a nível de endereçamento, encaminhamento e segurança e apresenta os seguintes objectivos:

- Evitar saturação das tabelas de encaminhamento na Internet;
- Solucionar problemas de endereçamento;
- Introduzir mecanismos de segurança na camada de rede;
- O custo inicial de migração é baixo;
- O aumento de *sites* e serviços online baseados puramente em IPv6.

Por outro lado, esta transição apresenta os seguintes desafios e/ou problemas:

- *Routers* e máquinas devem ter seus programas de rede trocados sem que todos os outros no mundo tenham que trocar ao mesmo tempo;
- Quando as máquinas sofrerem a actualização, devem poder manter seus endereços IPv4, sem a necessidade de muitos planos de um re-endereçamento;
- Dificulta o surgimento de novas redes;
- Dificulta o surgimento de novas aplicações;
- Aumenta a utilização de técnicas, como o NAT, que quebram o modelo ponto-a-ponto da Internet.

Se comparamos os cabeçalhos de ambas as versões de endereço, verificamos que o IPv6 traz uma versão mais simplificada e com ela os seguintes benefícios:

- Melhor eficiência de roteamento para desempenho e escalabilidade;
- Ausência de broadcast e por isso não há ameaças de broadcast storms;
- Não há necessidade de se processar *checksums*;
- Mecanismos de cabeçalho de extensão simplificados e mais eficiente;
- Rótulo de fluxo para processamento sem a necessidade de abrir o pacote.

Existem de facto muitos mecanismos de transição testados e a escolha do mais apropriado para a nossa realidade deve ser sempre muito bem ponderada.

Consideremos 3 mecanismo de transição de IPv4 para IPv6:

- Pilha dupla (dual stack);
- Tunelamento (encapsulation ou tunnel);

- Tradução (translation).

Segundo Kurose e Ross [10], o IPv6 não é compatível com o IPv4, logo, as máquinas que utilizam apenas IPv4 e estão conectadas em redes IPv4, não se comunicam com máquinas IPv6 em redes IPv6. Para que as duas redes comuniquem entre si é necessário algum mecanismo de transição. Como estas duas versões são incompatíveis, foram desenvolvidas técnicas, denominadas por mecanismos de transição, que garantem a coexistência dos dois protocolos e o Interfuncionamento de ambos. Seguidamente descrevem-se os principais mecanismos que possibilitam esta comunicação.

2.4.1 Pilha Dupla (*Dual-Stack*)

A introdução da pilha dupla em uma rede permite que os hosts terminais e as aplicações efectuem uma transição baseada do IPv4 para o IPv6. A figura 2.3 mostra o esquema que ilustra este conceito.

O dual-stack é um método para integrar, activamente IPv6 e assim não são necessários mecanismos reais de tradução, a comunicação por IPv4 utiliza esse protocolo para o encaminhamento de pacotes IPv4 baseadas em rotas aprendidas por protocolos de encaminhamento específicos do IPv4. A comunicação IPv6 utiliza a pilha IPv6, através das rotas descobertas pelos protocolos de encaminhamento IPv6.

Esse mecanismo permite que nós IPv6 se comuniquem com nós IPv4 e realizem roteamento de pacotes IPv4 [10].

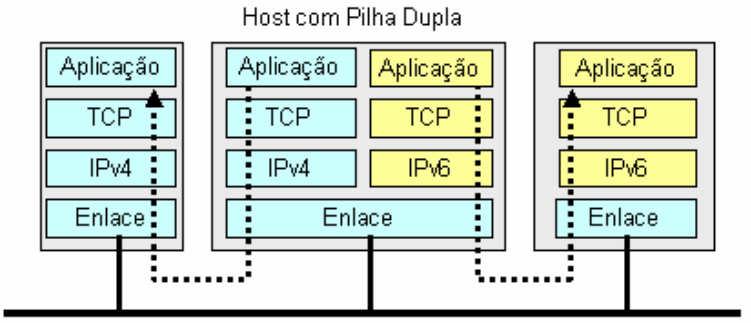

Fig. 2.3 Mecanismo de transição pilha dupla [11].

Um dos papéis preponderantes que este mecanismo desempenha é no serviço de DNS, onde o campo "versão" do cabeçalho do pacote IP, define que tipo de resposta será enviada: sempre que o DNS for perguntado e retornar a versão de IPv4 então a parte v4 da pilha é utilizada, ode forma idêntica acontece para o caso da versão 6.

Pode-se facilmente depreender que a técnica de pilha dupla adapta-se melhor as circunstancias onde haja necessidade de coexistência dos dois métodos de endereçamento. Esta necessidade surge certamente em grandes corporações, pois a substituição total de dezenas de milhares de máquinas implica grandes custos.

O problema deste método é o facto de requerer o uso de dois endereços IP para os dois casos possíveis. A tabela de roteamento também sofre mudanças no seu funcionamento pois passa-se a contar com duas tabelas, uma para cada versão, o que naturalmente aumenta exigências na quantidade de memória dos routeadores assim como o desgaste da CPU. Outros serviços como *Firewall* e DNS são igualmente adaptados de modo a trabalharem nas duas modalidades, o mesmo acontece com os comandos de gerenciamento que deverão ser adaptados.

2.4.2 Tunelamento (Encapsulation ou Tunneling)

O tunelamento se baseia em encapsular todo o tráfego IPv6 em pacotes IPv4, de forma a permitir uma comunicação entre dois hosts IPv6, através de uma rede IPV4, essas técnicas são tratadas no RFC 4213 [12].

Na prática, o roteador de borda conversa com a rede interna em IPv6, e quando há necessidade de comunicar-se com outra rede em IPv6 passando em IPv4, encapsula o pacote IPv6 em IPv4 formado assim o túnel de conexão com o site remoto.

O estabelecimento deste túnel pode ocorrer de duas formas:

- Manual - É necessário configurar manualmente os routeadores nas duas pontas de comunicação. Isto nos permite ter controlo sobre quem foi o autor do túnel;
- Automático – É criado conforme necessidade de modo automático.

Existem várias implementações de mecanismos baseados em túneis (*Tunnel Brokers, 6to4, ISATAP etc.*), diferindo apenas no modo e na facilidade de configuração.

2.4.2.1 Tunnel Brokers

É um método de tunelamento que usa o serviço de um provedor, que é propriamente responsável pela criação de todos os túneis. Está definido no RFC 3053 [13]. O cliente requerente da ligação com rede em IPv6 em primeiro lugar se regista no sistema do broker e faz uma solicitação do túnel em IPv4, o broker cria então o túnel e fornece as informações sobre o túnel ao cliente para sua configuração.

O broker mais conhecido é o www.freenet6.net [14] que se encontra hospedada em Canadá. O uso do broker tem no entanto a desvantagem de ser dependente do tipo de ligação entre o cliente e o broker, outra desvantagem reside no facto não ser conveniente para ligações que possuem IP atribuídos dinamicamente como são as ligações discadas.

2.4.2.2 6to4

O 6to4 serve principalmente para interligar duas redes IPv6 separadas por uma de IPv4 de forma que o prefixo 2002::/16 é usado para indicar que se trata de um IP 6to4, e os 32 bits seguintes são reservados para o endereço IPv4.

O 6to4 apresenta as seguintes desvantagens:

- Pode ser abusado com ataques DoS;
- Os túneis IPv6 podem sofrer spoofing.

2.4.2.3 ISATAP (Intra – Site Automatic Tunnel Adressing Protocol)

É um método de tunelamento dinâmico que utiliza um formato especifico de endereço do host – EUI-64, definido no RFC 4214 [15].

Este formato é constituído por:

- Prefixo reservado pelo organismo IANA (00-00-5e);
- Um valor fixo de 8 bit em Hexadecimal;

O prefixo FE::5EFE/96 é denominado prefixo do ISATAP. Todos os pacotes direccionados para o prefixo FE::5EFE/96 são enviados através da interface de ISATAP, que efectua o tunelamento para a interface IPv4 do host de destino de forma automática, para o desempenho das funções o ISATAP necessita que os dois hostes sejam munidos de pilha dupla.

2.4.3 Tradução (Translation)

Esta técnica consiste em usar algum dispositivo na rede que transforme os pacotes de IPv4 para IPv6 e vice-versa. Esse dispositivo deve ser capaz de realizar a tradução nos dois sentidos de modo de permitir a comunicação, e está definido no RFC 2766 [16].

A tradução pode ser feita nos elementos da rede ou nos sistemas finais, pode ter em consideração as características e estados anteriores do pacote (statefull) ou sem referência neles (stateless).

A conversão fundamental no processo de tradução é a conversão dos pacotes de IP e do ICMP. Para sua realização é utilizado o algoritmo cuja função é percorrer pacote à pacote e traduzir os cabeçalhos entre IPv4 e IPv6, permitindo que os usuários IPv6 tenham atribuídos um enderenço IPv4 temporariamente.

A utilização dos mecanismos de tradução permite a comunicação entre um domínio de rede IPv6 com um domínio IPv4, e vice-versa. Um desses mecanismos é o NAT-PT (*Network Address Translation – Protocol Translation*), baseado na filosofia do NAT, este mecanismo é instalado na fronteira entre os dois domínios de rede. Os terminais conhecem apenas um protocolo: IPv6 ou IPv4. O NAT-PT faz tradução de endereços e de cabeçalhos IPv4 e IPv6. Um dos problemas associados a este mecanismo está relacionado com o facto de existirem aplicações que não são totalmente independentes da camada de rede (por exemplo, FTP e

DNS - *Domain Name Service*). A solução para este problema é o uso de ALGs (*Application Layer Gateways*) que, ao nível da aplicação, são responsáveis pela tradução dos campos necessários.

Conclusão

Este capítulo apresentou os conceitos fundamentais necessários para o enquadramento do problema de investigação. Começou por descrever resumidamente o trabalho efectuado por diferentes autores relativos ao problema estudado, e continuou descrevendo os principais conceitos sobre o padrão Ethernet e sobre os mecanismos de transição e co-existências das versões 4 e 6 do protocolo IP. Esta secção encerra este capítulo.

3 DADOS EM ANÁLISE

3.1 Introdução

Este capítulo descreve a origem e os dados escolhidos para a análise do algoritmo implementado. Descreve-se ainda a relevância e adequação dos dados escolhidos para o estudo e compara-se este conjunto de dados com os dados do estudo de [2].

3.2 Dados IPv4

Para a análise do tráfego IPv4, foi utilizado um conjunto de registos de tráfego disponível em [17], que representam o tráfego medido numa série de servidores, dentro de uma janela de tempo conhecido.

Estes registos foram iniciados pelo projecto WIDE (Wideley Integrated Distributed Environment), que foi criado em 1988 e têm desempenhado um papel fundamental para o desenvolvimento e a construção da Internet no Japão. O projecto WIDE mantém registos de tráfego de dados para várias ligações. Uma dessas ligações, um cabo trans-Pacífico de 150 Mbps de largura de banda, tem um ponto de recolha de dados que foi denominado *samplepoint-F*.

Neste *samplepoint-F*, foram utilizados os ficheiros de dados relativos a 6 dias do mês de Outubro de 2010, concretamente, os ficheiros correspondentes às recolhas do dia 1 até ao dia 6. Os ficheiros foram escolhidos tendo em atenção um grupo de datas que não coincidisse com feriados ou com época de férias escolares ou estivais, numa tentativa de conseguir amostras ergódicas. Foi escolhido o princípio do mês de Outubro de 2010, por reunir as condições antes enunciadas, e porque foi possível conseguir um conjunto sequencial de seis dias de gravações.

Apesar de estes ficheiros não conterem todos os pacotes registados no dia em questão, estes seis ficheiros totalizam cerca de 40 GB de dados, e estão descritos na tabela 1.

Tabela 3.1: Descrição dos ficheiros de dados.

Nome	Data	Tamanho (KB)	Tempo (s)
201010011400.dump	01-10-2010	7.739.839	898.78
201010021400.dump	02-10-2010	8.036.128	900.04
201010031400.dump	03-10-2010	7.533.995	798.88
201010041400.dump	04-10-2010	8.185.729	900.16
201010051400.dump	05-10-2010	8.017.944	900.02
201010061400.dump	06-10-2010	3.343.862	899.57

O ficheiro 201010011400.dump tem as seguintes características relevantes que podemos encontrar em [18], tem um *Start Time: Fri Oct 1 14:00:02 2010* e o *End Time: Fri Oct 1 14:15:01 2010*, o tempo total é de 898.78 segundos, quanto a distribuição de tamanho de pacotes vê-se pelo gráfico que o *Packets* atinge a percentagem de cerca de 98% enquanto o Bytes atinge os 100% e que existe uma intersecção entre o *Packets* e o Bytes conforme se pode ver no gráfico da figura 3.1.

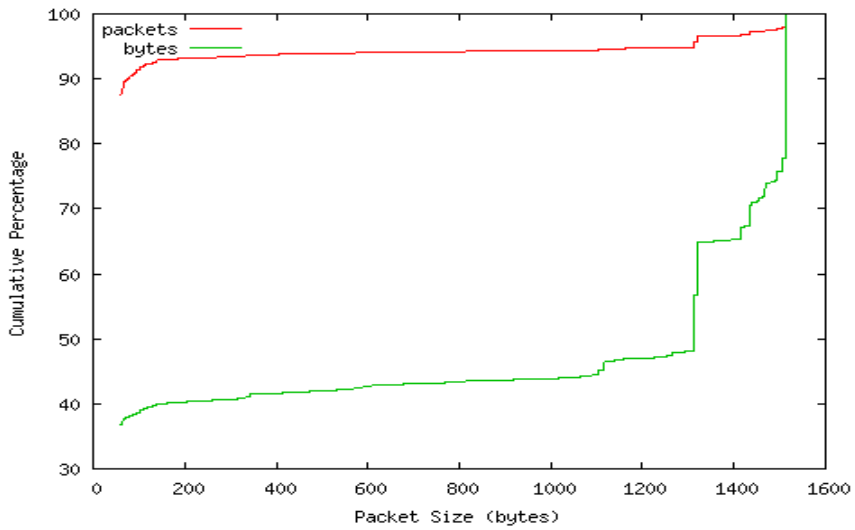

Fig. 3.1 Gráfico de distribuição cumulativa de tamanho de pacotes [18].

Quanto ao protocolo de distribuição vê-se claramente a partir do gráfico que o protocolo IP em termos de Bytes tem uma média de 100% (18634401417), o protocolo TCP tem em média 40% (7463749770), o http tem uma média de 28% (5176338148) enquanto que o protocolo UDP apresenta 49% (9224778683) como média e o IP6 10% (1863222537), conforme a figura 3.2.

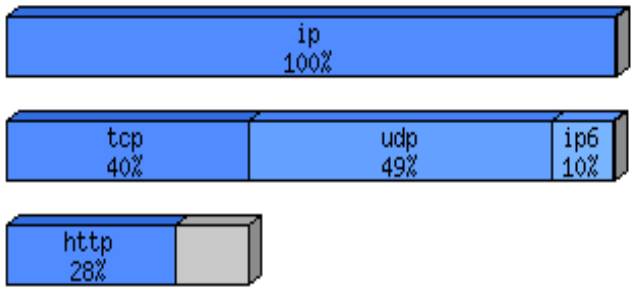

Fig. 3.2 protocolo de distribuição 1 [18].

Quanto ao ficheiro 201010021400.dump, apresenta as seguintes características [19], tem um *Start Time: Sat Oct 2 14:00:00 2010* e o *End Time: Sat Oct 2 14:15:00 2010*, o tempo total é de 900.04 segundos, quanto a distribuição de tamanho de pacotes vê-se pelo gráfico que o *Packets* atinge a percentagem de 98% enquanto o Bytes atinge os 100% e que existe uma intercessão entre o *Packets* e o Bytes conforme se pode ver no gráfico da figura 3.3.

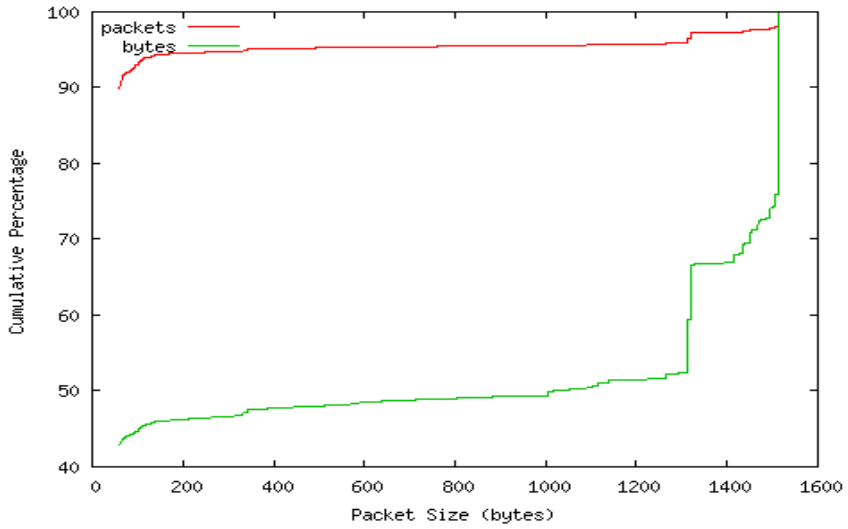

Fig. 3.3 Gráfico de distribuição cumulativa de tamanho de pacotes [19].

Quanto ao protocolo de distribuição o ficheiro 201010021400.dump apresenta os seguintes dados conforme o gráfico, o protocolo IP em termos de Bytes apresenta uma média de 96 % (16562621208), o protocolo TCP tem em média 34 % (5793435595), o http tem uma média que perfaz 26% (4513614803), o protocolo IP6 apresenta 8% (1454789392) em média enquanto que o protocolo UDP apresenta 54% (9254020996) como média, conforme a figura 3.4.

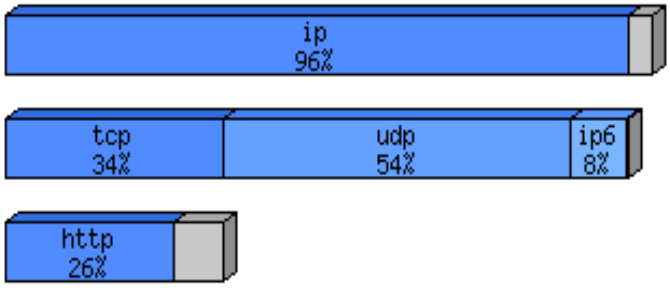

Fig. 3.4 protocolo de distribuição 2 [19].

Quanto ao ficheiro 201010031400.dump, apresenta as seguintes características [20], tem um Start Time: Sun Oct 3 14:00:03 2010 e o End Time: Sun Oct 3 14:15:01 2011, o tempo total é de 798.88 segundos, quanto a distribuição de tamanho de pacotes vê-se pelo gráfico que o *Packets* atinge a percentagem de 98% enquanto o Bytes atinge os 100% e que existe uma intercessão entre o *Packets* e o Bytes conforme se pode ver no gráfico da figura 3.5.

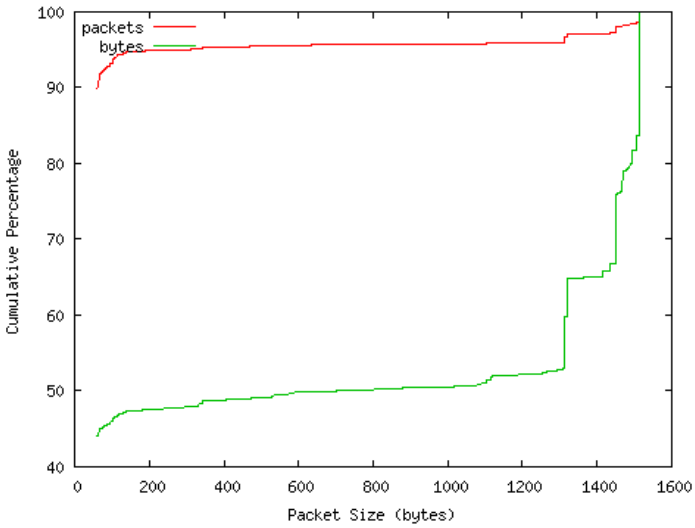

Fig. 3.5 Gráfico de distribuição de tamanho de pacotes [20].

Quanto ao protocolo de distribuição o ficheiro 201010031400.dump apresenta os seguintes dados conforme o gráfico, o protocolo IP em termos de Bytes apresenta uma média de 98 % (15386656065), o protocolo TCP tem em média 36 % (5609246908), o http tem uma média que perfaz 29% (4541518063), o protocolo IP6 tem como média 8% (1305091670), por sua vez o protocolo UDP apresenta 54% (8443557564) como média, conforme a figura 3.6.

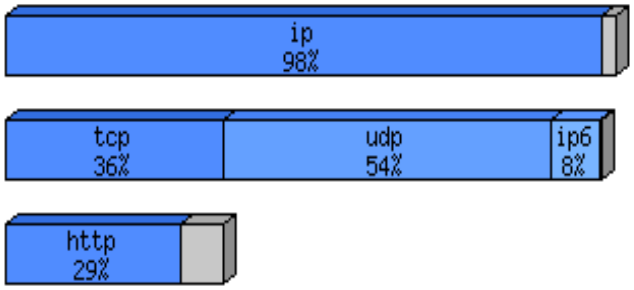

Fig. 3.6 protocolo de distribuição3 [20].

O ficheiro 201010041400.dump, apresenta as seguintes características [20], tem um *Start Time: Mon Oct 4 14:00:00 2010* e o *End Time: Mon Oct 4 14:15:01 2010*, o tempo total é de 900.16 segundos, quanto a distribuição de tamanho de pacotes vê-se pelo gráfico que o *Packets* atinge a percentagem de 98% enquanto o Bytes atinge os 100% e que existe uma intercessão entre o *Packets* e o Bytes conforme se pode ver no gráfico da figura 3.7.

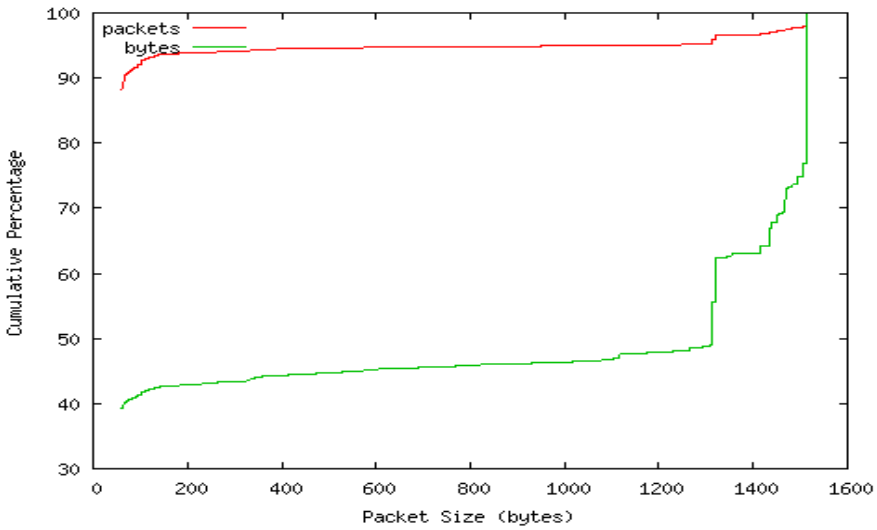

Fig. 3.7 Gráfico de distribuição de tamanho de pacotes [21]

Quanto ao protocolo de distribuição o ficheiro 201010041400.dump apresenta os seguintes dados conforme o gráfico, o protocolo IP em termos de Bytes apresenta uma média de 100 % (18642388096), o protocolo TCP tem em média 42 % (7896722373), o http tem uma média que perfaz 33% (6172965703), o protocolo IP6 tem como média 8% (1432716791), por sua vez o protocolo UDP apresenta 50% (9263239947) como média, conforme a figura 3.8.

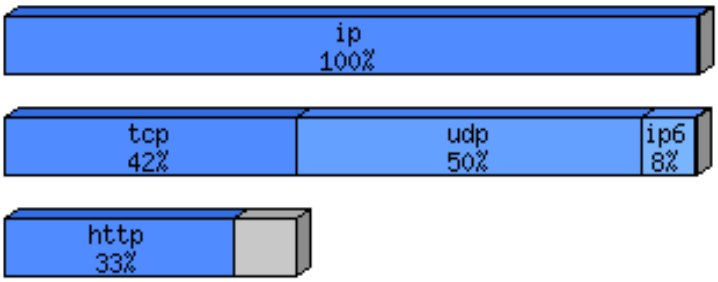

Fig. 3.8 protocolo de distribuição 4 [21].

O ficheiro 201010051400.dump, apresenta as seguintes características [22], tem um *Start Time: Mon Oct 5 14:00:01 2010* e *o End Time: Mon Oct 5 14:15:01 2010*, o tempo total é de 900.02 segundos, quanto a distribuição de tamanho de pacotes vê-se pelo gráfico que o *Packets* atinge a percentagem de cerca de 98% enquanto o Bytes atinge os 100% e que existe uma intercessão entre o *Packets* e o Bytes conforme se pode ver no gráfico da figura 3.9.

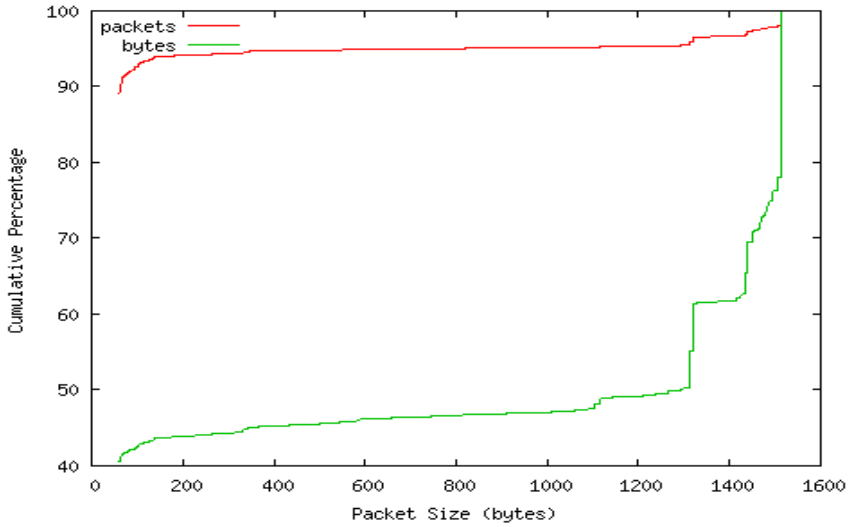

Fig. 3.9 Gráfico de distribuição de tamanho de pacotes [22].

Quanto ao protocolo de distribuição o ficheiro 201010051400.dump apresenta os seguintes dados conforme o gráfico, o protocolo IP em termos de Bytes apresenta uma média de 100 % (17918430699), o protocolo TCP tem em média 41 % (7379711498), o http tem uma média que perfaz 29% (5267428253), o protocolo IP6 tem como média 6% (1076753984), enquanto que o protocolo UDP apresenta 53% (9443197342) como média, conforme a figura 3.10.

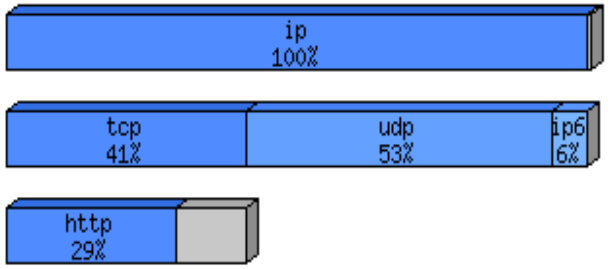

Fig. 3.10 protocolo de distribuição 5 [22].

O ficheiro 201010061400.dump, apresenta as seguintes características [23], tem um *Start Time: Wed Oct 6 14:00:01 2010* e o *End Time: Wed Oct 6 14:15:01 2010*, o tempo total é de 899.57 segundos, quanto a distribuição de tamanho de pacotes vê-se pelo gráfico que o *Packets* atinge a percentagem de cerca de 80% enquanto o Bytes atinge os 100% e que existe uma intercessão entre o *Packets* e o Bytes conforme se pode ver no gráfico da figura 3.11.

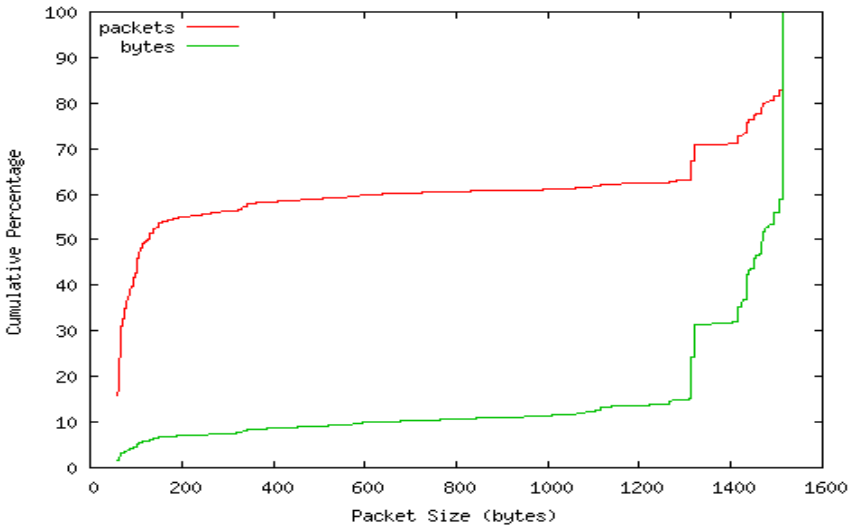

Fig. 3.11 Gráfico de distribuição de tamanho de pacotes [23]

Quanto ao protocolo de distribuição o ficheiro 201010061400.dump apresenta os seguintes dados conforme o gráfico, o protocolo IP em termos de Bytes apresenta uma média de 99 % (27850687483), o protocolo TCP tem em média 75 % (21010155463) o http tem uma média que perfaz 56% (15671399244), o protocolo IP6 tem como média 11% (2981141739), enquanto que o protocolo UDP apresenta 13% (3769028504) como média, conforme a figura 3.12.

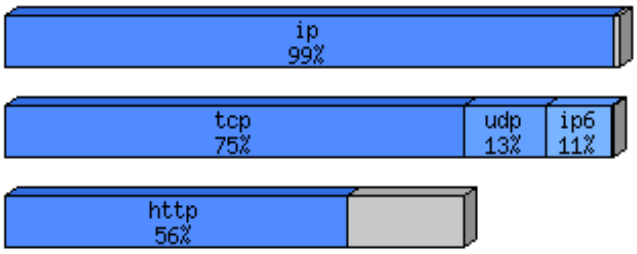

Fig. 3.12 protocolo de distribuição [23]

Apesar de os registos de pacotes conterem mais informação, para o estudo aqui apresentado, apenas interessam os seguintes campos de cada um dos registos:

a. Frame
b. Epoch Time
c. Source IP
d. Destination IP
e. Total Length
f. Protocol (TCP/UDP, …)
g. Source Port
h. Destination Port.

O campo *Frame* contém um número sequencial que identifica a ordem de captura da trama, e o campo *Epoch Time* contém um registo de tempo que nos identifica a hora relativa da captura do pacote. A resolução deste campo vai até ao nanosegundo.

Os campos *Source* e *Destination IP* contêm os endereços IPv4 de origem e de destino, assim como os campos *Source* e *Destination Port* contêm os números de portos de origem e de destino, relativos ao protocolo de transporte, tal como é descrito no campo *Protocol*. O campo *Total Length* descreve o comprimento total do pacote, em Bytes.

Os ficheiros foram inicialmente registados de acordo com o formato *.dump*, um formato usado pela aplicação Wireshark, entre outras. Procedeu-se de seguida, usando o Wireshark, à extracção das características relevantes para o estudo para um ficheiro de texto. A aplicação de

experimentação incidiu sobre estes ficheiros de texto. A figura 3.13 mostra o ecrã com a chamada à aplicação de linha de comando do Wireshark.

```
C:\Program Files (x86)\Wireshark>tshark -nr "d:\201010061400.dump" -e frame.number -e
frame.time -e ip.src -e ip.dst -e ip.len -e ip.proto –e tcp.srcport –e tcp.dstport –e
udp.srcport –e udp.dstport -V -T fields > d:/201010061400.txt
```

Fig. 3.13 Exemplo da linha de comando Wireshark para conversão dos ficheiro ".dump".

A necessidade de converter as características dos ficheiros para o formato de texto deveu-se ao facto de não ter sido possível encontrar as bibliotecas Java necessárias para interpretar directamente o formato nativo do Wireshark.

O trabalho descrito em [2] apresenta um estudo feito para ficheiros capturados em 2006, em servidores norte-americanos, representando diferentes tipos de tráfego.

Este estudo incide sobre dados recolhidos em 2010, transmitidos numa ligação inter-continental entre o Japão e os Estados Unidos, e decorrente da natureza desta ligação, presume-se que está salvaguardada a representatividade da pluralidade de aplicações e de utilizadores. Esta garantia é ainda mais reforçada dado o elevado número de pacotes capturados e presentes em cada ficheiro.

3.3 Conclusão

Este capítulo descreveu os ficheiros de dados escolhidos para o estudo do problema desta dissertação. Os ficheiros foram escolhidos de forma a poder ter uma amostra heterogénea de tráfego, evitando dias não laborais e com um elevado número de pacotes registados. O tamanho e características das amostras garantem a adequação dos dados ao objectivo desta investigação.

4 MÉTODOS EXPERIMENTAIS E RESULTADOS

4.1 Introdução

Este capítulo apresenta a arquitectura da aplicação construída, e descreve com detalhe os métodos experimentais; apresenta ainda os pressupostos que serviram de suporte a cada um dos cenários. Apresenta também os resultados obtidos e as conclusões da experiência. O capítulo termina com um breve resumo.

4.2 Arquitectura da aplicação

A aplicação construída para ler os ficheiros de texto gerados pela linha de comandos do *Wireshark* tem apenas três classes. A linguagem escolhida para construir esta aplicação foi o Java, porque desta forma era possível operacionalizar o seu desenvolvimento de uma forma mais célere.

As classes que foram implementadas na aplicação são as seguintes:

a) Classe FicheiroIP
b) Classe PacoteIP
c) Classe MapaPacotes.

A classe FicheiroIP encarrega-se de abrir o ficheiro de origem, criar o ficheiro de destino, e ler cada uma das linhas do ficheiro de origem até que a condição de saída seja encontrada (fim do ficheiro ou 10.000.000 de pacotes lidos com o mapa de pacotes em condições de ser limpo). Esta classe contém ainda as constantes necessárias ao funcionamento do programa, em particular, a do tempo de agregação. Esta classe ainda fabrica a chave que irá ser usada pela classe MapaPacotes (figura 4.1)

```
public String chave() {
      return srcIP + dstIP + (new Integer(protocol).toString())
                  + (new Integer(srcPort).toString())
                  + (new Integer(dstPort).toString());
}
```

Fig. 4.1 Segmento de código da classe PacoteIP responsável pela formação da chave de pesquisa.

A classe PacoteIP permite a instanciação de objectos do tipo PacoteIP, os quais contêm todos os dados referidos na secção 3.2. Esta classe implementa ainda métodos que permitem avaliar se dois pacotes que têm a mesma chave estão próximos ou não, de acordo com o tempo de agregação definido.

A classe MapaPacotes implementa uma estrutura do tipo TreeMap. Esta estrutura usa como chave os atributos mostrados na figura 4.1, gerados na classe PacoteIP, e como objecto, uma instância do tipo PacoteIP. É esta classe que implementa a agregação de pacotes (figura 4.2), permitindo seguir este algoritmo aquando da adição de um pacote ao mapa:

1. verifica se o mapa contém já pacotes com a mesma chave.
2. Se não,
    a. Adiciona o pacote ao mapa
3. Se sim,
    a. Verifica se o pacote que já existe no mapa é cronologicamente próximo deste novo pacote
    b. Se sim,
        i. Agrega o novo pacote ao pacote já existente no mapa, depois de retirar o comprimento do cabeçalho IP e o comprimento do cabeçalho TCP ou UDP
    c. Se não,
        i. Escreve para o ficheiro de resultados os pacotes cujo *Epoch time* é anterior ao tempo do novo pacote considerando o tempo de agregação
        ii. Adiciona o novo pacote no mapa.

```
     public void adicionarPacote(PacoteIP p) {
          PacoteIP novo = new PacoteIP(p);
          if (!map.containsKey(novo.chave())) {
// o pacote novo ainda não existe no mapa
               novo.setSize(novo.getSize()-20);
//retirar o cabeçalho IP
               map.put(novo.chave(), novo);
          } else { // o pacote novo já existe no mapa
               PacoteIP original = new PacoteIP(bW);
               original = map.get(novo.chave());
               if (original.estaProximo(novo, aggThreshold)) {
// o pacote novo está próximo e vai ser agregado
                    original.addSize(novo.getSize()-20);
//retirar os 20B do cabeçalho IP do 2º pacote
                    original.addSize(novo.getSize()-
                         novo.sizeOfL3Header());
//retirar compr cab. TCP/UDP
                    original.incNumPacotes();
                    updateStats(0, 1, 0, 0, 0);
                    map.put(original.chave(), original);
               } else {
// o pacote novo não está próximo, é preciso escrever o
// purgar o mapa de pacotes antigos e escrever o novo pacote no mapa
                    gravarMapaAntigo(novo.getTime(), aggThreshold);
                    map.put(novo.chave(), novo);
                    System.out.println("---------a gravar o pacote "
                              + novo.getFramenum());
               }
          }
     }
```

Fig. 4.2 Segmento de código da classe MapaPacotes responsável pela adição do pacote ao TreeMap.

A classe FicheiroIP irá instanciar um objecto do tipo MapaPacotes e tantos objectos do tipo PacoteIP quantos os necessários para povoar o mapa de registo, à medida que os dados são lidos do ficheiro de texto previamente gerado.

4.3 Métodos Experimentais

O processo experimental iniciou-se com a selecção e descarga dos ficheiros de registo de tráfego de pacotes para as datas consideradas relevantes. Os ficheiros seleccionados e descritos na Tabela 3.1 foram posteriormente processados por forma a extrair deles apenas os atributos

relevantes para a experiência, como foi descrito no capítulo 3. Estes ficheiros resumidos contêm assim apenas os seguintes dados: *Frame number, Epoch Time, Source IP, Destination IP, Total Length, Protocol (TCP/UDP, …), Source Port e Destination Port*.

Os ficheiros também contêm tramas que não contêm pacotes IP. Estas tramas são descartadas porque se presume que são tramas com origem e destino numa sub-rede específica dentro das instalações do ponto de captura, provavelmente geradas por um protocolo de gestão de equipamentos de rede, e que não têm relevância para o estudo da alteração do tráfego IP.

Os pacotes IP que suportam algoritmos específicos, como por exemplo, os pacotes que sustentam o *three-way handshake* do TCP, foram mantidos porque são importantes para esta análise. O algoritmo implementado sobre os ficheiros intermédios contendo estes registos resumidos de pacotes foi o seguinte:

1. Os pacotes são ordenados de acordo com o seu *Epoch Time*, por forma a que se mantenha a sequência cronológica de captura (e supostamente de transmissão);

2. Se dois pacotes têm as mesmas características, *i.e.*, têm os mesmos endereços IP de origem e de destino, e os mesmos portos de origem e de destino, e ainda o mesmo protocolo, e se os seus *Epoch Time* estão suficientemente próximos, *i.e.*, foram capturados num intervalo de tempo pequeno, por exemplo 1 ms, então provavelmente estes pacotes foram gerados pela mesma aplicação e pelo mesmo evento na máquina tributária; no cálculo do tamanho deste *payload*, é retirado ainda o valor correspondente ao cabeçalho da camada de transporte, por exemplo, é retirado o tamanho de um cabeçalho de um segmento TCP.

3. A verificação descrita em 2. é realizada para o primeiros dez milhões de pacotes contíguos no registo de dados, aproximadamente.

O resultado é um novo ficheiro que contém a data de geração do *payload* agregado dos vários pacotes do registo, sendo que aqui o parâmetro experimental é o tempo usado para estimar o que são "pacotes gerados no mesmo instante pela mesma aplicação".

Os detalhes armazenados neste novo ficheiro são ainda os descritos acima, sendo o *Epoch Time* considerado o primeiro tempo do primeiro dos pacotes eventualmente agregados.

Sobre este novo ficheiro de *payloads* é agora simulado o processo de encapsulação, tendo em consideração um factor: a alteração do formato do cabeçalho IPv4 para o cabeçalho IPv6, com o seu consequente aumento de tamanho e portanto, diminuição da carga útil do *payload*.

A refabricação dos pacotes IPv4 para pacotes IPv6 é assim conseguida retirando 20 bytes ao tamanho do cabeçalho IPv4 agora substituido pelo cabeçalho IPv6, de 40 bytes de comprimento, aumentados ao tamanho do pacote. A reencapsulação é assim feita tendo em consideração que a soma de payload e tamanho do cabeçalho não deve exceder a *Maximum Transmission Unit (MTU)* da trama Ethernet.

4.4 Resultados

Neste processo experimental, foram considerados os seguintes valores admissíveis para a vizinhança da agregação de carga de pacotes: 0 µs, 300 µs, 700 µs, 1 ms. Estes valores resultam da análise realizada em [2]. Os resultados numéricos da emulação estão apresentados no Apêndice I.

A figura 4.3 mostra o gráfico que representa a variação do rácio do número de pacotes criados quando se aplica o algoritmo de regeneração do tráfego IPv4 em tráfego IPv6. O eixo vertical mostra o rácio de (Número de pacotes criados com IPv6) / (Número de pacotes originais em IPv4), e o eixo horizontal mostra o tempo definido para o limiar de agregação, em micro segundos. No gráfico são exibidos os dados relativos a aproximadamente os primeiros dez milhões de pacotes para cada um dos seis ficheiros previamente apresentados.

O cenário de agregação para 0 µs corresponde na prática à troca simples do cabeçalho dos pacotes IPv4 por um pacote IPv6, e logo, todos os pacotes que tiverem um tamanho próximo do limite imposto pela MTU, uma vez que o cabeçalho IPv6 é maior, geram um segundo pacote adicional com a carga excedente.

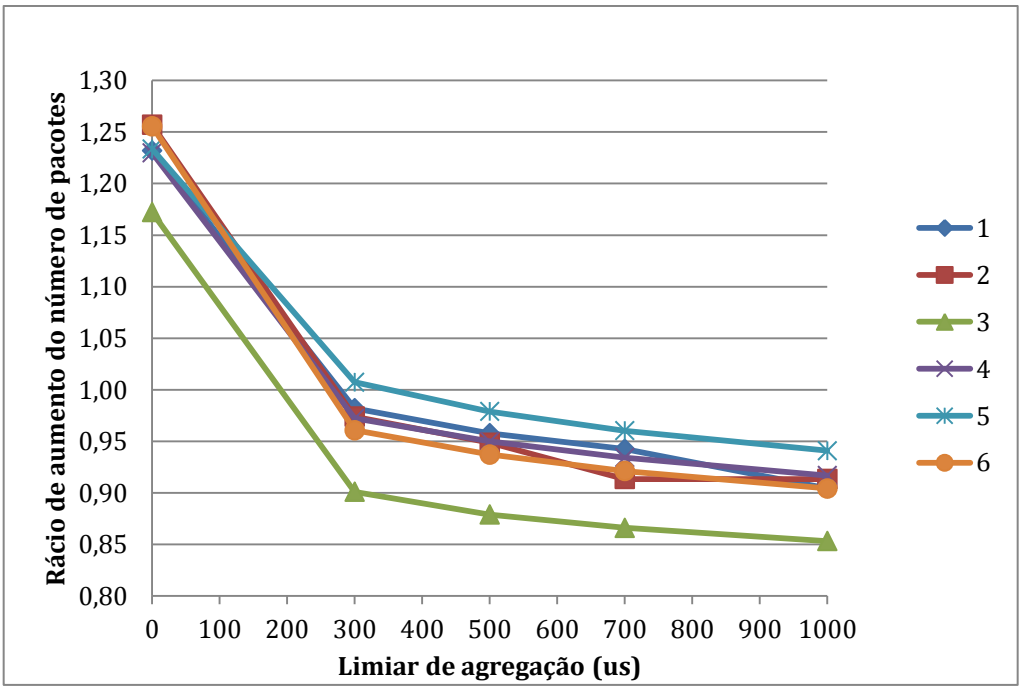

Fig. 4.3 Variação do rácio do número de pacotes produzidos sobre o número de pacotes originais pela mudança para o protocolo IPv6 face ao limiar de agregação.

Pode verificar-se que a simples troca de cabeçalho provoca, nos ficheiros selecionados, um aumento do número de pacotes que oscila entre os 17.2% e os 25.5% (dependendo do ficheiro). Se o tempo de geração do pacote for considerado como sendo de 1 ms, nesse caso poder-se-ia verificar uma diminuição do número de pacotes gerados que pode variar entre os 6% e os 15% aproximadamente. Estes dados são apresentados na tabela 4.1.

Tal como tinha sido sugerido em [2], também aqui se verifica que o tempo de geração de pacotes nas máquinas tributárias parece estar entre os 200 e os 350 µs, dependendo do ficheiro de dados, uma vez que acima desse tempo, se verifica já o fenómeno de multiplexagem estatística dos pacotes, isto é, é quando o número de pacotes gerados pelo algoritmo é inferior ao número original de pacotes.

Tabela 4.1 Variação do rácio do número de pacotes produzidos sobre o número de pacotes originais pela mudança para o protocolo IPv6 face ao limiar de agregação.

	0 (µs)	300 (µs)	500 (µs)	700 (µs)	1000 (µs)
ficheiro 201010011400	1,232042696	0,981897274	0,957751874	0,9425035	0,904426
ficheiro 201010021400	1,257061701	0,973870022	0,948570658	0,9132969	0,9132969
ficheiro 201010031400	1,1723417	0,900885958	0,878882721	0,865994	0,8530712
ficheiro 201010041400	1,229900755	0,972200519	0,949794736	0,9341182	0,9168451
ficheiro 201010051400	1,233554624	1,007330397	0,978825169	0,960202	0,94091
ficheiro 201010061400	1,25564377	0,960626038	0,937112908	0,92117	0,9042598

Tabela 4.2 Variação do rácio do número de pacotes produzidos sobre o número de pacotes originais pela mudança para o protocolo IPv6 face ao limiar de agregação.

	0 (µs)	300 (µs)	500 (µs)	700 (µs)	1000 (µs)
ficheiro 201010011400	1,153580929	1,129789531	1,127728	1,1264841	1,1251603
ficheiro 201010021400	1,054269576	1,030182522	1,0282537	1,0256459	1,0256459
ficheiro 201010031400	1,050331572	1,025274117	1,0235	1,0225035	1,0215368
ficheiro 201010041400	1,053185445	1,030591426	1,0288985	1,0277468	1,0264868
ficheiro 201010051400	1,053073317	1,033512815	1,0312943	1,0298874	1,0284649
ficheiro 201010061400	1,052494535	1,028025958	1,0263861	1,0253036	1,0241998

A figura 4.4 mostra a variação do rácio entre o número de bytes gerados e o número de bytes nos pacotes originais, em função do tempo limiar para a agregação, para cada um dos ficheiros de pacotes. Podemos ver que para a maioria dos ficheiros de dados, este aumento se situa entre os 5% e os 2%.

Para o ficheiro 201010011400 (assinalado como 1 no gráfico 4.4), apesar de ter um comportamento semelhante ao dos outros ficheiros, exibe uma maior sobrecarga resultante do processo de agregação. Por causa deste resultado, o algoritmo e os dados foram revistos e analisados, tendo as iterações seguintes da aplicação desenvolvida mostrado os mesmos resultados. Apesar da diferença ser significativa, de cerca de 10% em média, este ficheiro foi recolhido no mesmo ponto de rede, diferenciando-se dos outros apenas pela data de recolha.

Mais ainda, o rácio de número de pacotes criados / número de pacotes originais é coerente com o dos outros ficheiros. A investigação da razão do comportamento deste ficheiro não cabe no âmbito do trabalho desta investigação, e não parece adequado estar agora a fornecer uma hipótese sem que ela seja devidamente demonstrada.

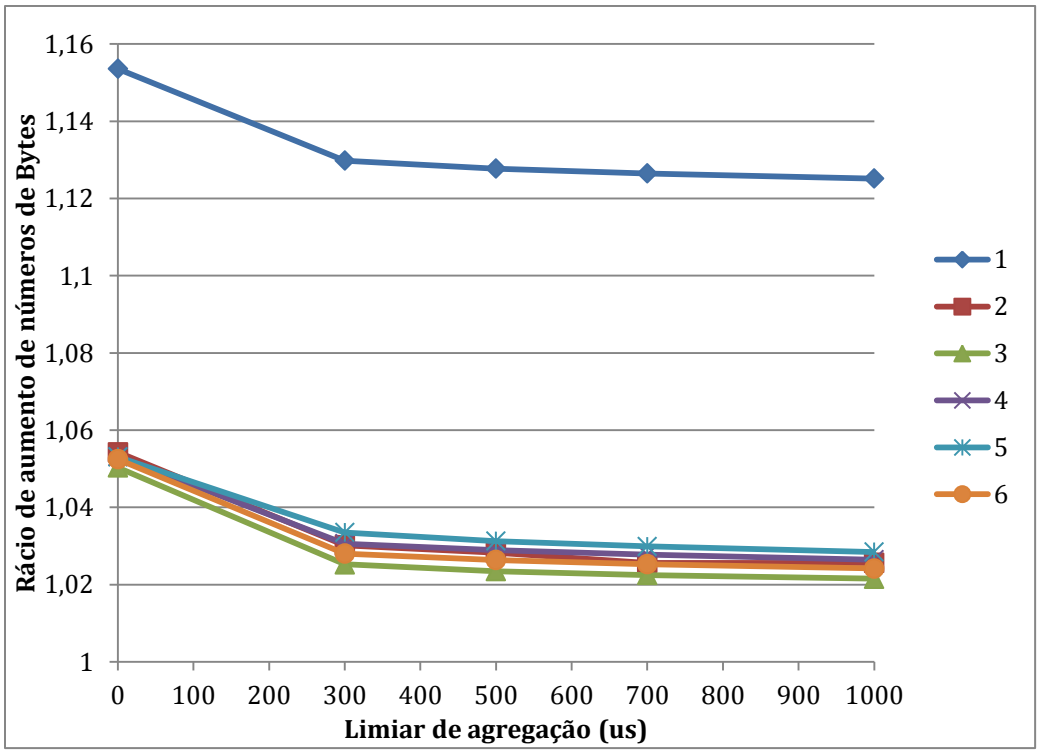

Fig. 4.4 Variação do rácio do número de bytes transmitidos sobre o número de bytes originais pela mudança para o protocolo IPv6 face ao limiar de agregação.

4.5 Conclusão

Este capítulo descreveu a aplicação usada, o método experimental e os resultados obtidos para os cerca de 77 milhões de pacotes dos 6 ficheiros processados. As conclusões dos resultados obtidos estão apresentadas no capítulo 5.

5 Conclusões

O objectivo principal deste trabalho foi responder à questão "A mudança da versão 4 para a versão 6 do protocolo IP provoca alteração no perfil do tráfego de rede na Internet?", tendo em consideração que um trabalho anterior [2] concluía que a mudança do protocolo IPv4 para o protocolo IPv6 poderia causar um aumento entre 15% e 45% no número de pacotes transmitidos na rede.

Para tal, os autores de [2] usaram um conjunto de dados de rede previamente gravados, e aplicaram um algoritmo de refabricação desse tráfego, como forma de inferir qual seria o tráfego resultante da mudança de protocolo. No entanto os autores de [2] não consideraram o efeito da duplicação dos cabeçalhos da camada de transporte.

Os autores em [2] concluíram ainda que o tempo de geração dos dados nas máquinas tributárias oscilava à volta dos 400 µs, dependendo do ficheiro de dados, sendo que nesse caso, vários ficheiros correspondiam a vários locais geográficos de registo dos pacotes.

Para este estudo foram usados ficheiros de registo de pacotes criados num ponto de agregação de tráfego intercontinental no Japão, relativos ao tráfego entre este ponto e um ponto dos Estados Unidos da América. Os ficheiros são portanto todos oriundos do mesmo local geográfico, e dadas as características de ponto de agregação, é seguro assumir que o tráfego é suficientemente heterogéneo para ser significativo. Mais ainda, o número de pacotes processados, cerca de 77 milhões, está em linha com o estudo [2], no qual foram processados 51 milhões de pacotes.

Uma das primeiras conclusões que é possível tomar é que o perfil do tráfego estudado em [2] e o perfil do tráfego aqui apresentado é diferente, concretamente, a maioria do tráfego estudado é tráfego UDP, como pode ser visto nas figuras 3.2, 3.4, 3.6, 3.8, 3.10 e 3.12. Os rácios reportados informalmente pelos autores de [2] para o tráfego então usado concluíam que em média 80% do tráfego de [2] era tráfego TCP. Uma das explicações possíveis para este facto é a crescente

importância que têm vindo a ter as comunicações de multimédia em tempo real, que geram tráfego UDP. Como se pode ver nos gráficos de análise de tráfego usado, no capítulo 3, apenas um dos ficheiros mostra predominância de tráfego TCP sobre o tráfego UDP.

A seguinte conclusão tem a ver com o tempo estimado para o que foi designado em [2] como "*same originating event*", *i.e.*, o tempo de geração dos pacotes de dados nas máquinas tributárias. Se em 2006 os dados apresentados em [2] apresentavam um tempo de geração médio próximo dos 400 µs, em 2010, o tempo de geração calculado varia entre os 200 µs e os 350 µs, aproximadamente. A conclusão possível para este resultado é que as máquinas tributárias e os equipamentos activos de rede são em 2010 mais rápidos do que os seus equivalentes em 2006, ou pelo menos, que as máquinas envolvidas neste tráfego são mais rápidas do que os seus equivalentes de 2006.

A última conclusão está directamente relacionada com a hipótese em estudo: "A mudança da versão 4 para a versão 6 do protocolo IP provoca alteração no perfil do tráfego de rede na Internet?". Para responder a esta questão, e como já foi referido anteriormente, foi implementado um algoritmo melhorado do apresentado em [2].

Os resultados obtidos confirmam parcialmente os resultados descritos em [2], uma vez que continua a verificar-se o aumento do número de pacotes gerados pela mudança do cabeçalho IPv4 para o cabeçalho IPv6. No entanto, os rácios de aumento estão longe dos 45% máximos encontrados em [2], e ficam-se entre os 17.2% e os 25.5%, variando com o ficheiro de dados. À semelhança do que foi descrito em [2] pode afirmar-se que se o tempo de geração dos pacotes de dados for mais rápido do que os estimados 200 µs ou 350 µs (variando de ficheiro para ficheiro), então a mudança do protocolo IPv4 pelo protocolo IPv6 pode causar um aumento de até 25% no número de pacotes que são transmitidos pela rede.

A conclusão final diz respeito ao número de bytes transmitidos nos dois cenários, o original com IPv4 e o refabricado com IPv6. Excluindo o caso particular do ficheiro número 1, que mostra um comportamento coerente com os dos outros ficheiros mas valores diferentes, pode afirmar-se que

é expectável um aumento do número de bytes transmitidos que pode oscilar entre os 2.2% e os 5.2%. Estes valores são compatíveis com o maior tamanho do cabeçalho IPv6.

Como trabalho futuro, fica sempre a validação destes resultados com outros ficheiros de dados capturados. No entanto, não é fácil encontrar este tipo de dados, uma vez que o foco da investigação internacional nos últimos tempos se tem virado para o tráfego das redes sem fios, sendo a área do tráfego em pontos de agregação uma área que não tem recebido a atenção que se pensa seria adequada, talvez até porque é mais difícil obter as necessárias autorizações para registar este tipo de dados. Fica ainda como trabalho futuro alargar este estudo no sentido de experimentar com diferentes tamanhos de agregação, em particular e seguindo a sugestão de [2], para cargas de 9KB e de 64KB, correspondendo ao tamanho máximo de uma Jumbo Frame Ethernet e ao potencial tamanho máximo de um pacote IP (v4 ou v6).

Referências

[1] Standard IEEE 802.3 Disponível em http://www.networksorcery.com/enp/protocol/IEEE8023.htm acedido pela última vez em 18/01/2013

[2] The Ethernet frame payload size and its effect on IPv4 and IPv6 traffic, Nuno M. Garcia, Mário M. Freire, Paulo P. Monteiro, published in proceedings of The International Conference on Information Networking 2008 (ICOIN 2008), 23 - 25 January 2008, Busan, South Korea.

[3] http://www.di.ubi.pt/~mario/files/PhDThesis-NunoGarcia.pdf

[4] Large MTUs and internet performance, Murray, D., Koziniec, T., Lee, K. and Dixon, M.W. (2012). In: IEEE HPSR 2012 - 13th IEEE Conference on High Performance Switching and Routing, 24 - 27 June, Belgrade, Serbia. http://researchrepository.murdoch.edu.au/9920/

[5] Open Systems Interconnection (OSI) Protocols http://www.cisco.com/en/US/tech/tk389/tk214/tsd_technology_support_protocol_home.html, acedido em 03/01/2013

[6] Intel and Ethernet
. http://www.intel.com/standards/case/case_ethernet.htm**,** acedido em 03/01/2013

[7] Beethoven Zanella Dias e Nilton Alves Jr. "Evolução do padrão Ethernet", 2002.

[8] http://www.csd.uoc.gr/~hy435/material/EA- Ethernet%20Jumbo%20Frames%20v0%201.pdf acedido pela última vez em 14/01/2013

[9] http://www.internet2.edu/, acedido em 05/02/2013

[10] Redes de computadores e a internet: uma abordagem top-down, James Kurose e Keith Ross 5ª edição, por editora Pearson Education - Br Ano de Edição 2010

[11] JAMHOUR, E. **IPv6 (Parte 2: Mecanismos de Transição)**, 2004. Disponível em http://www.ppgia.pucpr.br/~jamhour/Pessoal/Especializacao/Ano03/TARC/IPv6Trans.ppt. Acedido pela última vez em15/12/2012.

[12] RFC 4213 E.Nordmark, R. Gillingan. "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, Out 2005.

[13] RFC 3053 A. Durand, P.FasanoI, Guardini, CSELT S.p.A." IPv6 Tunnel Broker ", RFC 3053 January 2001.

[14] GoGo6 IPv6 Products, Community and Services, disponível em http://www.gogo6.com/freenet6, acedido pela última vez em 28/02/2012.

[15] RFC 4214 F. Templin, T. Gleeson, M. Talwar, D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 4214, October 2005.

[16] RFC 2766 G. Tsirtsis, P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT) RFC 2766"February 2000.

[17] MAWI (Measurement and Analysis on the WIDE Internet) Working Group Traffic Archive, Disponível em http://mawi.wide.ad.jp/mawi/, acedido pela última vez em 01/12/2012

[18] http://mawi.wide.ad.jp/mawi/samplepoint-F/2010/201010011400.html, acedido pela ultima vez em 23/10/2012

[19] http://mawi.wide.ad.jp/mawi/samplepoint-F/2010/201010021400.html ,acedido pela ultima vez em 06/11/2012

[20] http://mawi.wide.ad.jp/mawi/samplepoint-F/2010/201010031400.html, acedido pela ultima vez em 06/11/2012

[21] http://mawi.wide.ad.jp/mawi/samplepoint-F/2010/201010041400.html, acedido pela ultima vez em 06/11/2012

[22] http://mawi.wide.ad.jp/mawi/samplepoint-F/2010/201010051400.html, acedido pela ultima vez em 06/11/2012

[23] http://mawi.wide.ad.jp/mawi/samplepoint-F/2010/201010061400.html, acedido pela ultima vez em 06/11/2012

[24] P. Sayer, "Market Overview: US Ethernet Services," Forrester Research, 2010.

[25] S. Makineni and R. Iyer, "Architectural Characterization of TCP/IP Packet Processing on the Pentium; M Microprocessor," publicado em Proceedings of the 10th International Symposium on High Performance Computer Architecture, ser. HPCA '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 152–. [Online]. Disponível em http://dx.doi.org/10.1109/HPCA.2004.10024

[26] M. Ravot, Y. Xia, D. Nae, X. Su, H. Newman, and J. Bunn, "A Practical Approach to TCP High Speed WAN Data Transfers," publicado em Proceedings of PATHNets 2004. San Jose, CA, USA: IEEE, 2004.

APÊNDICE I

Abaixo estão as tabelas contendo os resultados numéricos devolvidos pela aplicação, e sobre os quais foram calculados os rácios apresentados no capítulo 4.

Tabela I.1 Resultados do processamento considerando tempo de agregação de 0 µs.

0 (µs)

	pacotes lidos	pacotes agregados	pacotes escritos simples	Bytes Lidos	Bytes Escritos
ficheiro 201010011400	10.158.415	0	12.515.601	6.255.909.806	7.216.698.246
ficheiro 201010021400	7.234.411	0	9.094.101	4.956.046.467	5.225.009.007
ficheiro 201010031400	7.936.605	0	9.304.413	5.491.758.948	5.768.167.808
ficheiro 201010041400	10.006.122	0	12.306.537	6.666.860.851	7.021.440.811
ficheiro 201010051400	8.902.928	0	10.982.248	5.844.234.289	6.154.407.189
ficheiro 201010061400	32.883.203	0	41.289.589	23.010.360.530	24.218.278.710

Tabela I.2 Resultados do processamento considerando tempo de agregação de 300 µs.

300 (µs)

	pacotes lidos	pacotes agregados	pacotes escritos simples	Bytes Lidos	Bytes Escritos
ficheiro 201010011400	10.158.415	3.307.679	9.974.520	6.255.909.806	7.067.861.406
ficheiro 201010021400	7.234.411	2.473.303	7.045.376	4.956.046.467	5.105.632.447
ficheiro 201010031400	7.936.605	3.277.681	7.149.976	5.491.758.948	5.630.558.308
ficheiro 201010041400	10.006.122	3.313.303	9.727.957	6.666.860.851	6.870.809.631
ficheiro 201010051400	8.902.928	2.546.678	8.968.190	5.844.234.289	6.040.091.029
ficheiro 201010061400	32.883.203	13.066.062	31.588.461	23.010.360.530	23.655.247.930

Tabela I.3 Resultados do processamento considerando tempo de agregação de 500 µs.

500 (µs)

	pacotes lidos	pacotes agregados	pacotes escritos simples	Bytes Lidos	Bytes Escritos
ficheiro 201010011400	10.158.415	3.642.460	9.729.241	6.255.909.806	7.054.964.746
ficheiro 201010021400	7.234.411	2.701.750	6.862.350	4.956.046.467	5.096.073.167
ficheiro 201010031400	7.936.605	3.524.952	6.975.345	5.491.758.948	5.620.815.408
ficheiro 201010041400	10.006.122	3.616.526	9.503.762	6.666.860.851	6.859.523.231
ficheiro 201010051400	8.902.928	2.875.600	8.714.410	5.844.234.289	6.027.125.409
ficheiro 201010061400	32.883.203	14.096.226	30.815.274	23.010.360.530	23.617.514.710

Tabela I.4 Resultados do processamento considerando tempo de agregação de 700 µs.

700 (µs)

	pacotes lidos	pacotes agregados	pacotes escritos simples	Bytes Lidos	Bytes Escritos
ficheiro 201010011400	10.158.415	3.853.979	9.574.342	6.255.909.806	7.047.182.646
ficheiro 201010021400	7.234.411	2.857.490	6.607.165	4.956.046.467	5.083.148.767
ficheiro 201010031400	7.936.605	3.666.439	6.873.052	5.491.758.948	5.615.342.848
ficheiro 201010041400	10.006.122	3.830.352	9.346.901	6.666.860.851	6.851.844.611
ficheiro 201010051400	8.902.928	3.088.996	8.548.609	5.844.234.289	6.018.903.349
ficheiro 201010061400	32.883.203	14.795.264	30.291.020	23.010.360.530	23.592.605.050

Tabela I.5 Resultados do processamento considerando tempo de agregação de 1000 µs.

1000 (µs)

	pacotes lidos	pacotes agregados	pacotes escritos simples	Bytes Lidos	Bytes Escritos
ficheiro 201010011400	10.158.415	4.077.748	9.406.742	6.255.909.806	7.038.901.346
ficheiro 201010021400	7.234.411	3.019.053	6.607.165	4.956.046.467	5.083.148.767
ficheiro 201010031400	7.936.605	3.803.106	6.770.489	5.491.758.948	5.610.033.628
ficheiro 201010041400	10.006.122	4.063.702	9.174.064	6.666.860.851	6.843.444.391
ficheiro 201010051400	8.902.928	3.306.342	8.376.854	5.844.234.289	6.010.590.009
ficheiro 201010061400	32.883.203	15.522.990	29.734.960	23.010.360.530	23.567.207.330