

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Sistema de Caracterização da Rede em Jogos 3D Online Multi-jogador

José Manuel Ilídio Queirós

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Telecomunicações

Orientador: Ricardo Santos Morla (Dr.)

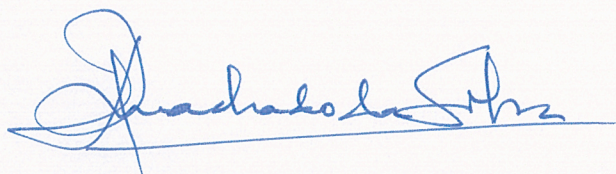
Julho de 2009

A Dissertação intitulada

“3D ONLINE MULTIPLAYER GAME NETWORK PERFORMANCE EVALUATION”

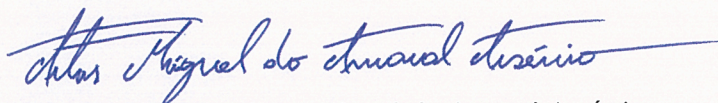
foi aprovada em provas realizadas em 17/Julho/2009

o júri



Presidente Professor Doutor José Alberto Peixoto Machado da Silva

Professor Associado do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Artur Miguel do Amaral Arsénio

Professor Auxiliar do Departamento de Engenharia Informática do Instituto Superior técnico da Universidade Técnica de Lisboa



Professor Doutor Ricardo Santos Morla

Professor Auxiliar Convidado do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.



Autor - JOSÉ MANUEL ILÍDIO QUEIRÓS

Resumo

Esta dissertação visa o estudo e elaboração de um sistema para a avaliação da performance de rede de jogos 3D *online* multi-jogador. Este sistema permitirá, a *developers*, investigadores e ISP's: 1) Efectuar a caracterização de tráfego deste tipo de jogos; 2) Permitir optimizá-los ao nível de tráfego gerado; 3) Dar a conhecer aos ISP's os padrões de tráfego gerados na rede aquando da utilização de determinado tipo de jogo, permitindo-lhes actuar no sentido de diminuir o congestionamento da internet.

Através do desenho e implementação deste sistema, nesta tese é concluído que é possível: 1) Automatizar o processo de recolha de dados de rede em diferentes sequências de acção num jogo 3D; 2) Utilizar esta ferramenta para ajudar à caracterização dos padrões de tráfego de rede deste tipo de jogos, nomeadamente relacionar padrões de tráfego com diferentes situações 3D do jogo.

Abstract

The aim of this thesis is to study and develop a system to evaluate 3D online multiplayer game network performance. This system will allow developers, researchers and ISP's: 1) Perform traffic characterization for this type of games; 2) Allow optimization of the generated traffic; 3) Allow the ISP's to know the traffic patterns generated in the network by a particular type of game, enabling them to act aiming to decrease internet congestion.

Through design and implementation of this system, this thesis concluded that it's possible to: 1) automatize the network data gathering system in different action sequences on a 3D game; 2) Use this tool to help characterize network traffic patterns for these type of games, namely to relate traffic patterns with different situations of the 3D game.

Agradecimentos

Aproveito esta oportunidade para expressar os meus sinceros agradecimentos ao Doutor Ricardo Morla pela forma como me apoiou no decorrer deste trabalho.

Ao meu colega Mário Ferreira, pela disponibilidade e apoio que sempre demonstrou no decorrer deste trabalho.

Ao Mestre Pedro Mimoso e aos pigmeus que deambulam na sua mente, pelo companheirismo, amizade, e acima de tudo pela insanidade que nos é comum.

Aos meus avós, José Queirós e Rosa Ribeiro, pelo apoio e carinho incondicionais.

À minha irmã, Raquel Queirós, cujo amor sempre fez de mim uma pessoa melhor.

Aos meus pais, Aurélio e Rosalina Queirós, sem os quais nada disto seria possível. O meu muito obrigado pelo seu amor, carinho, apoio e compreensão sem fim.

À Piu, cuja ternura e amor fizeram com que este trabalho fosse diariamente mais fácil e alegre.

José Manuel Ilídio Queirós

*“The brain is a wonderful organ.
It starts working the moment you get up in the morning
and does not stop until you get into the office.”*

Robert Frost (1874 - 1963)

Conteúdo

1	Introdução	1
1.1	Estrutura da Dissertação	3
2	Estado da Arte	5
2.1	Contextualização	5
2.1.1	Rede	5
2.1.2	Protocolos de comunicação	6
2.1.3	Arquitecturas de rede	7
2.2	Caracterização de tráfego de jogos <i>online</i>	11
2.2.1	Pré-MMOG (2003-2005)	11
2.2.2	Pós-MMOG (2005-2008)	12
3	Arquitectura	15
3.1	Caso de utilização	15
3.2	Definição da arquitectura	15
3.2.1	Elementos	16
3.2.2	Topologias lógicas de conexão	17
3.2.3	Automatização do processo	18
3.3	Conclusão	19
4	Implementação	21
4.1	Primeira fase	21
4.1.1	Elementos	21
4.1.2	Topologia lógica de conexão	22
4.1.3	Configuração e execução	22
4.1.4	Resultados e conclusões	24
4.2	Automatização	25
4.2.1	Elementos	25
4.2.2	Arquitectura	26
4.2.3	Implementação	27
4.3	Resumo	28
5	Validação	31
5.1	Validação do funcionamento do <i>SLNT</i>	31
5.1.1	Método	31
5.1.2	Condições de teste	32
5.1.3	Conclusão	33
5.2	Potencialidade de utilização	33

5.2.1	Casos de Uso	33
5.2.2	Processamento de dados	36
5.2.3	Análise dos resultados	42
5.3	Conclusões	42
6	Conclusões e Trabalho Futuro	43
6.1	Satisfação dos Objectivos	43
6.2	Trabalho Futuro	43
A	Tabelas das diferenças entre os histogramas de cada repetição dos testes repetitivos	45
A.1	<i>Script Back</i>	46
A.2	<i>Script Fly</i>	49
A.3	<i>Script Flyto</i>	52
A.4	<i>Script Forward</i>	55
A.5	<i>Script Inout</i>	58
A.6	<i>Script Inwaitout</i>	61
A.7	<i>Script Left</i>	64
A.8	<i>Script Right</i>	67
	Referências	71

Lista de Figuras

2.1	Arquitectura <i>Peer-to-Peer</i>	8
2.2	Arquitectura Cliente-Servidor.	9
2.3	Arquitectura Híbrida.	10
3.1	Caso de utilização.	16
3.2	Topologia: Cliente(modificado) <-> Servidor.	17
3.3	Topologia: Cliente <-> Servidor(modificado).	18
3.4	Topologia: Cliente <-> Observador de tráfego <-> Servidor.	19
3.5	Topologia: Cliente <-(Observador de tráfego)-> Servidor.	20
4.1	Topologia lógica de conexão.	22
4.2	<i>Screenshot</i> do ambiente de trabalho na altura do teste.	24
4.3	Funcionamento do <i>SLNT</i>	25
4.4	<i>Frontend</i> do <i>SLNT</i>	27
5.1	<i>Screenshots</i> do <i>SLNT</i> configurado para execução dos dois <i>scripts</i>	32
5.2	<i>Screenshot</i> do terreno da zona virtual.	33
5.3	Histograma do tamanho das frames (TCP) do teste “long-inwaitout”.	37
5.4	Histograma do tamanho das frames (TCP) do teste “long-flywait”.	37
5.5	Histograma do tamanho das frames (TCP) do teste “long-walk”.	37
5.6	Histograma do tamanho das frames (TCP) do teste “long-flyto”.	38
5.7	Histograma do tamanho das frames (UDP) do teste “long-inwaitout”.	38
5.8	Histograma do tamanho das frames (UDP) do teste “long-flywait”.	38
5.9	Histograma do tamanho das frames (UDP) do teste “long-walk”.	38
5.10	Histograma do tamanho das frames (UDP) do teste “long-flyto”.	38
5.11	Histograma comparativo do tamanho das frames (TCP) dos quatro testes.	39
5.12	Histograma comparativo do tamanho das frames (UDP) dos quatro testes.	39
5.13	Histograma comparativo do intervalo de tempo entre chegada das frames (TCP) dos quatro testes.	39
5.14	Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-inwaitout”.	40
5.15	Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-flywait”.	40
5.16	Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-walk”.	40
5.17	Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-flyto”.	41

Lista de Tabelas

2.1	Modelo de camadas TCP/IP.	6
2.2	Comparação entre TCP e UDP.	7
2.3	Protocolo de transporte utilizado em cada tipo de jogo.	7
4.1	Tabela de comandos de movimentos do <i>avatar</i>	29
4.2	Tabela de comandos relacionados com <i>Testclient</i>	29
A.1	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “back” obtidos nas diferentes repetições.	46
A.2	Diferenças relativas entre os histogramas de tamanhos (TCP) de “back” obtidos nas diferentes repetições.	46
A.3	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “back” obtidos nas diferentes repetições.	47
A.4	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “back” obtidos nas diferentes repetições.	47
A.5	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “back” obtidos nas diferentes repetições.	47
A.6	Diferenças relativas entre os histogramas de tamanhos (UDP) de “back” obtidos nas diferentes repetições.	48
A.7	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “back” obtidos nas diferentes repetições.	48
A.8	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “back” obtidos nas diferentes repetições.	48
A.9	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “fly” obtidos nas diferentes repetições.	49
A.10	Diferenças relativas entre os histogramas de tamanhos (TCP) de “fly” obtidos nas diferentes repetições.	49
A.11	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “fly” obtidos nas diferentes repetições.	50
A.12	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “fly” obtidos nas diferentes repetições.	50
A.13	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “fly” obtidos nas diferentes repetições.	50
A.14	Diferenças relativas entre os histogramas de tamanhos (UDP) de “fly” obtidos nas diferentes repetições.	51
A.15	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “fly” obtidos nas diferentes repetições.	51

A.16	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “fly” obtidos nas diferentes repetições.	51
A.17	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “flyto” obtidos nas diferentes repetições.	52
A.18	Diferenças relativas entre os histogramas de tamanhos (TCP) de “flyto” obtidos nas diferentes repetições.	52
A.19	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “flyto” obtidos nas diferentes repetições.	53
A.20	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “flyto” obtidos nas diferentes repetições.	53
A.21	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “flyto” obtidos nas diferentes repetições.	53
A.22	Diferenças relativas entre os histogramas de tamanhos (UDP) de “flyto” obtidos nas diferentes repetições.	54
A.23	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “flyto” obtidos nas diferentes repetições.	54
A.24	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “flyto” obtidos nas diferentes repetições.	54
A.25	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “forward” obtidos nas diferentes repetições.	55
A.26	Diferenças relativas entre os histogramas de tamanhos (TCP) de “forward” obtidos nas diferentes repetições.	55
A.27	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “forward” obtidos nas diferentes repetições.	56
A.28	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “forward” obtidos nas diferentes repetições.	56
A.29	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “forward” obtidos nas diferentes repetições.	56
A.30	Diferenças relativas entre os histogramas de tamanhos (UDP) de “forward” obtidos nas diferentes repetições.	57
A.31	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “forward” obtidos nas diferentes repetições.	57
A.32	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “forward” obtidos nas diferentes repetições.	57
A.33	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “inout” obtidos nas diferentes repetições.	58
A.34	Diferenças relativas entre os histogramas de tamanhos (TCP) de “inout” obtidos nas diferentes repetições.	58
A.35	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “inout” obtidos nas diferentes repetições.	59
A.36	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “inout” obtidos nas diferentes repetições.	59
A.37	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “inout” obtidos nas diferentes repetições.	59
A.38	Diferenças relativas entre os histogramas de tamanhos (UDP) de “inout” obtidos nas diferentes repetições.	60
A.39	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “inout” obtidos nas diferentes repetições.	60

A.40	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “inout” obtidos nas diferentes repetições.	60
A.41	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “inwaitout” obtidos nas diferentes repetições.	61
A.42	Diferenças relativas entre os histogramas de tamanhos (TCP) de “inwaitout” obtidos nas diferentes repetições.	61
A.43	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “inwaitout” obtidos nas diferentes repetições.	62
A.44	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “inwaitout” obtidos nas diferentes repetições.	62
A.45	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “inwaitout” obtidos nas diferentes repetições.	62
A.46	Diferenças relativas entre os histogramas de tamanhos (UDP) de “inwaitout” obtidos nas diferentes repetições.	63
A.47	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “inwaitout” obtidos nas diferentes repetições.	63
A.48	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “inwaitout” obtidos nas diferentes repetições.	63
A.49	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “left” obtidos nas diferentes repetições.	64
A.50	Diferenças relativas entre os histogramas de tamanhos (TCP) de “left” obtidos nas diferentes repetições.	64
A.51	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “left” obtidos nas diferentes repetições.	65
A.52	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “left” obtidos nas diferentes repetições.	65
A.53	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “left” obtidos nas diferentes repetições.	65
A.54	Diferenças relativas entre os histogramas de tamanhos (UDP) de “left” obtidos nas diferentes repetições.	66
A.55	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “left” obtidos nas diferentes repetições.	66
A.56	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “left” obtidos nas diferentes repetições.	66
A.57	Diferenças absolutas entre os histogramas de tamanhos (TCP) de “right” obtidos nas diferentes repetições.	67
A.58	Diferenças relativas entre os histogramas de tamanhos (TCP) de “right” obtidos nas diferentes repetições.	67
A.59	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “right” obtidos nas diferentes repetições.	68
A.60	Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (TCP) de “right” obtidos nas diferentes repetições.	68
A.61	Diferenças absolutas entre os histogramas de tamanhos (UDP) de “right” obtidos nas diferentes repetições.	68
A.62	Diferenças relativas entre os histogramas de tamanhos (UDP) de “right” obtidos nas diferentes repetições.	69
A.63	Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de <i>frames</i> , (UDP) de “right” obtidos nas diferentes repetições.	69

A.64 Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “right” obtidos nas diferentes repetições. 69

Abreviaturas e Símbolos

3D	<i>Three Dimensional; Três dimensões</i>
ADSL	<i>Asymmetric Digital Subscriber Line</i>
CSV	<i>Comma Separated Values</i>
FPS	<i>First Person Shooter</i>
ISP	<i>Internet Service Provider</i>
LAN	<i>Local Area Network</i>
MMO	<i>Massive Multiplayer Online</i>
MMOG	<i>Massive Multiplayer Online Game</i>
MMORPG	<i>Massive Multiplayer Online Role-Playing Game</i>
P2P	<i>Peer-to-Peer</i>
RAM	<i>Random Access Memory</i>
RDIS	<i>Rede Digital Integrada de Serviços</i>
RPG	<i>Role Playing Game</i>
RTS	<i>Real-Time Strategy</i>
SLNT	<i>Second Life Network Test</i>
SVN	<i>Subversion</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
URI	<i>Uniform Resource Identifier</i>
UUID	<i>Universally Unique Identifier</i>
WAN	<i>Wide Area Network</i>

Capítulo 1

Introdução

Pôr em jogo, em sentido figurado, significa funcionamento, movimentação, agitação que conjugados, combinados e harmonizados produzem um determinado efeito nos diferentes actores.

Actualmente com o avanço da tecnologia, jogo já não é só a acareação, com ou sem regras, entre os diferentes intervenientes mas sim um universo onde os jogos especializados ou genéricos abarcam situações diferenciadas num mundo comunicacional lúdico.

Entre os vários tipos de jogos de computador podem ser identificados jogos de: Acção, Aventura, Música, *Role-Playing*¹ (RPG), Simulação, Desporto, Estratégia, *Massively Multiplayer Online* (MMO). De referir que os tipos de jogos, hoje em dia, já se encontram relacionados entre si e não existem categorias estanques (Acção-Aventura, MMO-RPG e outros), estando quase todos relacionados com a componente *online*.

Os jogos *online* estão a tornar-se aplicações de larga escala, altamente interactivos e baseados em mundos virtuais.

A popularidade dos jogos aumenta e diminui em questão de meses com novos jogos que facilmente substituem os anteriores, tornando-os imprevisíveis do ponto de vista de recursos de rede globais necessários. Esta característica, intrínseca aos jogos, torna a optimização de arquitecturas e protocolos de rede, uma tarefa inconclusiva no tempo. É portanto necessário um acompanhamento e evolução constantes das redes para que os utilizadores obtenham uma melhor *performance* do jogo independentemente da localização geográfica, da distância ao servidor (se existir) e do número de jogadores activos em determinado espaço de tempo.

Nesta nova geração de jogos *online*, são utilizadas de forma intensiva, tecnologias de redes para ligar vários jogadores em simultâneo em qualquer parte do globo. Toda esta nova dimensão apenas foi possível devido à Internet. Este conglomerado de redes em escala mundial de milhões de computadores interligados pelo Protocolo de Internet que permite o acesso a informações e todo tipo de transferência de dados (Wikipedia, 2009; de competência malha atlântica, 2005), veio disponibilizar a intercomunicação global ao alcance de todos.

¹Jogador desempenha o papel de uma personagem

A Internet é extremamente heterogénea e complexa. Formada por redes e sub-redes hierárquicas, esta assume comportamentos diferenciados em cada nó e ligação ponto a ponto (Wikipedia, 2009; de competência malha atlântica, 2005).

Com toda a envolvimento dos meios necessários a uma melhor performance ao nível do desempenho das redes, são notórias algumas situações inadequadas. Com o objectivo de melhorar o desempenho de todo o sistema *online* de jogos, neste trabalho surgem questões como:

- Quais os protocolos e arquitecturas de rede usados?
- Quais as componentes de jogo relacionadas com recursos de rede?
- O que interessa monitorar, e como fazê-lo, tanto em clientes como em servidores?
- O que interessa controlar, e como fazê-lo, tanto em clientes como em servidores?
- Como avaliar a performance dos jogos e cenários escolhidos e assim validar as optimizações?

Assim através destas interrogações foi feito um trabalho de pesquisa, avaliação e implementação de um *Sistema de caracterização da rede em jogos 3D online multi-jogador*, que permite obter informações de tráfego demonstrativas do desempenho dos jogos.

A fim de realizar a implementação de um sistema deste género, foi tomado como objecto de estudo o *Second Life* (Life, 2009). *Second Life* é um mundo virtual 3D imaginado e criado pelos seus utilizadores. Permite através do software cliente, andar, "teletransportar-se" e até voar até certos locais do mundo virtual. Permite ainda usar voz e texto para comunicar com outras pessoas de todo o mundo. Life (2009)

Para criar um sistema que permita a caracterização de tráfego do *Second Life* várias questões surgem e são respondidas ao longo deste documento:

- Como definir a arquitectura deste sistema?
- Que elementos formam o sistema?
- Como interligar esses mesmos elementos?
- Como tornar o sistema simples, fácil de usar, e eficaz?
- Como validar o sistema?
- As informações que o sistema fornece são úteis?

Com estas questões em mente foi desenvolvido um *Sistema de caracterização da rede em jogos 3D online multi-jogador*. De modo a validar este sistema, foram feitos testes funcionais afim de demonstrar o seu correcto funcionamento, e procedeu-se também à sua utilização para efectuar a caracterização de tráfego de situações específicas do *Second Life*.

1.1 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 5 capítulos. No capítulo 2 é descrito o estado da arte e são apresentados trabalhos relacionados. No capítulo 3 é apresentado o estudo da arquitectura de sistema. No capítulo 4 é descrita a implementação passo-a-passo da arquitectura definida no capítulo 3. No capítulo 5 procede-se à validação de todo o sistema através de testes e observação de seus resultados. No capítulo 6 são apresentadas as conclusões e o trabalho futuro.

Capítulo 2

Estado da Arte

2.1 Contextualização

2.1.1 Rede

A Internet é constituída por milhões de computadores interligados. Estão ligadas à internet diferentes tipos de entidades com diferentes tipos de utilização, disponibilizando e/ou acedendo a serviços/recursos da mesma.

Entidades que sejam responsáveis por servidores, como universidades, empresas, organizações, normalmente possuem ligações permanentes, já que disponibilizam recursos acessíveis 24 horas por dia.

O utilizador comum efectua o seu acesso à Internet através do seu *Internet Service Provider* (ISP) via ADSL, RDIS, internet por cabo, ou outro, podendo estar ligado temporariamente ou permanentemente. Mesmo em acesso permanente, na maioria dos casos, faz uso temporário do acesso aos recursos da internet (acesso a páginas *web*, jogos *online* e outros).

Com os avanços tecnológicos e a proliferação das redes assiste-se a um contínuo alargamento de banda disponível aos utilizadores. No entanto o acesso individual de cada utilizador não é uniforme, isto é, cada utilizador tem características de acesso variáveis — largura de banda disponível conforme o acesso à internet que possui; atrasos de transmissão variáveis devido ao congestionamento imprevisível de determinadas partes da rede, entre outros factores.

Do tráfego total da internet, uma fatia significativa é ocupada pelos jogos *online* multi-jogador. Prevê-se um contínuo aumento tanto na complexidade como na dimensão deste tipo de jogos pelo que é necessário estudá-los e aperfeiçoá-los para não congestionar o tráfego no espaço cibernético.

Dentro da categoria de jogos multi-jogador existem os denominados *Massive Multiplayer Online Games* (MMOG), que permitem a interacção, colaboração e competição entre jogadores a uma escala global usando, por necessidade, a internet (exemplos: World of Warcraft (Blizzard Entertainment, 2009), Second Life (Life, 2009), Entropia Universe (AB, 2009), Lineage 2 (NCSoft, 2009)). É nesta categoria que a avaliação e optimização de performance de rede são importantes

pois o número de utilizadores é enorme e muito facilmente se pode chegar a situações de saturação do sistema de cada jogo — World of Warcraft atingiu 11 milhões de utilizadores registados em outubro de 2008 (Entertainment, 2009); Lineage II tinha, à data de 29 de Abril de 2009, mais de 19 milhões de clientes a nível mundial (Corporation, 2009); *Second Life* atinge, no primeiro trimestre de 2009, o pico de 88200 utilizadores concorrentes (utilizadores activos no sistema ao mesmo tempo) (Linden Research, 2009).

Pelo exposto, importa estudar não só o meio pelo qual os jogos comunicam (Internet), mas também as suas características tais como protocolos e arquitecturas de rede.

2.1.2 Protocolos de comunicação

A interconexão computacional funciona devido à utilização de protocolos que permitem às máquinas terem uma “linguagem” comum de comunicação. A Internet utiliza o modelo TCP/IP de encapsulamento, que busca fornecer uma abstracção aos protocolos e serviços para diferentes camadas de uma pilha de estruturas de dados (Comer, 2000). Na tabela 2.1 pode-se ver o modelo de camadas TCP/IP.

Tabela 2.1: Modelo de camadas TCP/IP.

Nº	Camada	Descrição	Utilização no contexto de jogo online
5	Aplicação	Camada controlada pelas aplicações. (http, ftp, dns)	Protocolo criado por cada jogo.
4	Transporte	Protocolos utilizados para transportar a informação das aplicações entre pontos da rede (TCP, UDP, RTP, SCTP).	TCP, UDP, conforme o tipo de jogo.
3	Rede	Protocolo que permite a interconexão entre várias redes.	Protocolo IP, utilizado na internet.
2	Lógica	Esquema lógico utilizado pela tecnologia de acesso (Ethernet, Wi-Fi).	
1	Física	Ligação física feita por modems e outros.	

Conforme o tipo de jogo, são usados os protocolos de transporte TCP ou UDP. cabeçalho do pacote é de menor tamanho O UDP não suporta ordenação de pacotes¹, não garante a entrega dos mesmos e o seu cabeçalho é de menor tamanho (mais rápido, menos estável).

O TCP, por outro lado, ordena os pacotes, garante a entrega e a não duplicação dos mesmos, e tem um cabeçalho de maior tamanho (menos rápido, mais estável). Na tabela 2.2 é feita uma comparação qualitativa entre o TCP e UDP.

Os jogos podem ser divididos nas seguintes categorias:

- *Turn-based games* (jogadores fazem determinadas acções um de cada vez)

¹Pedaço de informação transmitida, constituída, na maioria dos casos, por cabeçalho e dados a transmitir.

Tabela 2.2: Comparação entre TCP e UDP.

	TCP	UDP
Garantia de entrega	Sim	Não
Ordenação de pacotes	Sim	Não
Estabilidade	Maior	Menor
Tamanho de cabeçalho	Grande	Pequeno
Suporte a tempo-real	Menor	Maior

- *Realtime Strategy games* (jogos de estratégia em tempo real)
- *Role-playing games* (jogador desempenha o papel de uma personagem)
- *Action games* (jogos de acção como first person shooters e jogos de corridas)

Cada categoria de jogo utiliza o protocolo de transporte que lhe é mais conveniente. Na tabela 2.3 está descrito o protocolo utilizado em cada tipo de jogo.

Tabela 2.3: Protocolo de transporte utilizado em cada tipo de jogo.

Tipo de jogo	Protocolo
<i>Turn-based</i>	TCP
<i>Realtime Strategy</i>	TCP (UDP apenas em casos extremos)
<i>Role-playing</i>	UDP - jogos com muitos objectos e interacções TCP - jogos mais lentos como MUDs
<i>Action</i>	UDP

2.1.3 Arquitecturas de rede

Clientes e servidores podem ligar-se utilizando lógicas de ligação chamadas arquitecturas de rede. Destacam-se:

- *Peer-to-Peer* (P2P)
- Cliente-Servidor
- Arquitecturas híbridas

2.1.3.1 *Peer-to-Peer*

Nesta topologia cada utilizador estabelece uma conexão com todos os outros utilizadores (Figura 2.1).

- **Prós:**
 - Clientes obtêm um bom tempo de resposta, pois estão ligados directamente ao destino.

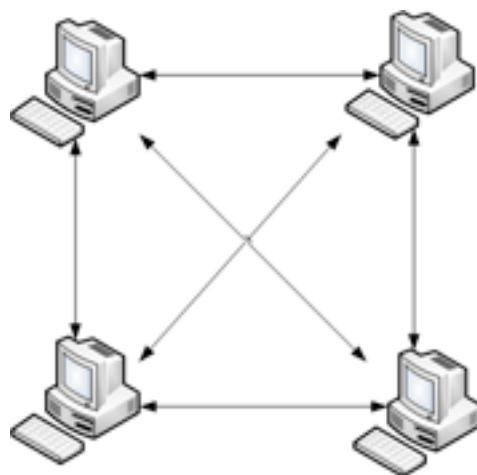


Figura 2.1: Arquitectura *Peer-to-Peer*.

- Tem um bom potencial de gestão já que cada cliente pode gerir o seu tráfego pois possui múltiplas conexões (uma a cada utilizador do sistema).

- **Contras:**

- Comunicação redundante – a mesma informação, como o estado do jogo, é replicada pela rede múltiplas vezes pois cada utilizador tem que enviar a informação do seu estado a todos os intervenientes
- Pouco escalável – cada novo utilizador exige $N-1$ novas ligações (N : Número actual de utilizadores do sistema)
- Pobre controlo administrativo – não existe possibilidade de controlo centralizado.

Um dos projectos que utiliza esta arquitectura de rede é o *Open Cobalt* (University et al., 2009). Esta aplicação funciona exclusivamente como software cliente, utilizando um protocolo que lhe permite: 1) trocar informação directamente com os outros softwares cliente; 2) manter um estado consistente do sistema em todos os clientes, sem necessitar de um servidor.

2.1.3.2 Cliente-Servidor

Na lógica cliente-servidor, existe um servidor central, que gere toda a informação e a disponibiliza a cada cliente ligado a este (Figura 2.2).

- **Prós:**

- Razoavelmente escalável – um novo utilizador apenas exige uma nova ligação ao servidor.
- Possível controlo administrativo – permite uma gestão centralizada do sistema já que todas as ligações passam pelo servidor.

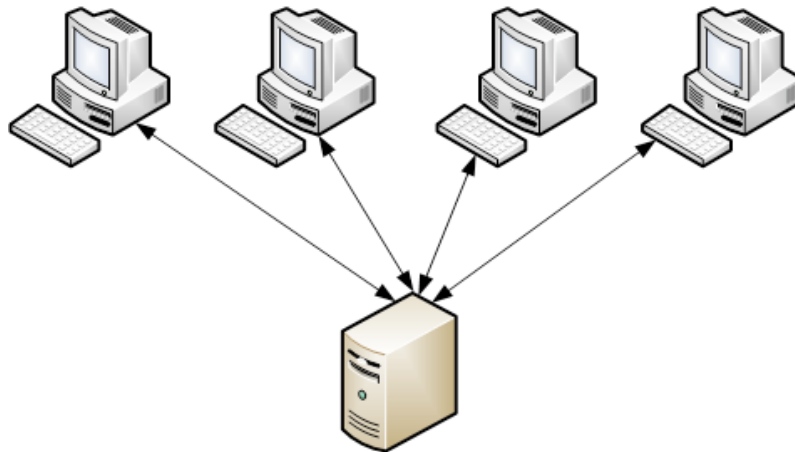


Figura 2.2: Arquitectura Cliente-Servidor.

- **Contras:**

- Servidor ou rede, rapidamente se tornam num *bottleneck*² – com o crescimento do número de utilizadores, o servidor vai ficando cada vez mais sobrecarregado em termos de processamento e largura de banda necessária, fazendo com que se torne o principal factor de atraso nas comunicações. Para resolver o problema de processamento são utilizados frequentemente *clusters*³ de servidores, tornando a rede o único *bottleneck* desta arquitectura.
- Topologia estática – só existe um local onde conectar.
- Exige posicionamento geográfico em relação aos clientes – com esta topologia é impossível obter bons desempenhos se os clientes estiverem dispersos geograficamente, pois as distâncias variáveis introduzem atrasos variáveis.

Refere-se como exemplo de utilização desta arquitectura, os projectos *Open Croquet* (The Croquet Consortium, 2009) e *Second Life* (Life, 2009). Ambos utilizam software cliente que necessita ligar-se a um servidor para comunicar com todo o sistema.

2.1.3.3 Arquitecturas híbridas

Como exemplo de arquitectura híbrida menciona-se uma *Mirrored Server Architecture*. Esta arquitectura consiste em múltiplas arquitecturas Cliente-Servidor interconectadas por uma arquitectura *Peer-to-Peer* (Figura 2.3).

- **Prós:**

²*Bottleneck* é um fenómeno onde o desempenho e a capacidade de um sistema é limitado por um certo componente.

³Um *cluster* é formado por um conjunto de computadores ligados em rede que, utilizando um sistema operacional distribuído, trabalham como se fossem uma única máquina de grande porte.

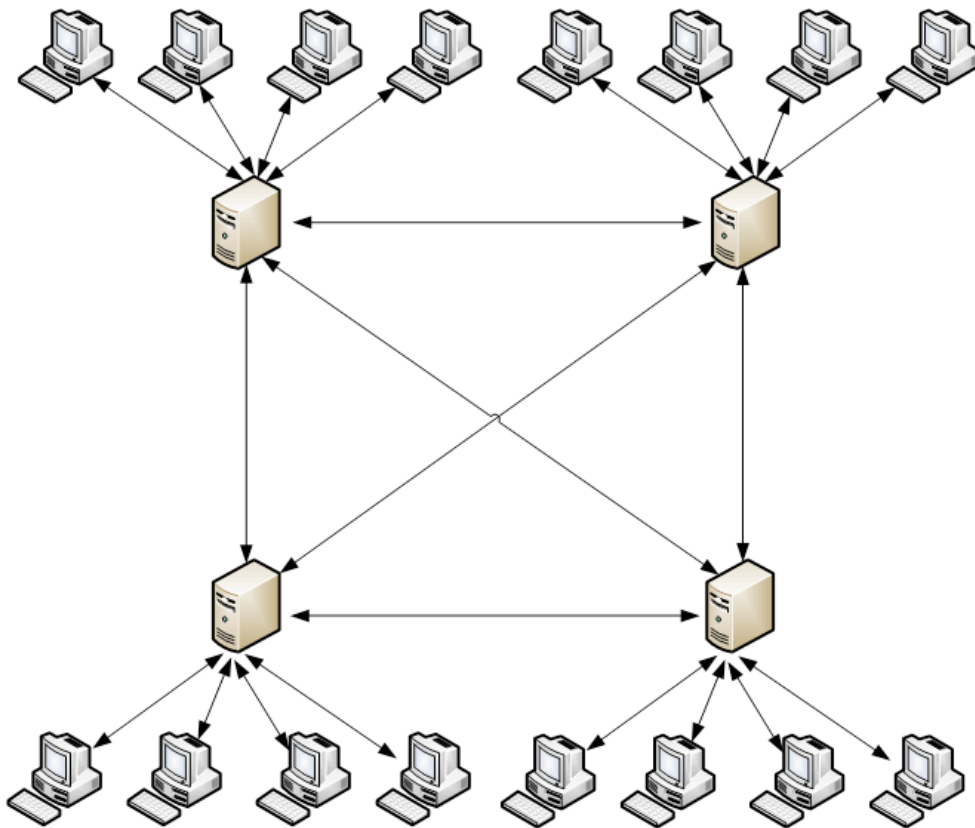


Figura 2.3: Arquitectura Híbrida.

- Escalável — um novo grupo de utilizadores próximo apenas necessita que estes se liguem a um servidor próximo; um novo grupo de utilizadores, distante, exige apenas a disponibilização de um servidor central para a área geográfica em questão.
- Uniforme em resposta temporal, pois as ligações são efectuadas ao servidor mais próximo.
- Permite controlo administrativo – é possível controlar cada servidor e através destes controlar todo o sistema.

• **Contras:**

- Comunicação redundante – advém da parte *peer-to-peer*.
- Topologia estática – não existe qualquer tipo de conexões dinâmicas pois os clientes ligam-se sempre ao servidor mais próximo.
- Inconsistência nos estados dos jogos, nos servidores – como o jogo é gerido por múltiplos servidores, o estado momentâneo do jogo nos múltiplos servidores é diferente.

Existem já estudos sobre a utilização deste tipo de arquitecturas em jogos, como por exemplo em (Jared Jardine, 2008).

2.2 Caracterização de tráfego de jogos *online*

Jogos de computador eram no início isolados, sem comunicação. A única maneira de ter mais do que um jogador a jogar simultaneamente (na maioria das vezes com um máximo de dois jogadores) era estarem fisicamente no mesmo local e jogarem no mesmo terminal (computador pessoal ou consola de jogos). Com o aparecimento das redes de computadores, começaram a aparecer jogos que permitiam a participação simultânea de vários jogadores. Eram jogados em *Local Area Networks* (LAN) pelo que era necessário que os terminais (habitualmente um por jogador) estivessem interligados na mesma rede local (normalmente através de um *hub* ou *switch*). Entre vários tipos de jogos, refere-se os *First Person Shooters* (FPS) e *Real Time Strategy* (RTS). Foi então que se deu o grande passo nas redes de computadores e apareceu a internet. Com a internet tornou-se possível a comunicação entre jogadores em diferentes pontos do globo, fazendo aparecer os jogos *online* multi-jogador. Entre as várias categorias já existentes, o mundo *online* deu lugar aos *Massive Multiplayer Online Games* (MMOG).

Ao longo de toda esta evolução e a partir do momento em que os jogos passaram a comunicar entre si, começaram a surgir estudos acerca do comportamento dos mesmos tanto a nível de caracterização de tráfego como a nível de impacto de certas características da rede, no desempenho dos jogadores. A pesquisa feita neste âmbito pode ser dividida em dois blocos cronológicos: o primeiro no período 2003/05 (pré-MMOG) (Nathan Sheldon, 2003; Tanja Lang, 2004; James Nichols, 2004; Tom Beigbender, 2004; Claypool, 2005; Wu-chang Feng, 2005); o segundo no período 2005/08 (pós-MMOG) (Kuan-Ta Chen, 2005; Jaecheol Kim, 2005; Stênio Fernandes, 2007; James Kinicki, 2008; Rafael Antonello, 2008).

2.2.1 Pré-MMOG (2003-2005)

Em (Nathan Sheldon, 2003) (2003), é feito um estudo do efeito da latência da rede, na performance do jogador de um jogo RTS. Este estudo é feito numa LAN e contém um elemento que controla a latência entre vários pontos da rede (Para efectuar este controle utilizam uma ferramenta chamada *NIST Net*). Para atingir o seu objectivo, variaram a latência em múltiplas situações de jogo, e observaram o desempenho do jogador. Em (Tanja Lang, 2004) (2004), é feita uma caracterização de tráfego de um jogo FPS. Neste estudo é montada uma LAN com múltiplos clientes e um servidor. Durante vários meses o sistema foi utilizado pela comunidade universitária, o que permitiu aos autores fazer o estudo de uma situação real de utilização dos clientes do jogo. Para analisar o tráfego instalaram no servidor uma ferramenta (*pkthisto*) que lhes retornava para cada direcção do tráfego (cliente->servidor e servidor->cliente), o tamanho dos pacotes, a diferença temporal de chegada dos pacotes, o número de pacotes por segundo e o número de bits por segundo. A ferramenta gerava ainda os histogramas, a cada 2000 pacotes, das informações de tráfego respeitantes a UDP. Com esta configuração obtiveram uma caracterização do tráfego gerado ao longo do tempo pelo sistema tanto no sentido ascendente (cliente->servidor) como no sentido descendente (servidor->cliente). Em (James Nichols, 2004) (2004), é feito um estudo do efeito da latência num jogo *online*. Como método de teste, ligam dois terminais ao servidor oficial do jogo, através da

internet. Numa das ligações colocam um router com a aplicação *NIST Net* a controlar a latência de um dos terminais. Através deste esquema estudam uma técnica de compensação da latência, o impacto da latência na *performance* do jogador e fazem uma caracterização do tráfego gerado nas situações de latência induzida e sem latência. Em (Tom Beigbeder, 2004) (2004), é feito um estudo dos efeitos de perda⁴ e de latência na *performance* dos jogadores num jogo de FPS. Utilizam uma arquitectura controlada já que fazem os testes utilizando uma LAN, clientes e um servidor próprios. Tal como em estudo prévios, utilizam o *NIST Net* para controlar a latência entre os clientes e o servidor, e com a variação dessa variável verificam o impacto da perda e da latência no desempenho dos jogadores. Em (Claypool, 2005) (2005), é efectuado mais um estudo dos efeitos da latência no desempenho dos jogadores, em vários jogos RTS. Mais uma vez os testes são efectuados numa LAN, desta vez composto apenas por dois terminais e um *proxy* com *NIST Net* instalado. Neste estudo varia-se a latência e relaciona-se essa variação com o desempenho do jogador em múltiplas situações do jogo. Neste documento é feita ainda a caracterização do tráfego gerado. Ainda em 2005 surge mais um artigo sobre caracterização de múltiplos jogos *online* (Wu-chang Feng, 2005). Como método criaram o próprio servidor do jogo e aguardaram pela sua utilização por parte de jogadores de todo o mundo. A procura de servidores dos jogos estudados era tão grande que rapidamente ficavam com o servidor totalmente ocupado. Através desta abordagem obtiveram informações como largura de banda ocupada, tempo de ligação por parte de cada jogador, tamanho dos pacotes recebido e enviados. O facto de terem controle sobre o servidor e de estarem a estudar uma situação real permitiu-lhes ainda obter a localização dos jogadores em cada período do dia.

Através destes artigos pode-se concluir que a maioria dos estudos feitos até 2004 eram direccionados ao estudo do impacto da latência no desempenho dos jogadores (os jogos estudados eram na maioria FPS's e RTS's que são jogos que dependem muito da interacção entre jogadores em tempo real). Grande parte dos testes eram efectuados em LAN's e a maioria das caracterizações de tráfego eram efectuadas sem ter em conta a sua relação com as acções no jogo. Já em 2005 aparecem dois artigos acerca de jogos *online* (James Nichols, 2004; Wu-chang Feng, 2005) mas as caracterizações de tráfego continuam independentes das acções nos jogos. De notar que em todos os artigos os testes eram feitos manualmente, sem qualquer componente automática de teste.

2.2.2 Pós-MMOG (2005-2008)

2.2.2.1 *Massive Multiplayer Online Role-Playing Games* (MMORPGs)

A partir de 2005, começaram a surgir estudos acerca dos MMORPG (Kuan-Ta Chen, 2005; Jaecheol Kim, 2005).

Em (Kuan-Ta Chen, 2005) (2005), é feita a análise de tráfego de um MMORPG. Através da cooperação da equipa de desenvolvimento do jogo, colocaram um computador a captar todo o tráfego transferido entre os servidores do jogo e os utilizadores (utilizando o software *tcpdump*). Este método permitiu obter tráfego de uma situação real. Com a informação obtida fizeram

⁴Perda neste contexto refere-se à perda de pacotes.

análise do tamanho dos pacotes, largura de banda usada e diferença temporal entre pacotes. Em (Jaecheol Kim, 2005) (2005), é feita a análise de tráfego de outro MMORPG. Como método, criaram o próprio servidor do jogo e disponibilizaram-no *online*. A partir daí e à medida que os utilizadores, de toda a parte do globo, se ligavam ao servidor, fizeram uma caracterização do tráfego tendo em conta: total de pacotes transferidos, tamanho dos pacotes, diferença temporal entre chegada/envio de pacotes, largura de banda ocupada e relação entre número de utilizadores activos e largura de banda ocupada (mais uma vez obtiveram os dados utilizando o *tcpdump*).

Os estudos apresentados dos MMORPGs, apesar de terem uma componente *online* muito forte e de se tratarem de métodos de teste reais, não revelam qualquer relação do tráfego com as acções decorridas no jogo.

2.2.2.2 *Second Life*

A partir de 2007 começam a aparecer artigos de caracterização de tráfego e até de criação de modelos sintéticos de tráfego, do *Second Life* (Stênio Fernandes, 2007; James Kinicki, 2008; Rafael Antonello, 2008). Em (Stênio Fernandes, 2007) (2007), é feita uma análise de tráfego do *Second Life*. Para efectuar esta análise, os autores visitaram (através do software cliente) diferentes locais a diferentes horas do dia e definiram dois locais de teste: um popular, outro impopular. A “popularidade” foi definida através do *traffic index* encontrado no *Second Life client*. Os testes foram efectuados utilizando dois acessos diferentes (um na universidade e outro residencial). Cada experiência foi repetida para 3 acções do *avatar* (parado, andar, voar) e foram feitos com e sem música, no sentido de ter em conta tráfego externo. Para efectuar a análise utilizaram o *tcpdump* de onde obtiveram taxa de transferência, tamanho dos pacotes, diferença temporal entre chegada de pacotes e volume de tráfego, tanto para TCP como para UDP. Em (James Kinicki, 2008) (2008), é feita uma análise de tráfego do *Second Life*. Para efectuar esta análise, os autores definiram três locais de teste com características diferentes: 1) Muitos objectos, muitos *avatar's*; 2) Muitos objectos, poucos *avatar's*; 3) Poucos objectos, poucos *avatar's*. Cada experiência consistiu em executar quatro acções diferentes com o *avatar*: *teleporting*, parado, andar, voar. Para efectuar a análise utilizaram o *wireshark* de onde obtiveram largura de banda (ascendente e descendente), tamanho dos pacotes e diferença temporal entre chegada de pacotes. Em (Rafael Antonello, 2008) (2008), é feita uma análise de tráfego do *Second Life* praticamente igual à de (Stênio Fernandes, 2007) com o acréscimo de utilizarem os dados obtidos para modelizar o tráfego.

De notar que em todos os estudos efectuados, nenhum utiliza um sistema automatico de teste. Todos os testes são efectuados passando horas em frente ao computador executando os movimentos necessários com o *avatar* e retirando os dados necessários, de softwares como *Wireshark* e *Tcpdump*. De notar também que as situações de simulação são sempre voláteis, pois utilizam sempre o servidor nativo do *Second Life*, e nunca conseguem garantir um certo número de utilizadores simultâneos nem o número de objectos na zona de teste. Esta pesquisa sugere que se crie um sistema de caracterização do tráfego do *Second Life* em que se controle todas as variáveis (servidor incluído) e que possa ser feito de forma automática, reduzindo o tempo dispendido, pelo utilizador do sistema, a recolher dados, e permitindo-o focar mais na análise dos mesmos.

Capítulo 3

Arquitectura

Neste capítulo é apresentado o estudo da arquitectura de sistema.

3.1 Caso de utilização

Uma das primeiras questões que se levantam ao tentar definir a arquitectura do sistema é "Que passos segue um utilizador quando pretende usar um sistema deste tipo?". Esta questão levou à definição da estrutura de um caso de utilização genérico.

Em primeiro lugar há a definição da experiência - o utilizador (operador do sistema) planeia uma experiência e as condições em que deve ser feita. Seguidamente, através das ferramentas disponíveis, procede à recolha de dados fornecidos pela experiência. Após a recolha, faz uma análise dos dados recolhidos seja com a criação de histogramas, tabelas correlacionais, gráficos representativos dos dados recolhidos, comparações de dados ou outros métodos que sejam relevantes para o utilizador. Por fim e após análise dos dados seguem-se as conclusões – como em qualquer experiência o objectivo é tirar conclusões acerca dela, informações relevantes ao estudo que esteja a ser feito. Este processo está graficamente representado na figura 3.1.

Esta tese diz respeito ao processo de recolha de dados (área definida pelo losango da figura 3.1), ou seja, à aquisição de dados de tráfego que permitam efectuar a caracterização do mesmo. Pretende-se criar um sistema que permita ao utilizador definir a sua experiência nele, e que, de uma maneira autónoma, o sistema faça a recolha de dados, facilitando a execução de múltiplas experiências e sua repetição.

3.2 Definição da arquitectura

No sentido de atingir o objectivo desta dissertação, é necessária a elaboração de uma arquitectura que permita, ao utilizador, realizar a avaliação e caracterização de tráfego de rede de jogos,

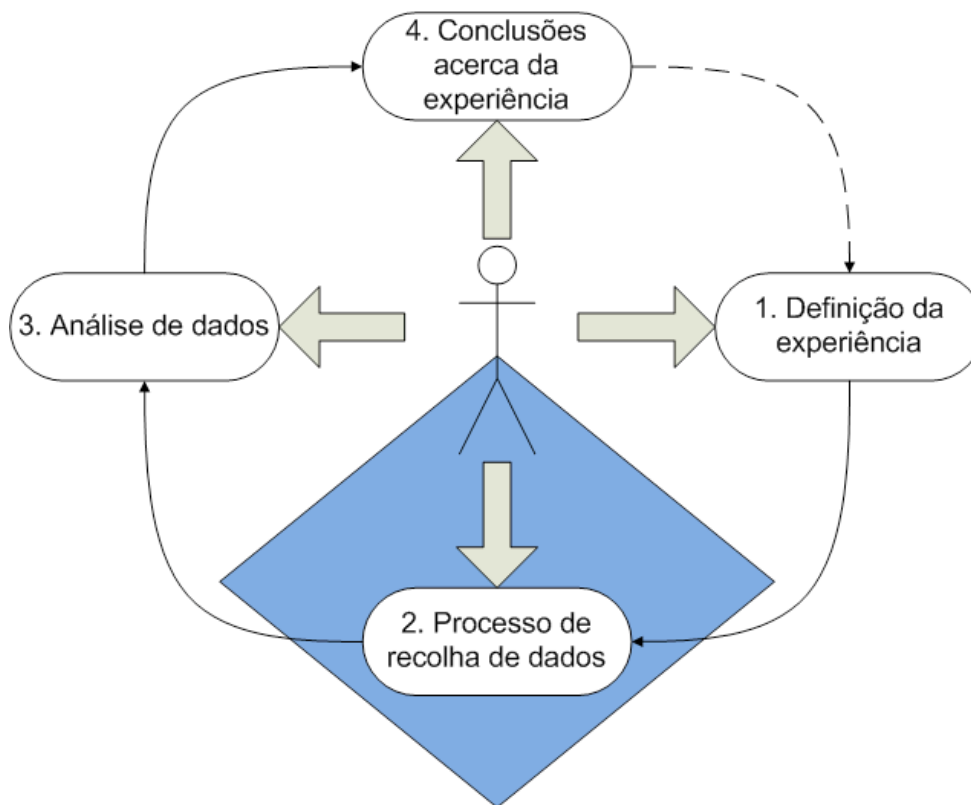


Figura 3.1: Caso de utilização.

nomeadamente do *Second Life*. Para tal é necessário definir os elementos que a constituem e depois definir como interligar esses mesmos elementos.

3.2.1 Elementos

Os jogos online, do género do *Second Life*, são constituídos por múltiplos clientes e servidores. No *Second Life* os servidores não desempenham todas as mesmas funções já que há *Login servers*, *User servers*, *Space servers*, *Data servers*, *Simulator servers*, entre outros (Rafael Antonello, 2008). Para criar um sistema que permita a análise do tráfego gerado pela simulação, optou-se pela estrutura mais simples: um cliente e um servidor (que desempenhe todas as funções necessárias). Para analisar o tráfego é necessário um terceiro elemento que forneça informações detalhadas sobre o tráfego transmitido entre os dois elementos a que chamaremos, observador de tráfego. Assim sendo os elementos constituintes da arquitectura são:

- Cliente.
- Servidor.
- Observador de tráfego.

3.2.2 Topologias lógicas de conexão

Para além da definição dos elementos, é necessário saber como interligá-los. Para satisfazer esta necessidade foram pensadas múltiplas topologias lógicas de conexão:

1. **Cliente(modificado) <-> Servidor** — Com esta topologia seria necessário modificar o código do software cliente para ele próprio desempenhar o papel de observador de tráfego. Esta topologia está representada graficamente na figura 3.2.

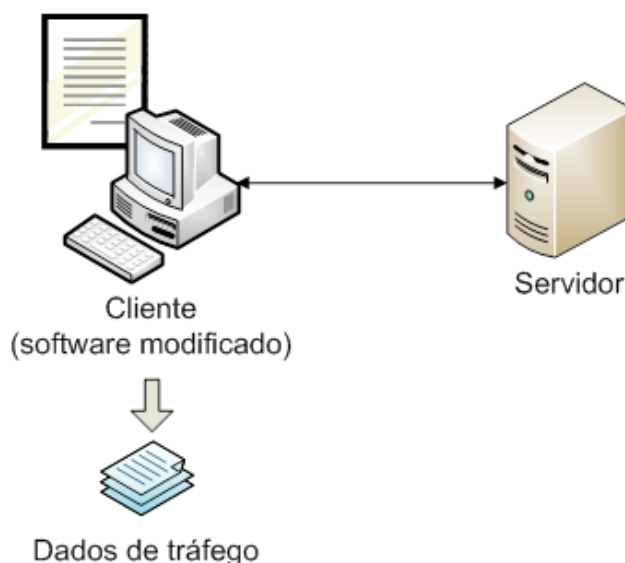


Figura 3.2: Topologia: Cliente(modificado) <-> Servidor.

2. **Cliente <-> Servidor(modificado)** — Nesta topologia seria necessário modificar o código do software servidor para ele próprio desempenhar o papel de observador de tráfego. Esta topologia está representada graficamente na figura 3.3.
3. **Cliente <-> Observador de tráfego <-> Servidor** — Neste caso não são efectuadas quaisquer alterações aos softwares cliente e servidor e é inserido um observador de tráfego entre eles. Aqui o observador de tráfego funciona como *Proxy* (o cliente comunica com o servidor através do observador de tráfego). O observador de tráfego recebe os dados de um dos elementos (cliente ou servidor), guarda uma cópia dos mesmos para futura análise, e transmite-os ao elemento de destino (servidor ou cliente). Esta topologia está representada graficamente na figura 3.4.
4. **Cliente <-(Observador de tráfego)-> Servidor** — Neste caso não são efectuadas quaisquer alterações aos softwares cliente e servidor. Ao contrário do ponto anterior 3, o cliente comunica com o servidor directamente. O observador de tráfego é apenas ligado a um ponto em que o tráfego passa, e recolhe cópias do mesmo para futura análise. Esta topologia está representada graficamente na figura 3.5.

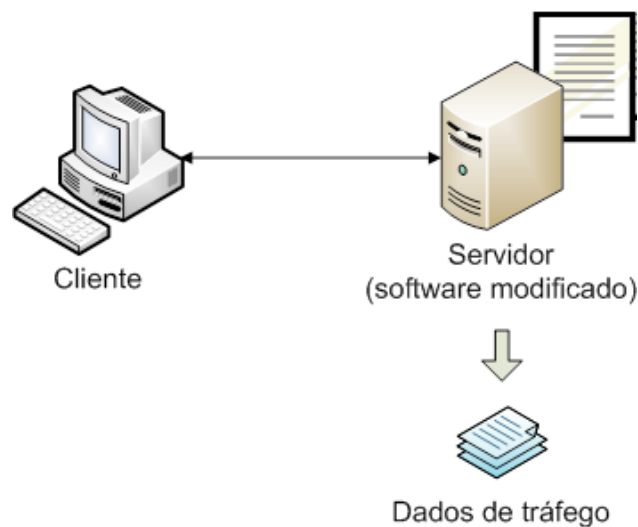


Figura 3.3: Topologia: Cliente \leftrightarrow Servidor(modificado).

Nas topologias 1 e 2 é necessário modificar o código dos softwares, o que é possível apenas em aplicações *Open Source*. Mesmo quando é possível modificar o código fonte dos softwares, o código pode ser tão extenso e/ou complexo que poderia ser necessário o trabalho conjunto com as equipas de desenvolvimento originais. O maior entrave nestas duas topologias é ainda o facto de não permitirem a utilização do sistema com os softwares como eles são originalmente, correndo o risco de alterar os padrões de tráfego criados por eles originalmente.

Na topologia 3 é necessário interromper a conexão directa entre os softwares para inserir o observador de tráfego (*Proxy*), o que num teste alargado (vários cliente e servidores) acrescenta um enorme tempo de reconfiguração das conexões desses elementos para que comuniquem através do *Proxy*.

Por fim a topologia 4 é a mais indicada já que numa situação normal do sistema, se se pretender começar a analisar tráfego é apenas necessário ligar o observador de tráfego a um ponto de travessia do mesmo.

3.2.3 Automatização do processo

Quando se tem um sistema de análise em mente, a automatização do mesmo é praticamente imprescindível. Através de um sistema de automatização pretende-se reduzir ao mínimo a intervenção humana na configuração, execução, e obtenção de resultados e permitir a repetição do processo tantas vezes quantas as necessárias, sem acréscimo de intervenção do utilizador.

Neste caso para um simples teste manual seria necessário executar os seguintes passos tantas vezes quantas as repetições do teste que se pretende:

1. Configurar o cliente
2. Configurar o servidor



Figura 3.4: Topologia: Cliente <-> Observador de tráfego <-> Servidor.

3. Configurar o observador de tráfego
4. Executar o servidor
5. Executar o cliente
6. Executar o observador de tráfego
7. Utilizar o cliente de forma a gerar o padrão necessário (exemplo: no caso do *Second Life* poder-se-ia testar andar para a frente durante 10 s)
8. Desligar o cliente
9. Desligar o servidor
10. Desligar o observador de tráfego

Executar este teste manual múltiplas vezes seria demasiado moroso o que dificultaria, se necessário, a obtenção de um grande número de repetições. Adicionalmente controlar o software cliente manualmente tornaria impossível repetir com exactidão o mesmo padrão referido no ponto 7 da lista de passos de um teste manual.

Tendo isto em mente torna-se necessária a elaboração de um sistema de automatização que lide com a topologia escolhida em 3.2.2.

3.3 Conclusão

Após análise das topologias (3.2.2) foi escolhida a topologia 4: **Cliente <-(Observador de tráfego)-> Servidor**. Foi ainda definida a necessidade de criar um sistema de automatização (3.2.3) para a arquitectura.

É possível ver na figura 3.5 o esquema da arquitectura definida.

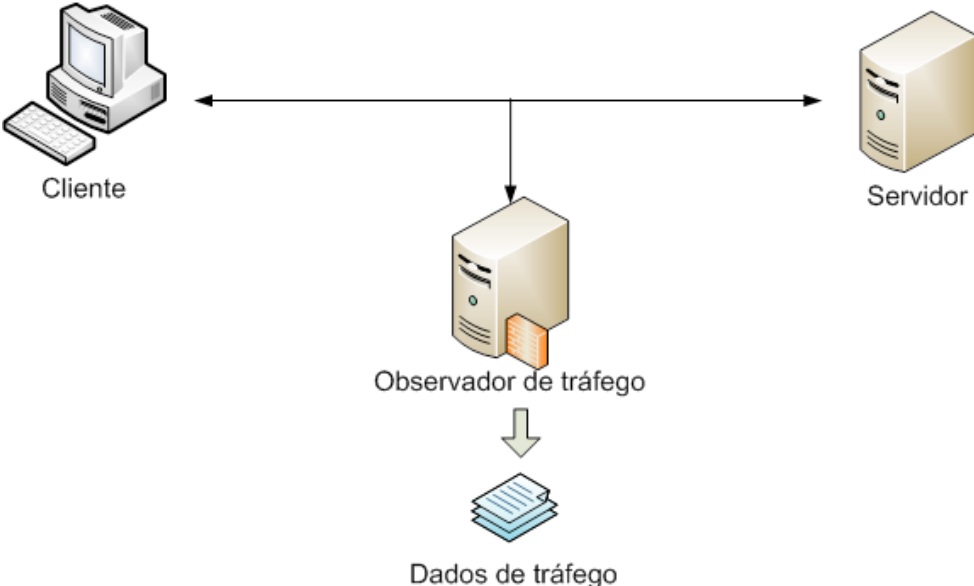


Figura 3.5: Topologia: Cliente <-(Observador de tráfico)-> Servidor.

Capítulo 4

Implementação

Neste capítulo é descrita a implementação passo-a-passo da arquitectura definida no capítulo 3.

4.1 Primeira fase

Antes de começar a construir o sistema foi necessário fazer uma primeira análise de tráfego para perceber o comportamento do sistema e dos seus elementos, e ainda definir os requisitos para a automatização do processo.

4.1.1 Elementos

Para realizar esta primeira análise foram utilizados os seguintes elementos:

- **Cliente** — Como software cliente foi utilizado o software cliente nativo do *Second Life* (Life, 2009).
- **Servidor** — Como software servidor foi utilizado o *OpenSimulator v0.6.5 Alpha with post-release fixes* (net, 2009). *OpenSimulator* é um servidor de aplicações 3D e vem de origem pronto a ser usado para simular ambientes virtuais similares ao *Second Life* (incluindo compatibilidade com software cliente). A utilização de um servidor próprio e não de um servidor disponível na internet, serve o propósito de controlar todas as variáveis de teste. Com um servidor controlado é possível definir com certeza as condições experimentais tais como número de *avatar's* na zona, número de objectos na zona e configuração da zona (tipo de terreno e sua forma). Se fosse utilizado um servidor disponível na internet as variáveis de teste oscilariam no tempo e conforme a popularidade da zona em que se fizesse *login*.
- **Observador de tráfego** — Como software observador de tráfego foi utilizado o *Wireshark* (Foundation, 2009c). *Wireshark* é um analisador de protocolos de rede e como tal permite recolher informação de todo o tráfego em determinada ligação e inspeccionar o conteúdo de todas as *frames* que a atravessam.

4.1.2 Topologia lógica de conexão

O cliente *Second Life* foi instalado num computador com o processador Intel Core 2 Duo T7200 (2.00 GHz), 2.0 GB de RAM e com o sistema operativo *Microsoft Windows XP Professional*. No sentido de simplificar a configuração de rede e de apenas necessitar de um computador para efectuar toda a simulação, foi usada uma máquina virtual para a instalação do software servidor. Esta máquina virtual foi configurada para utilizar 512 MB de RAM e usar apenas um dos *Core's* do processador da máquina em que estava instalada (*host*). Nessa máquina virtual foi instalado o sistema operativo *Debian 5.0*. Como a máquina virtual disponibiliza interfaces virtuais de rede em que o cliente e servidor comunicam, ficou simplificado o acesso ao tráfego por parte do *Wireshark*. É possível ver na figura 4.1 uma representação da topologia lógica de conexão:

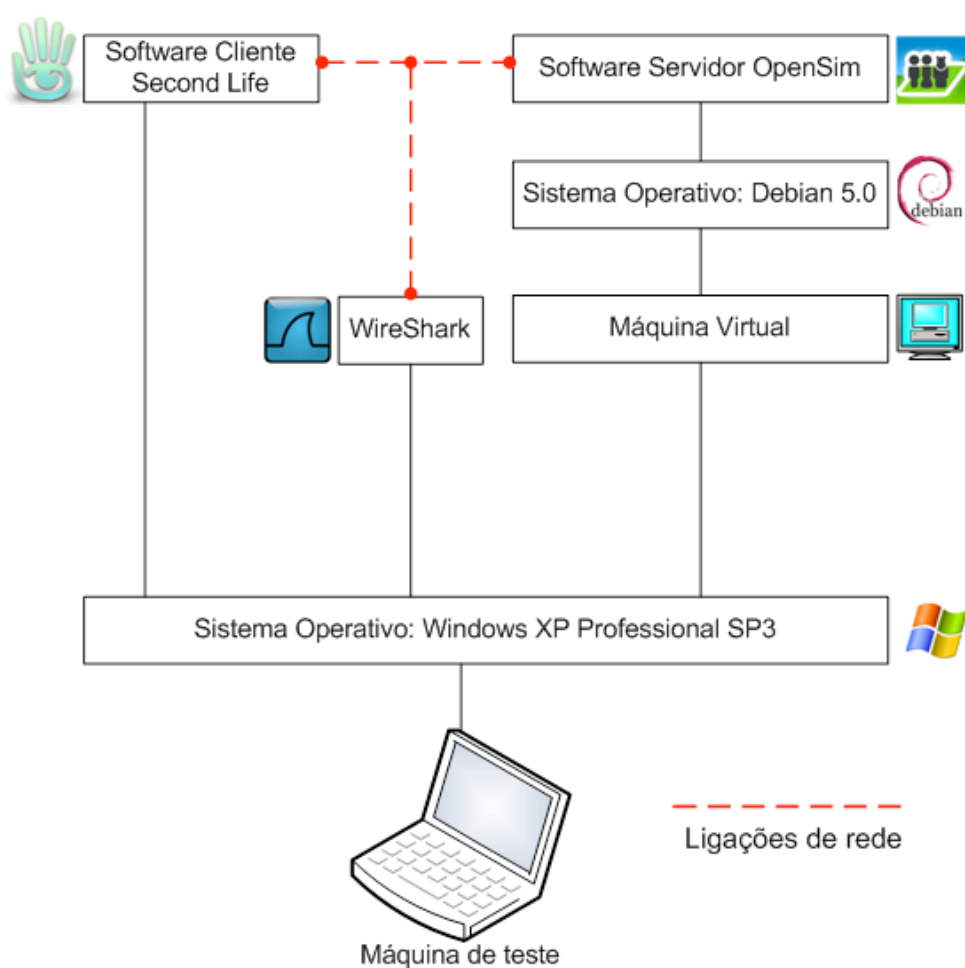


Figura 4.1: Topologia lógica de conexão.

4.1.3 Configuração e execução

Para realizar esta primeira análise de tráfego seguiram-se os seguintes passos:

1. Instalar e configurar o cliente *Second Life* - A instalação é simples mas para o software estabelecer a ligação ao servidor foi necessário alterar os argumentos de execução da aplicação para:

```
--set InstallLanguage en -loginuri http://192.168.43.131:9000/  
-loginpage http://192.168.43.131:9000/?method=login
```

O endereço IP “192.168.43.131” é o endereço do servidor, e a porta 9000 a utilizada pelo *Second Life* para comunicar.

2. Instalar e configurar o servidor (máquina virtual + *OpenSimulator*) - A instalação faz-se recorrendo ao comando “svn co http://opensimulator.org/svn/opensim/branches/0.6.5-post-fixes opensim”. A configuração do *OpenSimulator* é feita através do ficheiro “OpenSim.ini” na directoria “bin” do software. Originalmente este ficheiro não existe, mas basta alterar o nome do ficheiro “OpenSim.ini.example” para “OpenSim.ini” e o software está pronto a ser executado. Para correr o *OpenSimulator* basta correr o comando “mono OpenSim.exe” dentro da directoria “bin” do software e seguir as instruções. É necessário ainda criar um utilizador com o comando “create user <primeiro nome>” e seguir as instruções. Todas estas informações de instalação e configuração estão disponíveis no site oficial do *OpenSimulator*.
3. Instalar e configurar o *Wireshark*.
4. Executar todos os softwares: (cliente, servidor e *Wireshark*).
5. Iniciar recolha de *frames* com o *Wireshark*.
6. Fazer login com o cliente.
7. Executar alguma acção com o avatar (Neste caso aguardou-se 60 minutos com o avatar parado). Pode ver-se um *screenshot* do ambiente de trabalho do computador no momento deste passo (À esquerda o *Wireshark*, no canto superior direito o cliente de *Second Life*, e por fim no canto inferior direito, a consola do *OpenSimulator* a correr na máquina virtual).
8. Desligar o cliente.
9. Desligar o servidor - Para desligar o servidor basta executar o comando "quit".
10. Parar a recolha de *frames* com o *Wireshark*.
11. Exportar os dados recolhidos com o *Wireshark* para um ficheiro CSV legível.
12. Desligar o *Wireshark*.

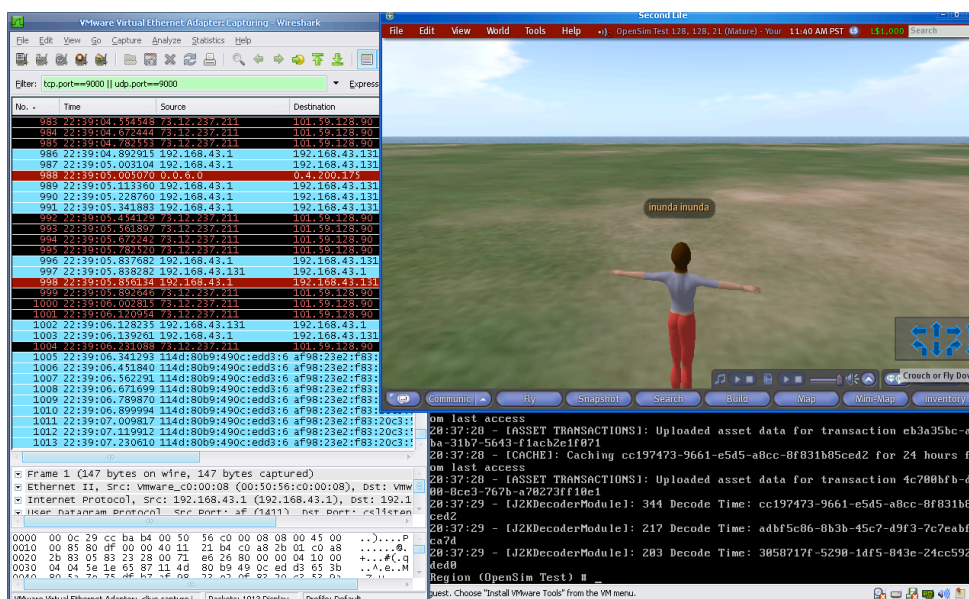


Figura 4.2: Screenshot do ambiente de trabalho na altura do teste.

4.1.4 Resultados e conclusões

Após execução da primeira análise e analisando os passos descritos em 4.1.3 é possível verificar que para repetição do mesmo teste, apenas é necessário executar os passos a partir do ponto 4.

É de notar que para um número de repetições alargado o trabalho feito manualmente seria moroso e não seria possível obter um grande número de repetições em tempo útil. Para além desta desvantagem há ainda a presença obrigatória do utilizador durante todo o período de teste para controlar o *avatar* e todos os softwares utilizados.

No ponto 7 é onde tomamos controlo do *avatar* do *Second Life* e executamos uma sequência de passos (*script*¹). Estas sequências podem ser tão extensas quanto se desejar, e repeti-las, uma vez que seja, exactamente do mesmo modo, é praticamente impossível através do cliente nativo do *Second Life* (Um simples *script* em que se faz *login* e se anda para a frente durante 5 segundos, é difícil de reproduzir porque controlando manualmente vamos ter variação no tempo em que estamos a andar para a frente e no tempo que demoramos a partir do *login* até começar a andar para a frente).

Toda a análise aponta para a necessidade de um sistema de automatização que não cometa os erros introduzidos pela componente humana na realização manual dos testes.

Ainda através desta primeira análise e observando o *log*², foi possível ainda verificar que mesmo com o *avatar* parado, continua a ser gerado tráfego continuamente o que torna obrigatória a execução contínua do *Wireshark* durante todo o processo de teste.

¹*script* é uma sequência de comandos a seguir. No caso do *Second Life* será a sequência de comandos a executar com o *avatar* no mundo virtual.

²*log* significa o ficheiro de registo. Neste caso o ficheiro de registo do tráfego.

4.2 Automatização

Atendendo à necessidade de criar um sistema de automatização, verificada na primeira fase em 4.1, foi projectada e desenvolvida uma ferramenta a que se deu o nome de *Second Life Network Test (SLNT)*.

Esta ferramenta tem como objectivo automatizar todo o processo de teste independentemente do *script* ou do número de repetições do teste sem que seja necessária a presença humana durante a execução e que retorne os resultados já prontos a serem lidos por um qualquer outro programa que analise ou represente graficamente.

Com o *SLNT* deverá ser possível definir um *script* e o número de repetições desejadas. Após execução do teste deve-se obter os *logs* de cada repetição do teste. Pode ver-se a representação do funcionamento do *SLNT* na figura 4.3.

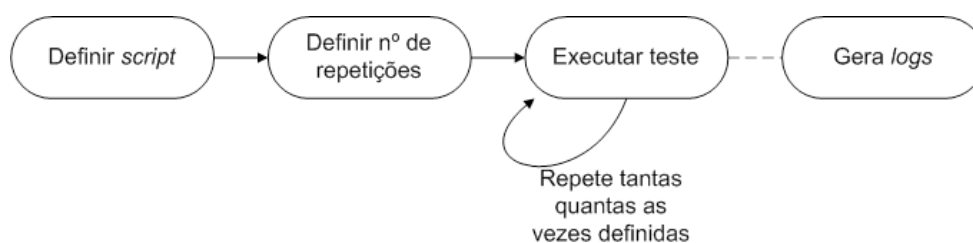


Figura 4.3: Funcionamento do *SLNT*.

4.2.1 Elementos

Na primeira fase foram definidos certos elementos para o cliente, servidor e observador de tráfego. No entanto a fim de automatizar o processo alguns desses elementos tiveram que ser alterados.

4.2.1.1 Cliente

Para automatizar os movimentos do avatar foi necessário deixar o cliente nativo do *Second Life* de parte, e passar a utilizar uma biblioteca *Open Source* de nome *libopenmetaverse* (anteriormente chamada de *libsecondlife*) (Foundation, 2009b,a). Esta biblioteca, baseada na tecnologia *.NET*, permite o acesso e criação de mundos virtuais 3D, e mantém compatibilidade com o protocolo do *Second Life*, podendo ser utilizada para criar clientes e *automatons* no *Second Life*, *OpenSimulator* e noutros mundos virtuais que usem o protocolo *Second Life*.

Como primeira tentativa do uso desta biblioteca tentou-se programar um software cliente de raiz. Esta abordagem rapidamente foi descartada pois não só seria demasiado morosa, como a documentação disponível ainda não é consistente e estável o suficiente para permitir um rápido e fácil desenvolvimento de um software baseado nesta biblioteca.

Dentro do mesmo projecto é disponibilizado um software (desenvolvido pela mesma equipa) que permite o controlo de um avatar através da linha de comandos – *Testclient*. O *Testclient*

mostrou-se também demasiado instável (na versão 0.6.3), mas após compilar e testar algumas versões mais antigas, foi possível obter uma versão estável o suficiente para ser utilizada (versão 0.6.1.1)³. Ainda assim é um projecto em estado *alpha* e portanto apresenta alguns obstáculos tanto na sua utilização como na eventual modificação do código fonte.

O *Testclient* coloca ao dispor do utilizador 100 comandos possíveis⁴. Entre todos os comandos disponíveis, foram seleccionados os que dizem respeito ao movimento do *avatar* e alguns necessários à execução dos *scripts*, listados nas tabelas 4.1 e 4.2 com a lista de comandos, seus argumentos e suas acções.

4.2.1.2 Servidor

O software servidor (*OpenSimulator*) foi mantido tal como o esquema de o ter instalado numa máquina virtual afim de utilizar apenas um computador para efectuar todos os testes sem interferir com a rede física a que este esteja ligado (uso das interfaces virtuais disponibilizadas pela máquina virtual).

4.2.1.3 Observador de tráfego

Na primeira análise foi utilizado o *Wireshark*. No entanto o *Wireshark* não permite automatizar a recolha e exportação dos dados recolhidos. Assim foi utilizada a ferramenta *Tshark* que é parte integrante do software *Wireshark* (é instalado na mesma directoria do *Wireshark*). Por ser uma ferramenta de linha de comandos, permite a sua configuração através da definição de argumentos aquando da sua execução e como o retorno é configurável e direccionado à parte visual da consola em que é executado, permite a recolha dos dados para um ficheiro já devidamente formatado (CSV).

4.2.2 Arquitectura

4.2.2.1 Requisitos

O *SLNT* foi pensado com um funcionamento muito simples: aceita dados de entrada, processa-os e retorna os resultados (dados de saída).

- **Dados de entrada:**

- *script* a executar.
- Número de repetições do teste.
- Informações de login [nome de utilizador(primeiro e ultimo), palavra-passe].
- URI (endereço do servidor, por exemplo: <http://127.0.0.1:9000>), interface de rede por onde comunicam cliente e servidor.

³De notar que as versões disponibilizadas para download no site, e as disponíveis no *Subversion* SVN, são diferentes mesmo quando catalogadas com o mesmo índice de versão (as do SVN são mais fiáveis).

⁴Estes comandos podem ser consultados apenas no próprio software. Após execução do mesmo e *login* no mundo virtual, basta escrever o comando “help” (nem todos os comandos listados estão implementados).

- Directório onde guardar os *logs* gerados.
- Directório e ficheiro do *Testclient*.
- Directório e ficheiro do *Tshark*.

- **Dados de saída:**

- O *SLNT* gerará *logs* do teste efectuado e deverá guardá-los no directório definido. Os *logs* terão como conteúdo:
 - * *frame.number* — Número da *frame* captada (todas as *frames* captadas são numeradas a partir de 1. Claro que nestes *logs* só aparecerão as *frames* respeitantes ao tráfego do *Second Life* (TCP ou UDP, porta 9000)).
 - * *frame.time* — Altura em que foi captada a *frame* (tempo em relação ao início da captura).
 - * *frame.time_delta* — Diferença temporal entre a *frame* actual e a anterior.
 - * *frame.protocols* — Protocolos usados (como estão filtrados só vai aparecer TCP e UDP)
 - * *frame.len* — Tamanho da *frame*

4.2.3 Implementação

O *SLNT* foi programado em C# tal como os projectos relacionados com o *Second Life* (*libOpenMetaverse*, *OpenSimulator*) e permite assim a possibilidade de uma futura junção dos projectos. Na figura 4.4 é possível ver o *frontend* do *SLNT*.

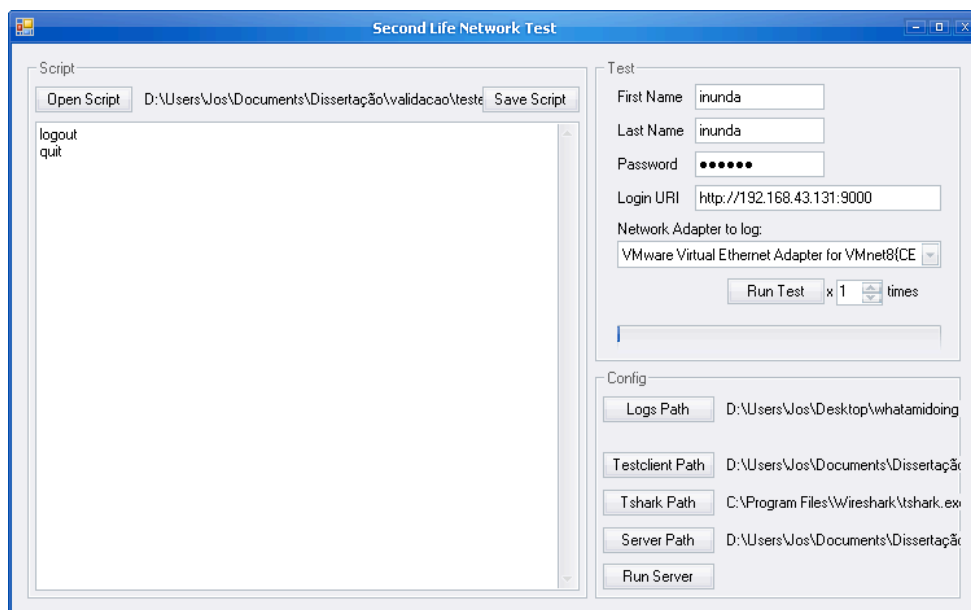


Figura 4.4: *Frontend* do *SLNT*.

A janela está dividida em três áreas:

1. *Script*
2. *Test*
3. *Config*

A área *Script* é onde se podem abrir, salvar e editar os *scripts*. Em *Test* definem-se os parâmetros de teste: utilizador (primeiro e último nome), palavra-passe do utilizador, URI do servidor, adaptador do qual queremos retirar informação de tráfego e número de repetições. Clicando em *Run Test*, o programa começa a execução do *script* aberto (quando alterado deve ser salvo antes da execução). Na área de *Config* configuram-se: o directório onde os *logs* vão ser guardados, o caminho para o executável do *Testclient*, o caminho para o executável do *Tshark* e o caminho para o servidor de *OpenSimulator* local, se se pretender efectuar testes com o software na mesma máquina. Está ainda disponível o botão *Run Server* que executa o servidor local. Todas as configurações da área *Config* são guardadas num ficheiro XML (*config.xml*) no mesmo directório do executável *SLNT*.

4.3 Resumo

Através da criação da ferramenta de automatização *SLNT* o sistema de teste resume-se à elaboração de *scripts* e execução dos mesmos com o *SLNT*. A partir daí é possível o utilizador analisar os dados gerados (*logs*) e tirar as conclusões necessárias. Através deste processo simplificado a quantidade de testes e repetição dos mesmos é alargada e em tempo útil conseguem-se múltiplos testes.

Tabela 4.1: Tabela de comandos de movimentos do *avatar*.

Comando	Argumentos	Ação
left	[seconds]	Envia o comando de movimento “esquerda” para um pacote ou para um certo número de segundos.
siton	UUID	Tenta sentar o <i>avatar</i> em determinado <i>prim</i> com um UUID específico.
follow	[FirstName LastName]/off	Segue outro <i>avatar</i> .
fly	[start/stop]	Começa ou pára de voar.
back	[seconds]	Envia o comando de movimento “trás” para um pacote ou para um certo número de segundos.
turnto	x y z	Roda o <i>avatar</i> para um ponto específico.
stand	—	Levanta-se.
sit	—	Tenta sentar-se no <i>Prim</i> mais próximo.
location	—	Mostra a localização actual do <i>avatar</i> .
goto_landmark	UUID	“Teletransporta-se” para determinada <i>landmark</i> .
right	[seconds]	Envia o comando de movimento “direita” para um pacote ou para um certo número de segundos.
forward	[seconds]	Envia o comando de movimento “frente” para um pacote ou para um certo número de segundos.
jump	—	Salta ou voa para cima.
crouch	[start/stop]	Começa ou para de se agachar.
sethome	—	Regista a localização actual como <i>home</i> .
goto	sim/x/y/z	“Teletransporta-se” para localização determinada pelos parâmetros.
FlyTo	x y z [seconds]	Voa em direcção à localização especificada. Opcionalmente define-se o tempo máximo que demora a chegar ao destino.
gohome	—	“Teletransporta-se” para localização registada como <i>home</i> .
moveto	x y z	Move o <i>avatar</i> para a posição global especificada usando o <i>autopilot</i> do simulador.

Tabela 4.2: Tabela de comandos relacionados com *Testclient*.

Comando	Argumentos	Ação
logout	—	Desconecta-se do servidor.
quit	—	Terminar o programa.
sleep	[seconds]	Dorme o número de segundos especificado.
help	[command]	Lista os comandos disponíveis. Com o [command] definido, lista apenas informação acerca do mesmo.

Capítulo 5

Validação

Neste capítulo procede-se à validação de todo o sistema através de testes e observação de seus resultados.

5.1 Validação do funcionamento do *SLNT*

A primeira questão que se põe aquando da validação, é se o sistema funciona correctamente e de acordo com a implementação (4.2).

5.1.1 Método

Para validar o funcionamento devem ser verificadas as seguintes questões:

1. O avatar utilizado pelo *SLNT* faz exactamente o que está descrito no *script* seleccionado?
2. O processo é repetido exactamente tantas quantas as vezes definidas no *SLNT*?
3. Os *logs* gerados pelo *SLNT* correspondem exactamente aos capturados manualmente com o *Wireshark*?

Para verificar o primeiro item foi colocado um espectador no mundo virtual, isto é, utilizou-se o cliente nativo do *Second Life* para entrar no mundo virtual, com outro utilizador, e assim visualizar os movimentos efectuados pelo avatar do *SLNT*.

Para testar o segundo item foi definido no programa um número de repetições igual a 3.

Por fim para verificar o terceiro item, foi executada uma instância do *Wireshark* ao mesmo tempo que os testes foram efectuados. No fim dos testes foram comparados os *logs* gerados pelo *Wireshark* e pelo *SLNT*.

5.1.1.1 *Scripts* utilizados

Foram definidos, para este teste, dois *scripts*, um de curta duração e um de longa duração.

- *Script* de curta duração (10 segundos):

```
forward 10
logout
quit
```

- *Script* de longa duração (50 segundos):

```
forward 10
left 10
sleep 5
back 10
right 10
sleep 5
logout
quit
```

É possível visualizar o aspecto do *SLNT* configurado para a execução dos dois *scripts* na figura 5.1.

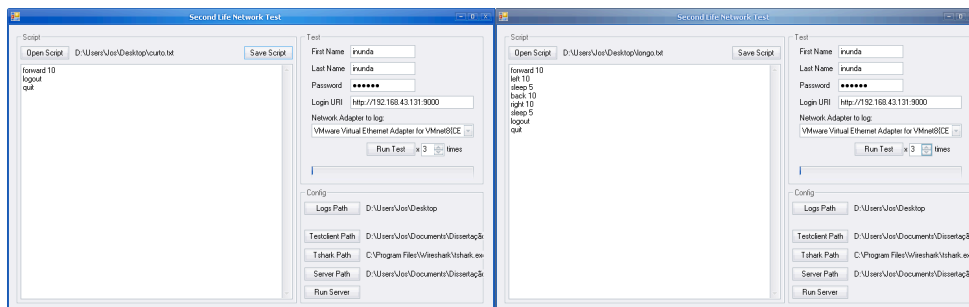


Figura 5.1: *Screenshots* do *SLNT* configurado para execução dos dois *scripts*.

5.1.2 Condições de teste

Todos os testes foram efectuados numa zona de 256 x 256 metros quadrados de área plana e deserta (sem quaisquer objectos adicionais) tendo como único elemento adicional, o avatar de verificação mencionado no método. Para criar esta zona de terra plana foi executado o seguinte comando no *OpenSimulator*: "Terrain fill 21". O número 21 refere-se à altitude do terreno e foi escolhido este valor porque corresponde ao nível do mar no simulador. É possível ver o resultado gráfico do terreno na figura 5.2.



Figura 5.2: *Screenshot* do terreno da zona virtual.

5.1.3 Conclusão

Após execução dos testes descritos, concluiu-se que o sistema funciona sem problemas. Foi possível verificar visualmente que o avatar executou as funções descritas pelos *scripts* tantas quantas as vezes definidas no *SLNT* e os *logs* obtidos foram os mesmo obtidos com a execução manual do *Wireshark*.

5.2 Potencialidade de utilização

Até este ponto temos todo um sistema arquitetado, implementado e testado. No entanto até que ponto será este sistema útil? Será possível obter informação importante com ele? Será possível analisar a informação fornecida pelo sistema de forma a retirar conclusões acerca do tráfego do *Second Life*? No intuito de responder a estas questões, foi analisada a potencialidade de utilização do *Sistema de caracterização da rede em jogos 3D online multi-jogador*.

Foram estipulados dois casos de uso com o intuito de testar as funcionalidades basicas de movimentos do avatar (andar, voar, frente, trás, direita, esquerda):

1. Testes de longa duração: Conjunto de *scripts* de longa duração com apenas uma repetição.
2. Testes repetitivos: Conjunto de *scripts* de pequena duração mas com largo número de repetições.

5.2.1 Casos de Uso

5.2.1.1 Testes de longa duração

Com este grupo de testes, pretende-se verificar se é possível distinguir os diversos tipos de acções no mundo virtual, através da análise de tráfego. Foi criado um conjunto de 4 *scripts* de

longa duração (10 minutos) a partir dos quais se tentou identificar o padrão de tráfego, em modo contínuo, de cada tipo de evento.

- *Script “long-inwaitout”*: Neste teste faz-se *login*, espera-se 10 minutos, e faz-se *logout*.

```
sleep 600
logout
quit
```

- *Script “long-flywait”*: Neste teste faz-se *login*, coloca-se o *avatar* a voar, espera-se 10 minutos, e faz-se *logout*.

```
fly start
sleep 600
logout
quit
```

- *Script “long-walk”*: Neste teste faz-se *login*, coloca-se o *avatar* a andar durante 10 minutos, e faz-se *logout*. Para o *avatar* não ficar “preso” no fim da zona, fez-se com que o *avatar* andasse às voltas. O comando “gohome” serve para iniciar o percurso a partir de um ponto específico em todos os testes que requerem deslocação do *avatar*.

```
gohome
forward 10
left 10
back 10
right 10
(...) -> repete-se as 4 últimas instruções 14 vezes.
          (4*10 + 14*4*10 = 600 segundos)
logout
quit
```

- *Script “long-flyto”*: Neste teste faz-se *login*, coloca-se o *avatar* a voar durante 10 minutos, e faz-se *logout*. Para o *avatar* não ficar “preso” no fim da zona, fez-se com que o *avatar* andasse às voltas.

```
gohome
fly start
forward 5
left 5
back 5
right 5
```

```
(...) -> repete-se as 4 últimas instruções 29 vezes.  
        (4*5 + 29*4*5 = 600 segundos)  
fly stop  
logout  
quit
```

5.2.1.2 Testes repetitivos

Neste conjunto de testes, pretende-se verificar: 1) se é possível fazer uma caracterização de tráfego relacionado com as ações no mundo virtual, utilizando testes de curta duração; 2) se através de repetições sucessivas dos mesmos *scripts*, se obtêm resultados semelhantes.

Foi criado um conjunto de 8 *scripts* de curta duração (10 segundos ou menos). Cada *script* foi repetido, automaticamente, 10 vezes.

- *Script* “inout”: Neste teste faz-se *login* e *logout* seguidamente.

```
logout  
quit
```

- *Script* “fly”: Neste teste faz-se *login*, coloca-se o *avatar* a voar, pára-se o vôo, e faz-se *logout*.

```
fly start  
fly stop  
logout  
quit
```

- *Script* “inwaitout”: Neste teste faz-se *login*, aguarda-se 60 segundos, e faz-se *logout*.

```
sleep 60  
logout  
quit
```

- *Script* “forward”: Neste teste faz-se *login*, anda-se para a frente durante 10 segundos, e faz-se *logout*.

```
gohome  
forward 10  
logout  
quit
```

- *Script* “flyto”: Neste teste faz-se *login*, começa-se a voar, movimenta-se o *avatar* para a frente durante 10 segundos, pára-se o vôo, e faz-se *logout*.

```
gohome
fly start
forward 10
fly stop
logout
quit
```

- *Script “back”*: Neste teste faz-se *login*, anda-se para trás durante 10 segundos, e faz-se *logout*.

```
gohome
back 10
logout
quit
```

- *Script “left”*: Neste teste faz-se *login*, anda-se para a esquerda durante 10 segundos, e faz-se *logout*.

```
gohome
left 10
logout
quit
```

- *Script “right”*: Neste teste faz-se *login*, anda-se para a direita durante 10 segundos, e faz-se *logout*.

```
gohome
right 10
logout
quit
```

5.2.2 Processamento de dados

5.2.2.1 Testes de longa duração

Cada *script* foi executado apenas uma vez, gerando 2 *logs* por execução (um UDP e um TCP). Dado que as informações extraídas foram o tamanho das *frames* e o intervalo temporal entre chegada de *frames*, obteve-se um conjunto total de 16 histogramas. Como o conjunto de histogramas a criar era pequeno, fez-se o processamento dos dados utilizando o Microsoft Excel 2007 (programa de folha de cálculo) e após tratamento dos dados obtiveram-se os seguintes histogramas:

- Tamanho das *frames* TCP — representados nas figuras 5.3, 5.4, 5.5 e 5.6.

- Tamanho das *frames* UDP — representados nas figuras 5.7, 5.8, 5.9 e 5.10.
- Intervalo de tempo entre chegada de *frames* TCP — representados na figura 5.13.
- Intervalo de tempo entre chegada de *frames* UDP — representados nas figuras 5.14, 5.15, 5.16 e 5.17.

De notar que, nos histogramas do tamanho de pacotes, apenas estão representados os índices que, em pelo menos um dos quatro histogramas de cada tipo (TCP e UDP), contêm uma ou mais ocorrências. Os índices que não contêm ocorrências em nenhum dos quatro histogramas, foram ocultados para melhor visualização da forma dos histogramas. Dado que os valores dos histogramas obtidos são bastante díspares, foi utilizada a escala logarítmica para a sua representação.

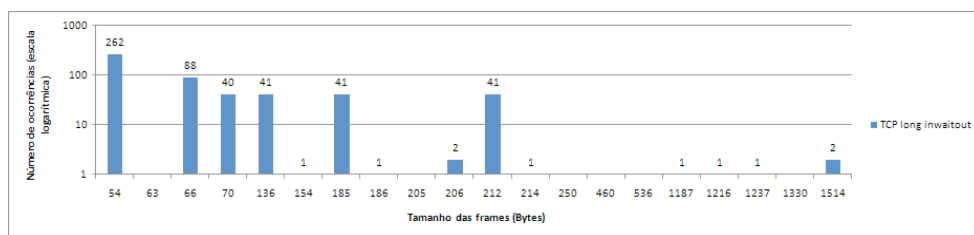


Figura 5.3: Histograma do tamanho das frames (TCP) do teste “long-inwaitout”.

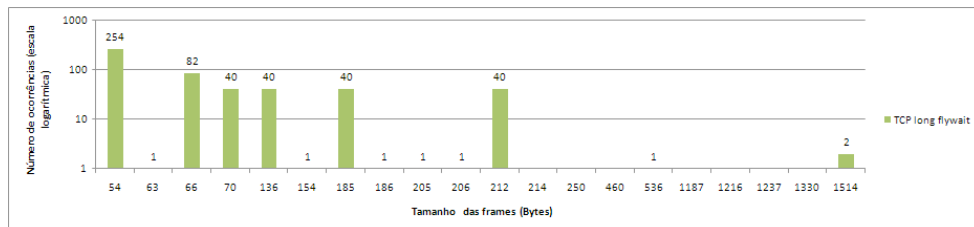


Figura 5.4: Histograma do tamanho das frames (TCP) do teste “long-flywait”.

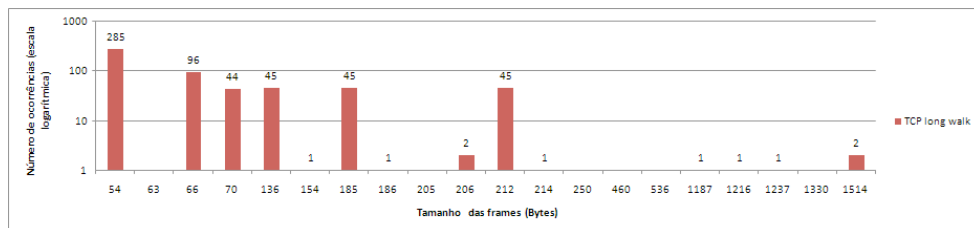


Figura 5.5: Histograma do tamanho das frames (TCP) do teste “long-walk”.

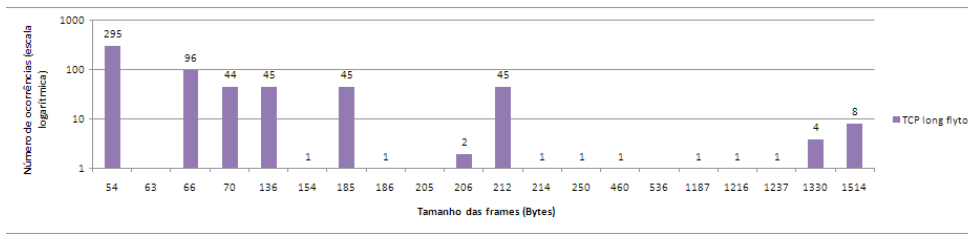


Figura 5.6: Histograma do tamanho das frames (TCP) do teste “long-flyto”.

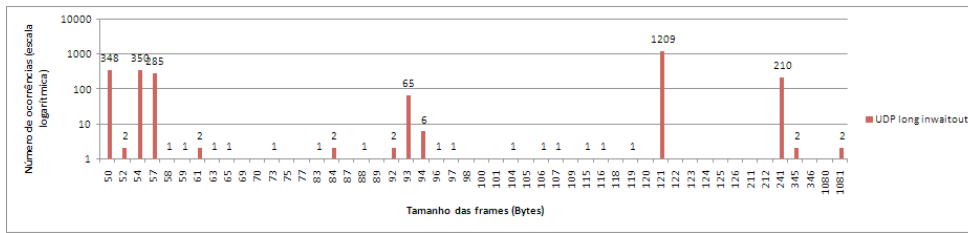


Figura 5.7: Histograma do tamanho das frames (UDP) do teste “long-inwaitout”.

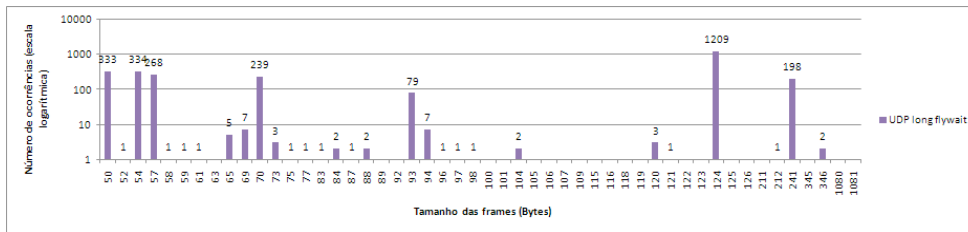


Figura 5.8: Histograma do tamanho das frames (UDP) do teste “long-flywait”.

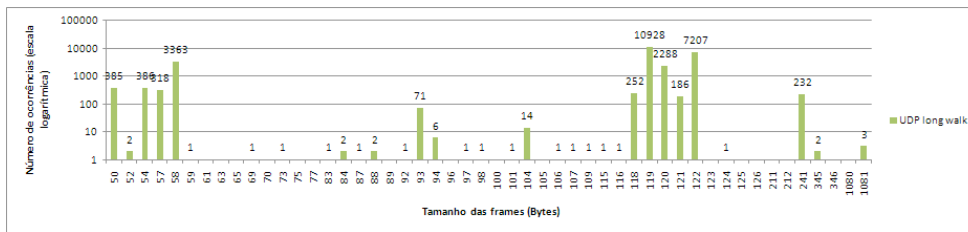


Figura 5.9: Histograma do tamanho das frames (UDP) do teste “long-walk”.

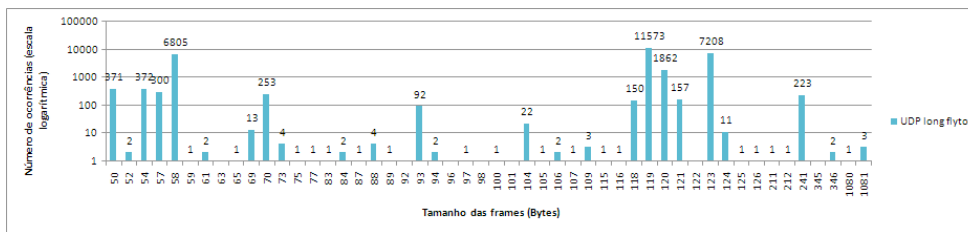


Figura 5.10: Histograma do tamanho das frames (UDP) do teste “long-flyto”.

Para uma melhor comparação visual dos histogramas dos tamanhos dos pacotes, foram elaboradas duas representações de todos os histogramas de cada tipo de tráfego (TCP, UDP), representadas nas figuras 5.11 e 5.12.

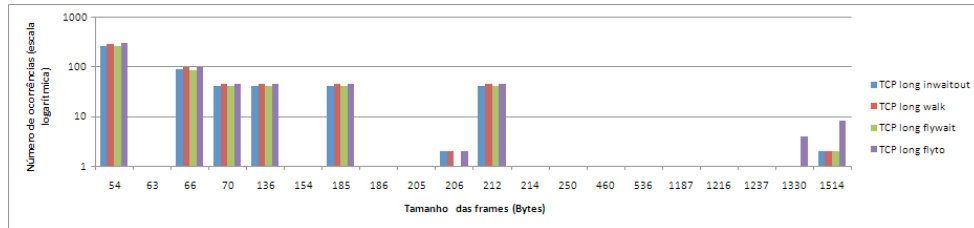


Figura 5.11: Histograma comparativo do tamanho das frames (TCP) dos quatro testes.

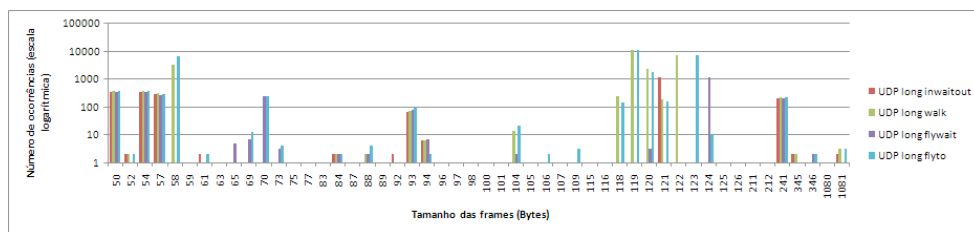


Figura 5.12: Histograma comparativo do tamanho das frames (UDP) dos quatro testes.

Para a representação do histograma dos intervalos de tempo entre chegada de frames (TCP) é apenas necessária a representação comparativa, pois os resultados são extremamente idênticos.

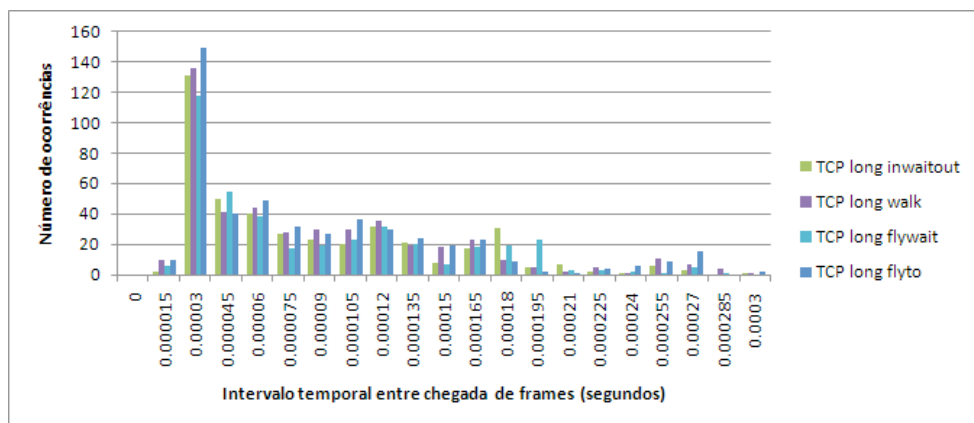


Figura 5.13: Histograma comparativo do intervalo de tempo entre chegada das frames (TCP) dos quatro testes.

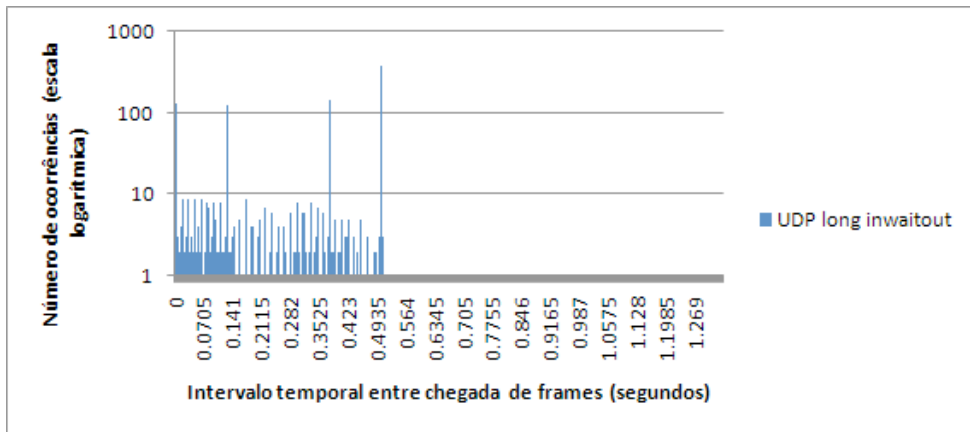


Figura 5.14: Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-inwaitout”.

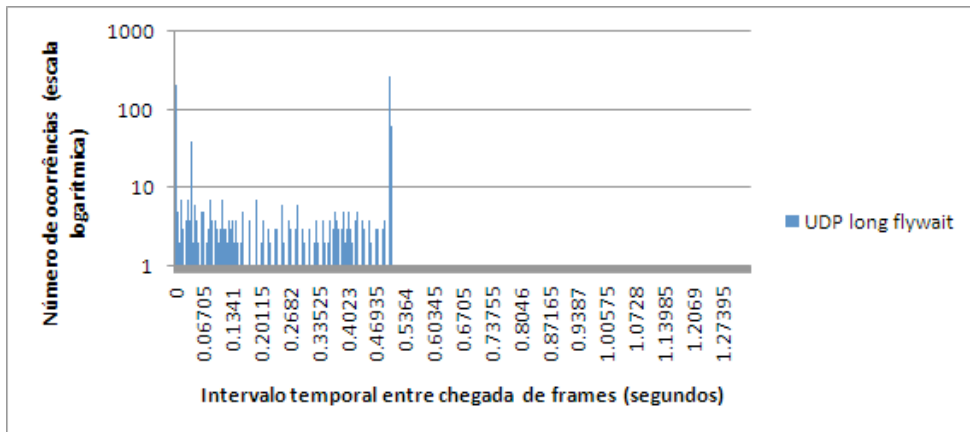


Figura 5.15: Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-flywait”.

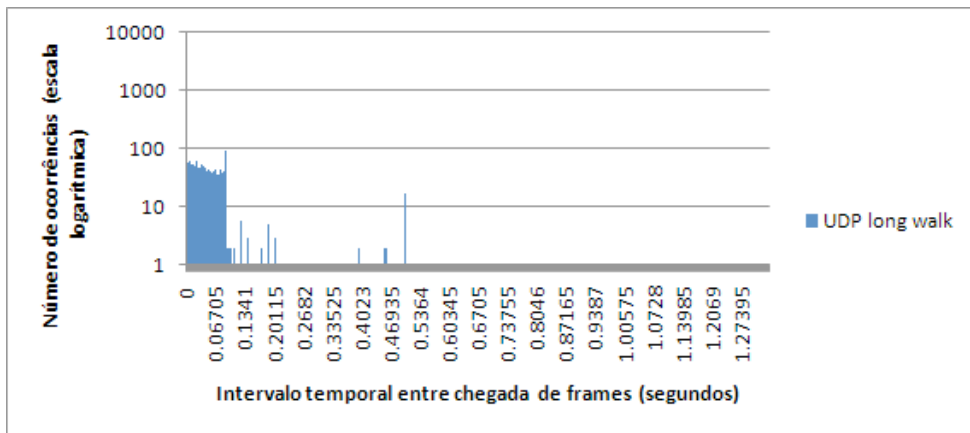


Figura 5.16: Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-walk”.

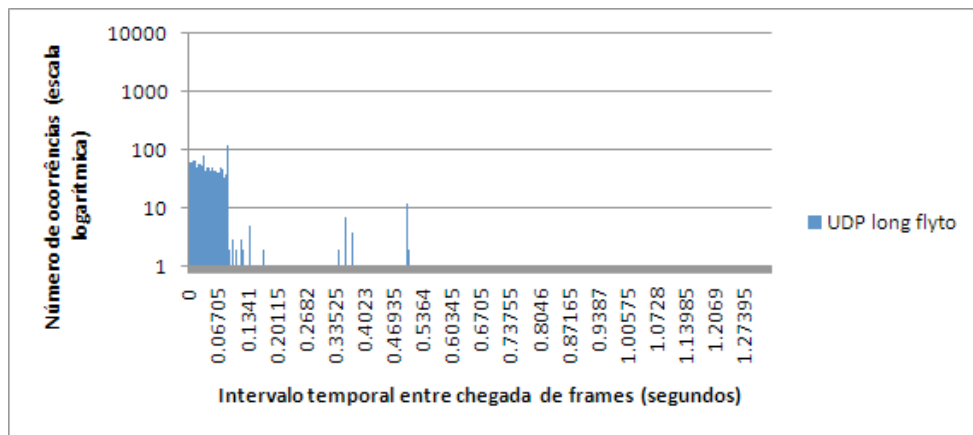


Figura 5.17: Histograma do intervalo de tempo entre chegada das frames (UDP) do teste “long-flyto”.

5.2.2.2 Testes repetitivos

Executando 8 *scripts*, 10 vezes cada um, sabendo que cada repetição gera 2 *logs* (TCP e UDP), e que de cada *log* se extraem 2 colunas de informação (tamanho das *frames* e intervalo temporal entre a *frame* actual e a anterior), obtêm-se 320 histogramas ($8 \times 10 \times 2 \times 2 = 320$). Processar tamanha quantidade de informação utilizando programas de folhas de cálculo, é impraticável. Para o correcto e rápido processamento de grandes quantidades de informação é necessário recorrer a linguagens de programação como o AWK. AWK é uma linguagem de programação para processamento de dados baseados em texto, sejam eles provenientes de arquivos ou de *streams* de dados (Wikipédia, 2009). Uma das implementações de AWK é o GAWK (Free Software Foundation, 2008). Foi criado um programa, nesta linguagem, para gerar os 320 histogramas necessários, a partir dos *logs* gerados pelo *SLNT*.

Representar graficamente 320 histogramas e compará-los visualmente seria moroso e pouco relevante. Tendo isto em mente foi criado outro programa para criar 32 tabelas (16 TCP e 16 UDP) representativas das diferenças absolutas entre os histogramas de cada repetição de cada teste (Anexo A). Aproveitando as diferenças absolutas e sabendo o número total de ocorrências de cada repetição, foram geradas outras 32 tabelas, desta vez relativas (resultado absoluto dividido pelo total de ocorrências do histograma da repetição referida em cada linha da tabela) (Anexo A).

5.2.3 Análise dos resultados

5.2.3.1 Testes de longa duração

Na análise aos histogramas do tamanho das *frames* (TCP), nota-se a extrema semelhança entre eles. Os testes foram efectuados tendo em conta apenas a autenticação e os movimentos do *avatar* e portanto o único tráfego TCP que é gerado deve-se às acções de *login* e *logout*. Nos histogramas do tamanho das *frames* (UDP) nota-se uma semelhança visual entre os histogramas que representam o *avatar* parado em pé e a voar, e entre os que representam o *avatar* a voar e a andar. No entanto apesar das semelhanças é possível, através da representação gráfica, distinguir com clareza cada uma das situações.

Na análise do intervalo de tempo entre chegada de *frames*, o caso TCP é muito semelhante em todas as experiências, confirmando mais uma vez que o tráfego TCP gerado deve-se apenas às acções de autenticação do utilizador. Em relação ao caso UDP são muito distintos os casos do *avatar* parado (em pé ou a voar) e em movimento (a pé ou a voar). A quantidade de *frames* por segundo é sem dúvida superior nos casos com movimento do que nos casos sem movimento do *avatar*.

Com estes testes consegue-se estabelecer uma relação entre o tráfego e as acções do *avatar*.

5.2.3.2 Testes repetitivos

A abordagem feita na análise deste grupo não se revelou ser a melhor. As percentagens representadas nas tabelas são exageradamente altas. Vários factores podem influenciar este resultado: 1) a resolução dos histogramas gerados foi arbitrada e portanto pode ser demasiado pequena de modo a gerar histogramas diferentes, por ter resolução a mais; 2) a duração temporal dos testes ser demasiado pequena para obter uma distribuição do tráfego real; 3) a duração temporal dos testes ser tão pequena, cuja repetição, sem intervalo, cause alguns problemas na autenticação com o servidor (caso algumas vezes verificado em acessos manuais).

5.3 Conclusões

A validação do funcionamento do *SLNT* foi efectuada com sucesso. O estudo da potencialidade de utilização permite obter as seguintes conclusões: 1) A análise de testes longos, apesar de não ter tido repetições, parece dar informações claras sobre o tráfego, relacionando-o com as acções do *avatar*. 2) A análise de testes pequenos com várias repetições permitiu verificar algumas falhas na análise que devem ser estudadas em trabalhos futuros. A escolha da resolução dos histogramas e o intervalo entre repetições deve ser pensado de forma a permitir configurar o *SLNT* de maneira a que repetições do mesmo *script* gerem o mesmo tráfego e portanto se obtenham histogramas semelhantes entre repetições.

O *SLNT* revelou-se em todo o caso extremamente útil. Executar todos estes testes manualmente, com o número de repetições mencionado, seria tarefa impossível no desenvolvimento desta tese.

Capítulo 6

Conclusões e Trabalho Futuro

Neste capítulo são apresentadas as conclusões e o trabalho futuro.

6.1 Satisfação dos Objectivos

Verificou-se que com a definição de uma arquitectura isolada (*offline*), e formada por apenas três elementos (cliente, servidor e observador de tráfego) se podem efectuar caracterizações do tráfego de MMORPGs e relacioná-las com as acções decorrentes no mundo virtual. Foi possível ainda concluir que, apesar de existirem projectos relevantes ao acesso a mundos virtuais sem o cliente gráfico, estão ainda numa fase de desenvolvimento inicial sem grande suporte. A automação do sistema permitiu tornar a execução de certos casos de uso, muito mais simples e eficaz, pelo que com o desenvolvimento a longo prazo, deste tipo de sistemas, se pode de facto obter em tempo útil, caracterizações de tráfego completas, importantes à infraestrutura que suporta as comunicações deste tipo de jogos (a internet). O sistema foi validado com sucesso. Provou-se a sua funcionalidade e também a relevância dos dados fornecidos, pois estes permitem obter informações importantes ao estudo e caracterização de jogos 3D *online*. Através desta dissertação, foi possível elaborar um *Sistema de caracterização da rede em jogos 3D online multi-jogador*. A concretização de um sistema deste género revelou-se um passo importante no estudo e caracterização de MMOGs já que permite reduzir o tempo dispendido pelo utilizador a obter dados para análise, alargando o espectro de hipóteses que o utilizador colocaria num estudo hipotético.

6.2 Trabalho Futuro

Como trabalho futuro, verificou-se ser necessário a continuação do desenvolvimento da ferramenta *SLNT*. Como melhorias sugerem-se: alargar a quantidade de informação fornecida (tamanho

dos pacotes e intervalo temporal entre *frames*), melhorar o *frontend* da aplicação e permitir o controle da captação de tráfego através dos *scripts* - isto permitiria, por exemplo, iniciar a recolha de dados depois do *login* e antes do *logout* e assim, retirar o tráfego gerado pela autenticação.

O software cliente utilizado é bastante instável o que sugere a criação de um software baseado na biblioteca *libopenmetaverse* ou a melhoria do *Testclient*. Com este passo pretende-se também alargar o leque de jogos alvo deste sistema.

Proceder a alterações no sistema de forma a permitir testes numa rede real e obter novos tipos de dados como atraso de propagação de pacotes, contabilização de perda de pacotes (pois a maioria é transmitida por UDP). Como sugestão de implementação deste novo tipo de teste, utilizar a plataforma PLANETLAB que permite a execução e teste reais de aplicações distribuídas.

Implementação de ferramentas de análise próprias no software, ou como aplicação auxiliar, de forma a tornar o processamento de dados mais fácil e acessível.

Estudar a resolução adequada para criação dos histogramas, de modo a obter histogramas semelhantes com repetições do mesmo teste.

Estudar o intervalo entre repetições óptimo, para evitar rejeições de conexão ao servidor, ou alterar o *SLNT* de forma a que este consiga detectar este tipo de situações e que, por exemplo, accione um mecanismo que registe o evento e que não contabilize a repetição como válida.

Fazer a análise de tráfego de um *script* que contenha múltiplas acções do *avatar*, ao longo do tempo, e verificar se com essa análise se consegue identificar que acções estão a ser desempenhadas a cada momento.

Anexo A

Tabelas das diferenças entre os histogramas de cada repetição dos testes repetitivos

Neste anexo estão representadas as 64 tabelas das diferenças, absolutas e relativas, entre os histogramas de cada repetição de determinado *Script*. As repetições de cada teste foram numeradas de 1 a 10, por ordem de execução. Cada valor das tabelas absolutas foi obtido calculando o somatório das diferenças absolutas entre cada valor do histograma. Os valores das tabelas relativas foram obtidos dividindo os valores das tabelas absolutas, pelo número de ocorrências da repetição indicada em cada linha.

Exemplo simplificado com dois histogramas representados apenas com os valores de cada índice:

Histograma 1	1	3	2	3
Histograma 2	3	2	2	

Tabela da diferença absoluta entre os histogramas 1 e 2:

Número do Histograma	1	2
1	0 = $ 1-1 + 3-3 + 2-2 + 3-3 $	6 = $ 1-3 + 3-2 + 2-2 + 3-0 $
2	6 = $ 3-1 + 2-3 + 2-2 + 0-3 $	0 = $ 3-3 + 2-2 + 2-2 $

Tabela da diferença relativa entre os histogramas 1 e 2:

Número do Histograma	1	2
1	0 = $0/(1+3+2+3)$	0.67 = $6/(1+3+2+3)$
2	0.86 = $6/(3+2+2)$	0 = $0/(3+2+2)$

A.1 Script Back

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	38	54	1	12	37	38	38	54	38	55
2	38	0	16	39	26	1	0	0	16	0	17
3	54	16	0	55	42	17	16	16	0	16	1
4	1	39	55	0	13	38	39	39	55	39	56
5	12	26	42	13	0	25	26	26	42	26	43
6	37	1	17	38	25	0	1	1	17	1	17
7	38	0	16	39	26	1	0	0	16	0	18
8	38	0	16	39	26	1	0	0	16	0	17
9	54	16	0	55	42	17	16	16	0	16	1
10	38	0	16	39	26	1	0	0	16	0	17

Tabela A.1: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “back” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,69	0,98	0,02	0,22	0,67	0,69	0,69	0,98	0,69
2	2,24	0,00	0,94	2,29	1,53	0,06	0,00	0,00	0,94	0,00
3	54,00	16,00	0,00	55,00	42,00	17,00	16,00	16,00	0,00	16,00
4	0,02	0,70	0,98	0,00	0,23	0,68	0,70	0,70	0,98	0,70
5	0,28	0,60	0,98	0,30	0,00	0,58	0,60	0,60	0,98	0,60
6	2,18	0,06	1,00	2,24	1,47	0,00	0,06	0,06	1,00	0,06
7	2,11	0,00	0,89	2,17	1,44	0,06	0,00	0,00	0,89	0,00
8	2,24	0,00	0,94	2,29	1,53	0,06	0,00	0,00	0,94	0,00
9	54,00	16,00	0,00	55,00	42,00	17,00	16,00	16,00	0,00	16,00
10	2,24	0,00	0,94	2,29	1,53	0,06	0,00	0,00	0,94	0,00

Tabela A.2: Diferenças relativas entre os histogramas de tamanhos (TCP) de “back” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	50	54	47	38	53	50	48	54	50	55
2	50	0	16	49	38	9	14	14	16	12	17
3	54	16	0	55	42	17	16	16	0	16	1
4	47	49	55	0	33	52	49	49	55	47	56
5	38	38	42	33	0	39	34	38	42	36	43
6	53	9	17	52	39	0	19	19	17	13	17
7	50	14	16	49	34	19	0	14	16	14	18
8	48	14	16	49	38	19	14	0	16	16	17
9	54	16	0	55	42	17	16	16	0	16	1
10	50	12	16	47	36	13	14	16	16	0	17

Tabela A.3: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “back” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,91	0,98	0,85	0,69	0,96	0,91	0,87	0,98	0,91
2	2,94	0,00	0,94	2,88	2,24	0,53	0,82	0,82	0,94	0,71
3	54,00	16,00	0,00	55,00	42,00	17,00	16,00	16,00	0,00	16,00
4	0,84	0,88	0,98	0,00	0,59	0,93	0,88	0,88	0,98	0,84
5	0,88	0,88	0,98	0,77	0,00	0,91	0,79	0,88	0,98	0,84
6	3,12	0,53	1,00	3,06	2,29	0,00	1,12	1,12	1,00	0,76
7	2,78	0,78	0,89	2,72	1,89	1,06	0,00	0,78	0,89	0,78
8	2,82	0,82	0,94	2,88	2,24	1,12	0,82	0,00	0,94	0,94
9	54,00	16,00	0,00	55,00	42,00	17,00	16,00	16,00	0,00	16,00
10	2,94	0,71	0,94	2,76	2,12	0,76	0,82	0,94	0,94	0,00

Tabela A.4: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “back” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	380	366	14	365	364	379	361	368	369	520
2	380	0	28	374	25	50	31	69	26	19	706
3	366	28	0	364	15	40	19	53	12	17	712
4	14	374	364	0	367	354	377	361	366	367	520
5	365	25	15	367	0	43	26	52	11	14	713
6	364	50	40	354	43	0	53	27	42	41	742
7	379	31	19	377	26	53	0	66	25	28	713
8	361	69	53	361	52	27	66	0	53	54	757
9	368	26	12	366	11	42	25	53	0	11	708
10	369	19	17	367	14	41	28	54	11	0	705

Tabela A.5: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “back” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,73	0,70	0,03	0,70	0,70	0,73	0,69	0,71	0,71
2	0,54	0,00	0,04	0,53	0,04	0,07	0,04	0,10	0,04	0,03
3	0,51	0,04	0,00	0,51	0,02	0,06	0,03	0,07	0,02	0,02
4	0,03	0,72	0,70	0,00	0,71	0,68	0,73	0,69	0,70	0,71
5	0,51	0,04	0,02	0,51	0,00	0,06	0,04	0,07	0,02	0,02
6	0,49	0,07	0,05	0,48	0,06	0,00	0,07	0,04	0,06	0,06
7	0,53	0,04	0,03	0,53	0,04	0,07	0,00	0,09	0,04	0,04
8	0,48	0,09	0,07	0,48	0,07	0,04	0,09	0,00	0,07	0,07
9	0,52	0,04	0,02	0,52	0,02	0,06	0,04	0,07	0,00	0,02
10	0,52	0,03	0,02	0,52	0,02	0,06	0,04	0,08	0,02	0,00

Tabela A.6: Diferenças relativas entre os histogramas de tamanhos (UDP) de “back” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	682	660	512	661	694	635	707	660	653	520
2	682	0	402	722	403	408	401	445	452	429	706
3	660	402	0	726	435	480	427	479	448	447	712
4	512	722	726	0	677	706	685	689	658	663	520
5	661	403	435	677	0	419	410	440	417	440	713
6	694	408	480	706	419	0	405	407	436	389	742
7	635	401	427	685	410	405	0	432	409	414	713
8	707	445	479	689	440	407	432	0	433	404	757
9	660	452	448	658	417	436	409	433	0	405	708
10	653	429	447	663	440	389	414	404	405	0	705

Tabela A.7: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “back” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,31	1,27	0,98	1,27	1,33	1,22	1,36	1,27	1,26
2	0,97	0,00	0,57	1,02	0,57	0,58	0,57	0,63	0,64	0,61
3	0,93	0,56	0,00	1,02	0,61	0,67	0,60	0,67	0,63	0,63
4	0,98	1,39	1,40	0,00	1,30	1,36	1,32	1,33	1,27	1,28
5	0,93	0,57	0,61	0,95	0,00	0,59	0,58	0,62	0,58	0,62
6	0,94	0,55	0,65	0,95	0,56	0,00	0,55	0,55	0,59	0,52
7	0,89	0,56	0,60	0,96	0,58	0,57	0,00	0,61	0,57	0,58
8	0,93	0,59	0,63	0,91	0,58	0,54	0,57	0,00	0,57	0,53
9	0,93	0,64	0,63	0,93	0,59	0,62	0,58	0,61	0,00	0,57
10	0,93	0,61	0,63	0,94	0,62	0,55	0,59	0,57	0,57	0,00

Tabela A.8: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “back” obtidos nas diferentes repetições.

A.2 *Script Fly*

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	6	38	0	38	5	38	38	0	38	39
2	6	0	44	6	44	1	44	44	6	44	45
3	38	44	0	38	0	43	0	0	38	0	1
4	0	6	38	0	38	5	38	38	0	38	39
5	38	44	0	38	0	43	0	0	38	0	1
6	5	1	43	5	43	0	43	43	5	43	1
7	38	44	0	38	0	43	0	0	38	0	44
8	38	44	0	38	0	43	0	0	38	0	1
9	0	6	38	0	38	5	38	38	0	38	39
10	38	44	0	38	0	43	0	0	38	0	1

Tabela A.9: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “fly” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,15	0,97	0,00	0,97	0,13	0,97	0,97	0,00	0,97
2	0,13	0,00	0,98	0,13	0,98	0,02	0,98	0,98	0,13	0,98
3	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00
4	0,00	0,15	0,97	0,00	0,97	0,13	0,97	0,97	0,00	0,97
5	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00
6	5,00	1,00	43,00	5,00	43,00	0,00	43,00	43,00	5,00	43,00
7	0,86	1,00	0,00	0,86	0,00	0,98	0,00	0,00	0,86	0,00
8	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00
9	0,00	0,15	0,97	0,00	0,97	0,13	0,97	0,97	0,00	0,97
10	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00

Tabela A.10: Diferenças relativas entre os histogramas de tamanhos (TCP) de “fly” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	34	38	28	38	33	38	38	26	38	39
2	34	0	44	28	44	31	44	44	30	44	45
3	38	44	0	38	0	43	0	0	38	0	1
4	28	28	38	0	38	29	38	38	26	38	39
5	38	44	0	38	0	43	0	0	38	0	1
6	33	31	43	29	43	0	43	43	33	43	1
7	38	44	0	38	0	43	0	0	38	0	44
8	38	44	0	38	0	43	0	0	38	0	1
9	26	30	38	26	38	33	38	38	0	38	39
10	38	44	0	38	0	43	0	0	38	0	1

Tabela A.11: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “fly” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,87	0,97	0,72	0,97	0,85	0,97	0,97	0,67	0,97
2	0,76	0,00	0,98	0,62	0,98	0,69	0,98	0,98	0,67	0,98
3	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00
4	0,72	0,72	0,97	0,00	0,97	0,74	0,97	0,97	0,67	0,97
5	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00
6	33,00	31,00	43,00	29,00	43,00	0,00	43,00	43,00	33,00	43,00
7	0,86	1,00	0,00	0,86	0,00	0,98	0,00	0,00	0,86	0,00
8	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00
9	0,67	0,77	0,97	0,67	0,97	0,85	0,97	0,97	0,00	0,97
10	38,00	44,00	0,00	38,00	0,00	43,00	0,00	0,00	38,00	0,00

Tabela A.12: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “fly” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	87	84	21	92	93	92	91	15	80	107
2	87	0	29	76	45	46	43	44	80	27	60
3	84	29	0	77	22	21	20	21	75	8	35
4	21	76	77	0	91	90	89	88	10	71	104
5	92	45	22	91	0	1	4	3	91	24	15
6	93	46	21	90	1	0	3	2	90	23	14
7	92	43	20	89	4	3	0	3	89	22	17
8	91	44	21	88	3	2	3	0	88	23	16
9	15	80	75	10	91	90	89	88	0	73	104
10	80	27	8	71	24	23	22	23	73	0	37

Tabela A.13: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “fly” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,81	0,79	0,20	0,86	0,87	0,86	0,85	0,14	0,75
2	1,45	0,00	0,48	1,27	0,75	0,77	0,72	0,73	1,33	0,45
3	2,40	0,83	0,00	2,20	0,63	0,60	0,57	0,60	2,14	0,23
4	0,20	0,73	0,74	0,00	0,88	0,87	0,86	0,85	0,10	0,68
5	6,13	3,00	1,47	6,07	0,00	0,07	0,27	0,20	6,07	1,60
6	6,64	3,29	1,50	6,43	0,07	0,00	0,21	0,14	6,43	1,64
7	5,41	2,53	1,18	5,24	0,24	0,18	0,00	0,18	5,24	1,29
8	5,69	2,75	1,31	5,50	0,19	0,13	0,19	0,00	5,50	1,44
9	0,14	0,77	0,72	0,10	0,88	0,87	0,86	0,85	0,00	0,70
10	2,16	0,73	0,22	1,92	0,65	0,62	0,59	0,62	1,97	0,00

Tabela A.14: Diferenças relativas entre os histogramas de tamanhos (UDP) de “fly” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	139	122	147	116	111	116	117	159	132	107
2	139	0	55	134	63	66	67	66	140	61	60
3	122	55	0	107	44	37	36	45	119	48	35
4	147	134	107	0	113	106	105	112	152	129	104
5	116	63	44	113	0	23	26	21	111	44	15
6	111	66	37	106	23	0	17	24	114	49	14
7	116	67	36	105	26	17	0	27	117	52	17
8	117	66	45	112	21	24	27	0	110	43	16
9	159	140	119	152	111	114	117	110	0	127	104
10	132	61	48	129	44	49	52	43	127	0	37

Tabela A.15: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “fly” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,30	1,14	1,37	1,08	1,04	1,08	1,09	1,49	1,23
2	2,32	0,00	0,92	2,23	1,05	1,10	1,12	1,10	2,33	1,02
3	3,49	1,57	0,00	3,06	1,26	1,06	1,03	1,29	3,40	1,37
4	1,41	1,29	1,03	0,00	1,09	1,02	1,01	1,08	1,46	1,24
5	7,73	4,20	2,93	7,53	0,00	1,53	1,73	1,40	7,40	2,93
6	7,93	4,71	2,64	7,57	1,64	0,00	1,21	1,71	8,14	3,50
7	6,82	3,94	2,12	6,18	1,53	1,00	0,00	1,59	6,88	3,06
8	7,31	4,13	2,81	7,00	1,31	1,50	1,69	0,00	6,88	2,69
9	1,53	1,35	1,14	1,46	1,07	1,10	1,13	1,06	0,00	1,22
10	3,57	1,65	1,30	3,49	1,19	1,32	1,41	1,16	3,43	0,00

Tabela A.16: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “fly” obtidos nas diferentes repetições.

A.3 Script Flyto

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	0	0	0	42	0	0	0	0	39	1
2	0	0	0	0	42	0	0	0	0	39	1
3	0	0	0	0	42	0	0	0	0	39	1
4	0	0	0	0	42	0	0	0	0	39	1
5	42	42	42	42	0	42	42	42	42	3	43
6	0	0	0	0	42	0	0	0	0	39	1
7	0	0	0	0	42	0	0	0	0	39	1
8	0	0	0	0	42	0	0	0	0	39	1
9	0	0	0	0	42	0	0	0	0	39	1
10	39	39	39	39	3	39	39	39	39	0	40

Tabela A.17: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “flyto” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
2	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
3	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
4	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
5	0,98	0,98	0,98	0,98	0,00	0,98	0,98	0,98	0,98	0,07
6	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
7	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
8	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
9	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
10	0,98	0,98	0,98	0,98	0,08	0,98	0,98	0,98	0,98	0,00

Tabela A.18: Diferenças relativas entre os histogramas de tamanhos (TCP) de “flyto” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	0	0	0	42	0	0	0	0	39	1
2	0	0	0	0	42	0	0	0	0	39	1
3	0	0	0	0	42	0	0	0	0	39	1
4	0	0	0	0	42	0	0	0	0	39	1
5	42	42	42	42	0	42	42	42	42	29	43
6	0	0	0	0	42	0	0	0	0	39	1
7	0	0	0	0	42	0	0	0	0	39	1
8	0	0	0	0	42	0	0	0	0	39	1
9	0	0	0	0	42	0	0	0	0	39	1
10	39	39	39	39	29	39	39	39	39	0	40

Tabela A.19: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “flyto” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
2	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
3	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
4	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
5	0,98	0,98	0,98	0,98	0,00	0,98	0,98	0,98	0,98	0,67
6	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
7	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
8	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
9	0,00	0,00	0,00	0,00	42,00	0,00	0,00	0,00	0,00	39,00
10	0,98	0,98	0,98	0,98	0,73	0,98	0,98	0,98	0,98	0,00

Tabela A.20: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “flyto” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	157	157	157	382	151	192	152	160	378	413
2	157	0	34	16	245	24	55	21	23	243	412
3	157	34	0	32	251	22	67	25	37	245	402
4	157	16	32	0	247	20	53	21	21	243	412
5	382	245	251	247	0	247	240	250	248	20	349
6	151	24	22	20	247	0	63	19	25	241	404
7	192	55	67	53	240	63	0	58	48	238	461
8	152	21	25	21	250	19	58	0	30	244	409
9	160	23	37	21	248	25	48	30	0	242	419
10	378	243	245	243	20	241	238	244	242	0	347

Tabela A.21: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “flyto” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,38	0,38	0,38	0,92	0,37	0,46	0,37	0,39	0,92
2	0,38	0,00	0,08	0,04	0,59	0,06	0,13	0,05	0,06	0,59
3	0,39	0,08	0,00	0,08	0,62	0,05	0,17	0,06	0,09	0,61
4	0,38	0,04	0,08	0,00	0,60	0,05	0,13	0,05	0,05	0,59
5	1,09	0,70	0,72	0,71	0,00	0,71	0,69	0,72	0,71	0,06
6	0,37	0,06	0,05	0,05	0,61	0,00	0,16	0,05	0,06	0,60
7	0,42	0,12	0,15	0,11	0,52	0,14	0,00	0,13	0,10	0,52
8	0,37	0,05	0,06	0,05	0,61	0,05	0,14	0,00	0,07	0,60
9	0,38	0,05	0,09	0,05	0,59	0,06	0,11	0,07	0,00	0,58
10	1,09	0,70	0,71	0,70	0,06	0,69	0,69	0,70	0,70	0,00

Tabela A.22: Diferenças relativas entre os histogramas de tamanhos (UDP) de “flyto” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	289	305	357	460	345	334	308	312	438	413
2	289	0	276	318	425	330	297	251	281	417	412
3	305	276	0	306	429	328	329	301	301	407	402
4	357	318	306	0	421	310	301	319	329	401	412
5	460	425	429	421	0	409	444	414	402	366	349
6	345	330	328	310	409	0	313	307	313	417	404
7	334	297	329	301	444	313	0	304	328	446	461
8	308	251	301	319	414	307	304	0	300	398	409
9	312	281	301	329	402	313	328	300	0	402	419
10	438	417	407	401	366	417	446	398	402	0	347

Tabela A.23: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “flyto” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,70	0,74	0,86	1,11	0,84	0,81	0,75	0,76	1,06
2	0,70	0,00	0,67	0,77	1,03	0,80	0,72	0,61	0,68	1,01
3	0,76	0,69	0,00	0,76	1,07	0,82	0,82	0,75	0,75	1,01
4	0,87	0,77	0,74	0,00	1,02	0,75	0,73	0,77	0,80	0,97
5	1,32	1,22	1,23	1,21	0,00	1,17	1,27	1,19	1,15	1,05
6	0,85	0,82	0,81	0,77	1,01	0,00	0,77	0,76	0,77	1,03
7	0,72	0,64	0,71	0,65	0,96	0,68	0,00	0,66	0,71	0,97
8	0,75	0,61	0,74	0,78	1,01	0,75	0,74	0,00	0,73	0,97
9	0,74	0,67	0,72	0,79	0,96	0,75	0,78	0,72	0,00	0,96
10	1,26	1,20	1,17	1,16	1,05	1,20	1,29	1,15	1,16	0,00

Tabela A.24: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “flyto” obtidos nas diferentes repetições.

A.4 *Script Forward*

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	24	40	40	24	45	27	24	27	41	17
2	24	0	16	16	0	21	3	0	3	17	39
3	40	16	0	0	16	19	13	16	13	1	55
4	40	16	0	0	16	19	13	16	13	1	55
5	24	0	16	16	0	21	3	0	3	17	39
6	45	21	19	19	21	0	20	21	20	20	58
7	27	3	13	13	3	20	0	3	0	14	42
8	24	0	16	16	0	21	3	0	3	17	39
9	27	3	13	13	3	20	0	3	0	14	42
10	41	17	1	1	17	20	14	17	14	0	56

Tabela A.25: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “forward” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,41	2,35	2,35	1,41	2,65	1,59	1,41	1,59	2,41
2	0,62	0,00	0,41	0,41	0,00	0,54	0,08	0,00	0,08	0,44
3	0,73	0,29	0,00	0,00	0,29	0,35	0,24	0,29	0,24	0,02
4	0,73	0,29	0,00	0,00	0,29	0,35	0,24	0,29	0,24	0,02
5	0,62	0,00	0,41	0,41	0,00	0,54	0,08	0,00	0,08	0,44
6	0,78	0,36	0,33	0,33	0,36	0,00	0,34	0,36	0,34	0,34
7	0,64	0,07	0,31	0,31	0,07	0,48	0,00	0,07	0,00	0,33
8	0,62	0,00	0,41	0,41	0,00	0,54	0,08	0,00	0,08	0,44
9	0,64	0,07	0,31	0,31	0,07	0,48	0,00	0,07	0,00	0,33
10	0,73	0,30	0,02	0,02	0,30	0,36	0,25	0,30	0,25	0,00

Tabela A.26: Diferenças relativas entre os histogramas de tamanhos (TCP) de “forward” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	32	46	44	32	47	35	30	33	49	17
2	32	0	38	30	24	35	37	26	21	29	39
3	46	38	0	44	46	41	45	42	33	41	55
4	44	30	44	0	40	31	51	38	33	35	55
5	32	24	46	40	0	43	37	30	27	39	39
6	47	35	41	31	43	0	50	45	36	34	58
7	35	37	45	51	37	50	0	37	36	50	42
8	30	26	42	38	30	45	37	0	29	41	39
9	33	21	33	33	27	36	36	29	0	34	42
10	49	29	41	35	39	34	50	41	34	0	56

Tabela A.27: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “forward” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,88	2,71	2,59	1,88	2,76	2,06	1,76	1,94	2,88
2	0,82	0,00	0,97	0,77	0,62	0,90	0,95	0,67	0,54	0,74
3	0,84	0,69	0,00	0,80	0,84	0,75	0,82	0,76	0,60	0,75
4	0,80	0,55	0,80	0,00	0,73	0,56	0,93	0,69	0,60	0,64
5	0,82	0,62	1,18	1,03	0,00	1,10	0,95	0,77	0,69	1,00
6	0,81	0,60	0,71	0,53	0,74	0,00	0,86	0,78	0,62	0,59
7	0,83	0,88	1,07	1,21	0,88	1,19	0,00	0,88	0,86	1,19
8	0,77	0,67	1,08	0,97	0,77	1,15	0,95	0,00	0,74	1,05
9	0,79	0,50	0,79	0,79	0,64	0,86	0,86	0,69	0,00	0,81
10	0,88	0,52	0,73	0,63	0,70	0,61	0,89	0,73	0,61	0,00

Tabela A.28: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “forward” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	501	498	492	737	501	489	503	496	497	600
2	501	0	13	31	264	18	22	26	13	20	535
3	498	13	0	24	259	23	15	21	10	13	526
4	492	31	24	0	265	37	21	23	20	21	522
5	737	264	259	265	0	268	270	262	255	266	523
6	501	18	23	37	268	0	34	20	29	26	535
7	489	22	15	21	270	34	0	34	19	18	525
8	503	26	21	23	262	20	34	0	21	24	527
9	496	13	10	20	255	29	19	21	0	21	524
10	497	20	13	21	266	26	18	24	21	0	529

Tabela A.29: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “forward” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,84	0,83	0,82	1,23	0,84	0,82	0,84	0,83	0,83
2	0,94	0,00	0,02	0,06	0,49	0,03	0,04	0,05	0,02	0,04
3	0,95	0,02	0,00	0,05	0,49	0,04	0,03	0,04	0,02	0,02
4	0,94	0,06	0,05	0,00	0,51	0,07	0,04	0,04	0,04	0,04
5	1,41	0,50	0,50	0,51	0,00	0,51	0,52	0,50	0,49	0,51
6	0,94	0,03	0,04	0,07	0,50	0,00	0,06	0,04	0,05	0,05
7	0,93	0,04	0,03	0,04	0,51	0,06	0,00	0,06	0,04	0,03
8	0,95	0,05	0,04	0,04	0,50	0,04	0,06	0,00	0,04	0,05
9	0,95	0,02	0,02	0,04	0,49	0,06	0,04	0,04	0,00	0,04
10	0,94	0,04	0,02	0,04	0,50	0,05	0,03	0,05	0,04	0,00

Tabela A.30: Diferenças relativas entre os histogramas de tamanhos (UDP) de “forward” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	659	652	668	657	681	645	655	706	639	600
2	659	0	429	455	514	484	462	534	559	514	535
3	652	429	0	452	515	455	473	505	546	509	526
4	668	455	452	0	531	439	459	493	554	507	522
5	657	514	515	531	0	506	520	448	473	450	523
6	681	484	455	439	506	0	506	508	529	486	535
7	645	462	473	459	520	506	0	552	585	538	525
8	655	534	505	493	448	508	552	0	477	446	527
9	706	559	546	554	473	529	585	477	0	497	524
10	639	514	509	507	450	486	538	446	497	0	529

Tabela A.31: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de frames, (UDP) de “forward” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,10	1,09	1,11	1,10	1,14	1,08	1,09	1,18	1,07
2	1,23	0,00	0,80	0,85	0,96	0,90	0,86	1,00	1,04	0,96
3	1,24	0,82	0,00	0,86	0,98	0,87	0,90	0,96	1,04	0,97
4	1,28	0,87	0,87	0,00	1,02	0,84	0,88	0,94	1,06	0,97
5	1,26	0,98	0,98	1,02	0,00	0,97	0,99	0,86	0,90	0,86
6	1,27	0,90	0,85	0,82	0,95	0,00	0,95	0,95	0,99	0,91
7	1,23	0,88	0,90	0,87	0,99	0,96	0,00	1,05	1,11	1,02
8	1,24	1,01	0,96	0,94	0,85	0,96	1,05	0,00	0,91	0,85
9	1,35	1,07	1,04	1,06	0,90	1,01	1,12	0,91	0,00	0,95
10	1,21	0,97	0,96	0,96	0,85	0,92	1,02	0,84	0,94	0,00

Tabela A.32: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de frames, (UDP) de “forward” obtidos nas diferentes repetições.

A.5 Script Inout

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	5	0	38	20	0	10	0	0	3	39
2	5	0	5	43	17	5	7	5	5	4	44
3	0	5	0	38	20	0	10	0	0	3	39
4	38	43	38	0	58	38	48	38	38	41	1
5	20	17	20	58	0	20	14	20	20	19	59
6	0	5	0	38	20	0	10	0	0	3	39
7	10	7	10	48	14	10	0	10	10	7	49
8	0	5	0	38	20	0	10	0	0	3	39
9	0	5	0	38	20	0	10	0	0	3	39
10	3	4	3	41	19	3	7	3	3	0	42

Tabela A.33: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “inout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,13	0,00	0,97	0,51	0,00	0,26	0,00	0,00	0,08
2	0,11	0,00	0,11	0,98	0,39	0,11	0,16	0,11	0,11	0,09
3	0,00	0,13	0,00	0,97	0,51	0,00	0,26	0,00	0,00	0,08
4	38,00	43,00	38,00	0,00	58,00	38,00	48,00	38,00	38,00	41,00
5	0,34	0,29	0,34	0,98	0,00	0,34	0,24	0,34	0,34	0,32
6	0,00	0,13	0,00	0,97	0,51	0,00	0,26	0,00	0,00	0,08
7	0,20	0,14	0,20	0,98	0,29	0,20	0,00	0,20	0,20	0,14
8	0,00	0,13	0,00	0,97	0,51	0,00	0,26	0,00	0,00	0,08
9	0,00	0,13	0,00	0,97	0,51	0,00	0,26	0,00	0,00	0,08
10	0,07	0,10	0,07	0,98	0,45	0,07	0,17	0,07	0,07	0,00

Tabela A.34: Diferenças relativas entre os histogramas de tamanhos (TCP) de “inout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	27	32	38	38	24	32	22	24	27	39
2	27	0	37	43	41	25	35	29	25	24	44
3	32	37	0	38	48	36	42	30	34	35	39
4	38	43	38	0	58	38	48	38	38	41	1
5	38	41	48	58	0	40	50	42	40	47	59
6	24	25	36	38	40	0	32	26	24	23	39
7	32	35	42	48	50	32	0	34	34	33	49
8	22	29	30	38	42	26	34	0	28	25	39
9	24	25	34	38	40	24	34	28	0	29	39
10	27	24	35	41	47	23	33	25	29	0	42

Tabela A.35: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “inout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,69	0,82	0,97	0,97	0,62	0,82	0,56	0,62	0,69
2	0,61	0,00	0,84	0,98	0,93	0,57	0,80	0,66	0,57	0,55
3	0,82	0,95	0,00	0,97	1,23	0,92	1,08	0,77	0,87	0,90
4	38,00	43,00	38,00	0,00	58,00	38,00	48,00	38,00	38,00	41,00
5	0,64	0,69	0,81	0,98	0,00	0,68	0,85	0,71	0,68	0,80
6	0,62	0,64	0,92	0,97	1,03	0,00	0,82	0,67	0,62	0,59
7	0,65	0,71	0,86	0,98	1,02	0,65	0,00	0,69	0,69	0,67
8	0,56	0,74	0,77	0,97	1,08	0,67	0,87	0,00	0,72	0,64
9	0,62	0,64	0,87	0,97	1,03	0,62	0,87	0,72	0,00	0,74
10	0,64	0,57	0,83	0,98	1,12	0,55	0,79	0,60	0,69	0,00

Tabela A.36: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “inout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	98	1	79	11	11	11	16	7	11	94
2	98	0	97	35	97	101	93	102	101	97	34
3	1	97	0	78	12	10	12	17	6	10	93
4	79	35	78	0	84	78	84	85	80	78	19
5	11	97	12	84	0	14	4	15	8	18	101
6	11	101	10	78	14	0	18	11	8	4	95
7	11	93	12	84	4	18	0	15	12	14	101
8	16	102	17	85	15	11	15	0	17	11	104
9	7	101	6	80	8	8	12	17	0	12	95
10	11	97	10	78	18	4	14	11	12	0	95

Tabela A.37: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “inout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,04	0,01	0,84	0,12	0,12	0,12	0,17	0,07	0,12
2	2,88	0,00	2,85	1,03	2,85	2,97	2,74	3,00	2,97	2,85
3	0,01	1,04	0,00	0,84	0,13	0,11	0,13	0,18	0,06	0,11
4	4,16	1,84	4,11	0,00	4,42	4,11	4,42	4,47	4,21	4,11
5	0,11	0,96	0,12	0,83	0,00	0,14	0,04	0,15	0,08	0,18
6	0,12	1,06	0,11	0,82	0,15	0,00	0,19	0,12	0,08	0,04
7	0,11	0,92	0,12	0,83	0,04	0,18	0,00	0,15	0,12	0,14
8	0,15	0,98	0,16	0,82	0,14	0,11	0,14	0,00	0,16	0,11
9	0,07	1,06	0,06	0,84	0,08	0,08	0,13	0,18	0,00	0,13
10	0,12	1,02	0,11	0,82	0,19	0,04	0,15	0,12	0,13	0,00

Tabela A.38: Diferenças relativas entre os histogramas de tamanhos (UDP) de “inout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	112	137	103	139	139	151	136	143	143	94
2	112	0	109	47	115	115	119	118	117	113	34
3	137	109	0	104	158	138	154	147	140	144	93
4	103	47	104	0	106	104	110	113	104	108	19
5	139	115	158	106	0	150	150	147	148	140	101
6	139	115	138	104	150	0	154	143	136	138	95
7	151	119	154	110	150	154	0	167	146	146	101
8	136	118	147	113	147	143	167	0	147	143	104
9	143	117	140	104	148	136	146	147	0	128	95
10	143	113	144	108	140	138	146	143	128	0	95

Tabela A.39: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “inout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,19	1,46	1,10	1,48	1,48	1,61	1,45	1,52	1,52
2	3,29	0,00	3,21	1,38	3,38	3,38	3,50	3,47	3,44	3,32
3	1,47	1,17	0,00	1,12	1,70	1,48	1,66	1,58	1,51	1,55
4	5,42	2,47	5,47	0,00	5,58	5,47	5,79	5,95	5,47	5,68
5	1,38	1,14	1,56	1,05	0,00	1,49	1,49	1,46	1,47	1,39
6	1,46	1,21	1,45	1,09	1,58	0,00	1,62	1,51	1,43	1,45
7	1,50	1,18	1,52	1,09	1,49	1,52	0,00	1,65	1,45	1,45
8	1,31	1,13	1,41	1,09	1,41	1,38	1,61	0,00	1,41	1,38
9	1,51	1,23	1,47	1,09	1,56	1,43	1,54	1,55	0,00	1,35
10	1,51	1,19	1,52	1,14	1,47	1,45	1,54	1,51	1,35	0,00

Tabela A.40: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “inout” obtidos nas diferentes repetições.

A.6 *Script Inwaitout*

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	1	0	0	7	38	34	5	20	0	87
2	1	0	1	1	6	39	33	4	19	1	88
3	0	1	0	0	7	38	34	5	20	0	94
4	0	1	0	0	7	38	34	5	20	0	87
5	7	6	7	7	0	45	29	4	13	7	87
6	38	39	38	38	45	0	72	43	58	38	49
7	34	33	34	34	29	72	0	31	16	34	119
8	5	4	5	5	4	43	31	0	17	5	92
9	20	19	20	20	13	58	16	17	0	20	107
10	0	1	0	0	7	38	34	5	20	0	87

Tabela A.41: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “inwaitout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,01	0,00	0,00	0,08	0,44	0,39	0,06	0,23	0,00
2	0,01	0,00	0,01	0,01	0,07	0,44	0,38	0,05	0,22	0,01
3	0,00	0,01	0,00	0,00	0,07	0,40	0,36	0,05	0,21	0,00
4	0,00	0,01	0,00	0,00	0,08	0,44	0,39	0,06	0,23	0,00
5	0,08	0,07	0,08	0,08	0,00	0,52	0,33	0,05	0,15	0,08
6	0,78	0,80	0,78	0,78	0,92	0,00	1,47	0,88	1,18	0,78
7	0,29	0,28	0,29	0,29	0,24	0,61	0,00	0,26	0,13	0,29
8	0,05	0,04	0,05	0,05	0,04	0,47	0,34	0,00	0,18	0,05
9	0,19	0,18	0,19	0,19	0,12	0,54	0,15	0,16	0,00	0,19
10	0,00	0,01	0,00	0,00	0,08	0,44	0,39	0,06	0,23	0,00

Tabela A.42: Diferenças relativas entre os histogramas de tamanhos (TCP) de “inwaitout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	47	50	50	55	66	80	49	72	58	87
2	47	0	53	65	60	63	81	62	81	61	88
3	50	53	0	62	51	64	74	57	74	46	94
4	50	65	62	0	61	66	80	63	68	50	87
5	55	60	51	61	0	69	79	50	69	49	87
6	66	63	64	66	69	0	92	65	84	68	49
7	80	81	74	80	79	92	0	65	68	70	119
8	49	62	57	63	50	65	65	0	65	55	92
9	72	81	74	68	69	84	68	65	0	68	107
10	58	61	46	50	49	68	70	55	68	0	87

Tabela A.43: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “inwaitout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,54	0,57	0,57	0,63	0,76	0,92	0,56	0,83	0,67
2	0,53	0,00	0,60	0,74	0,68	0,72	0,92	0,70	0,92	0,69
3	0,53	0,56	0,00	0,66	0,54	0,68	0,79	0,61	0,79	0,49
4	0,57	0,75	0,71	0,00	0,70	0,76	0,92	0,72	0,78	0,57
5	0,63	0,69	0,59	0,70	0,00	0,79	0,91	0,57	0,79	0,56
6	1,35	1,29	1,31	1,35	1,41	0,00	1,88	1,33	1,71	1,39
7	0,67	0,68	0,62	0,67	0,66	0,77	0,00	0,55	0,57	0,59
8	0,53	0,67	0,62	0,68	0,54	0,71	0,71	0,00	0,71	0,60
9	0,67	0,76	0,69	0,64	0,64	0,79	0,64	0,61	0,00	0,64
10	0,67	0,70	0,53	0,57	0,56	0,78	0,80	0,63	0,78	0,00

Tabela A.44: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “inwaitout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	12	278	16	16	624	362	14	15	11	346
2	12	0	274	12	10	616	358	12	5	9	340
3	278	274	0	276	266	362	616	274	275	277	346
4	16	12	276	0	10	624	352	2	7	13	348
5	16	10	266	10	0	620	352	12	11	13	346
6	624	616	362	624	620	0	300	622	617	623	540
7	362	358	616	352	352	300	0	354	355	357	576
8	14	12	274	2	12	622	354	0	7	15	348
9	15	5	275	7	11	617	355	7	0	14	341
10	11	9	277	13	13	623	357	15	14	0	347

Tabela A.45: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “inwaitout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,03	0,80	0,05	0,05	1,80	1,05	0,04	0,04	0,03
2	0,04	0,00	0,81	0,04	0,03	1,81	1,05	0,04	0,01	0,03
3	0,80	0,79	0,00	0,80	0,77	1,05	1,78	0,79	0,79	0,80
4	0,05	0,03	0,79	0,00	0,03	1,79	1,01	0,01	0,02	0,04
5	0,05	0,03	0,77	0,03	0,00	1,79	1,02	0,03	0,03	0,04
6	1,16	1,14	0,67	1,16	1,15	0,00	0,56	1,15	1,14	1,15
7	0,63	0,62	1,07	0,61	0,61	0,52	0,00	0,61	0,62	0,62
8	0,04	0,03	0,79	0,01	0,03	1,79	1,02	0,00	0,02	0,04
9	0,04	0,01	0,81	0,02	0,03	1,81	1,04	0,02	0,00	0,04
10	0,03	0,03	0,80	0,04	0,04	1,80	1,03	0,04	0,04	0,00

Tabela A.46: Diferenças relativas entre os histogramas de tamanhos (UDP) de “inwaitout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	518	454	492	446	686	742	440	471	493	346
2	518	0	534	536	476	750	758	548	485	487	340
3	454	534	0	430	422	670	724	438	479	507	346
4	492	536	430	0	432	684	726	466	481	523	348
5	446	476	422	432	0	662	698	470	435	445	346
6	686	750	670	684	662	0	444	688	719	715	540
7	742	758	724	726	698	444	0	702	765	747	576
8	440	548	438	466	470	688	702	0	473	451	348
9	471	485	479	481	435	719	765	473	0	450	341
10	493	487	507	523	445	715	747	451	450	0	347

Tabela A.47: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de frames, (UDP) de “inwaitout” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,50	1,31	1,42	1,29	1,98	2,14	1,27	1,36	1,42
2	1,52	0,00	1,57	1,58	1,40	2,21	2,23	1,61	1,43	1,43
3	1,31	1,54	0,00	1,24	1,22	1,94	2,09	1,27	1,38	1,47
4	1,41	1,54	1,24	0,00	1,24	1,97	2,09	1,34	1,38	1,50
5	1,29	1,38	1,22	1,25	0,00	1,91	2,02	1,36	1,26	1,29
6	1,27	1,39	1,24	1,27	1,23	0,00	0,82	1,27	1,33	1,32
7	1,29	1,32	1,26	1,26	1,21	0,77	0,00	1,22	1,33	1,30
8	1,26	1,57	1,26	1,34	1,35	1,98	2,02	0,00	1,36	1,30
9	1,38	1,42	1,40	1,41	1,28	2,11	2,24	1,39	0,00	1,32
10	1,42	1,40	1,46	1,51	1,28	2,06	2,15	1,30	1,30	0,00

Tabela A.48: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de frames, (TCP) de “inwaitout” obtidos nas diferentes repetições.

A.7 Script Left

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	0	1	27	1	1	37	1	1	17	18
2	0	0	1	27	1	1	37	1	1	17	17
3	1	1	0	28	0	0	38	0	0	16	18
4	27	27	28	0	28	28	16	28	28	38	39
5	1	1	0	28	0	0	38	0	0	16	17
6	1	1	0	28	0	0	38	0	0	16	17
7	37	37	38	16	38	38	0	38	38	54	55
8	1	1	0	28	0	0	38	0	0	16	17
9	1	1	0	28	0	0	38	0	0	16	17
10	17	17	16	38	16	16	54	16	16	0	1

Tabela A.49: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “left” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,00	0,06	1,50	0,06	0,06	2,06	0,06	0,06	0,94
2	0,00	0,00	0,06	1,59	0,06	0,06	2,18	0,06	0,06	1,00
3	0,06	0,06	0,00	1,56	0,00	0,00	2,11	0,00	0,00	0,89
4	0,69	0,69	0,72	0,00	0,72	0,72	0,41	0,72	0,72	0,97
5	0,06	0,06	0,00	1,65	0,00	0,00	2,24	0,00	0,00	0,94
6	0,06	0,06	0,00	1,65	0,00	0,00	2,24	0,00	0,00	0,94
7	0,67	0,67	0,69	0,29	0,69	0,69	0,00	0,69	0,69	0,98
8	0,06	0,06	0,00	1,65	0,00	0,00	2,24	0,00	0,00	0,94
9	0,06	0,06	0,00	1,65	0,00	0,00	2,24	0,00	0,00	0,94
10	17,00	17,00	16,00	38,00	16,00	16,00	54,00	16,00	16,00	0,00

Tabela A.50: Diferenças relativas entre os histogramas de tamanhos (TCP) de “left” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	10	13	35	11	11	47	15	11	17	18
2	10	0	15	33	11	9	47	13	11	17	17
3	13	15	0	34	16	14	50	14	16	16	18
4	35	33	34	0	36	32	46	32	32	38	39
5	11	11	16	36	0	12	48	14	12	16	17
6	11	9	14	32	12	0	46	12	10	16	17
7	47	47	50	46	48	46	0	48	44	54	55
8	15	13	14	32	14	12	48	0	10	16	17
9	11	11	16	32	12	10	44	10	0	16	17
10	17	17	16	38	16	16	54	16	16	0	1

Tabela A.51: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “left” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,56	0,72	1,94	0,61	0,61	2,61	0,83	0,61	0,94
2	0,59	0,00	0,88	1,94	0,65	0,53	2,76	0,76	0,65	1,00
3	0,72	0,83	0,00	1,89	0,89	0,78	2,78	0,78	0,89	0,89
4	0,90	0,85	0,87	0,00	0,92	0,82	1,18	0,82	0,82	0,97
5	0,65	0,65	0,94	2,12	0,00	0,71	2,82	0,82	0,71	0,94
6	0,65	0,53	0,82	1,88	0,71	0,00	2,71	0,71	0,59	0,94
7	0,85	0,85	0,91	0,84	0,87	0,84	0,00	0,87	0,80	0,98
8	0,88	0,76	0,82	1,88	0,82	0,71	2,82	0,00	0,59	0,94
9	0,65	0,65	0,94	1,88	0,71	0,59	2,59	0,59	0,00	0,94
10	17,00	17,00	16,00	38,00	16,00	16,00	54,00	16,00	16,00	0,00

Tabela A.52: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “left” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	49	91	376	96	49	93	48	47	53	669
2	49	0	62	377	67	20	64	27	20	24	696
3	91	62	0	375	25	54	42	61	60	54	752
4	376	377	375	0	368	367	375	374	371	379	523
5	96	67	25	368	0	55	41	64	61	55	757
6	49	20	54	367	55	0	50	15	10	16	710
7	93	64	42	375	41	50	0	53	52	52	754
8	48	27	61	374	64	15	53	0	11	21	711
9	47	20	60	371	61	10	52	11	0	16	708
10	53	24	54	379	55	16	52	21	16	0	708

Tabela A.53: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “left” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,07	0,14	0,56	0,14	0,07	0,14	0,07	0,07	0,08
2	0,07	0,00	0,09	0,54	0,10	0,03	0,09	0,04	0,03	0,03
3	0,12	0,08	0,00	0,50	0,03	0,07	0,06	0,08	0,08	0,07
4	0,72	0,72	0,72	0,00	0,70	0,70	0,72	0,72	0,71	0,72
5	0,13	0,09	0,03	0,49	0,00	0,07	0,05	0,08	0,08	0,07
6	0,07	0,03	0,08	0,52	0,08	0,00	0,07	0,02	0,01	0,02
7	0,12	0,08	0,06	0,50	0,05	0,07	0,00	0,07	0,07	0,07
8	0,07	0,04	0,09	0,53	0,09	0,02	0,07	0,00	0,02	0,03
9	0,07	0,03	0,08	0,52	0,09	0,01	0,07	0,02	0,00	0,02
10	0,07	0,03	0,08	0,54	0,08	0,02	0,07	0,03	0,02	0,00

Tabela A.54: Diferenças relativas entre os histogramas de tamanhos (UDP) de “left” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	495	477	636	424	455	509	430	475	537	669
2	495	0	490	667	519	452	494	461	462	510	696
3	477	490	0	669	487	422	490	431	486	502	752
4	636	667	669	0	696	693	691	670	721	713	523
5	424	519	487	696	0	455	515	464	469	493	757
6	455	452	422	693	455	0	490	409	434	460	710
7	509	494	490	691	515	490	0	481	536	542	754
8	430	461	431	670	464	409	481	0	433	467	711
9	475	462	486	721	469	434	536	433	0	454	708
10	537	510	502	713	493	460	542	467	454	0	708

Tabela A.55: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “left” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,74	0,71	0,95	0,63	0,68	0,76	0,64	0,71	0,80
2	0,71	0,00	0,70	0,96	0,75	0,65	0,71	0,66	0,66	0,73
3	0,63	0,65	0,00	0,89	0,65	0,56	0,65	0,57	0,65	0,67
4	1,22	1,28	1,28	0,00	1,33	1,33	1,32	1,28	1,38	1,36
5	0,56	0,69	0,64	0,92	0,00	0,60	0,68	0,61	0,62	0,65
6	0,64	0,64	0,59	0,98	0,64	0,00	0,69	0,58	0,61	0,65
7	0,68	0,66	0,65	0,92	0,68	0,65	0,00	0,64	0,71	0,72
8	0,60	0,65	0,61	0,94	0,65	0,58	0,68	0,00	0,61	0,66
9	0,67	0,65	0,69	1,02	0,66	0,61	0,76	0,61	0,00	0,64
10	0,76	0,72	0,71	1,01	0,70	0,65	0,77	0,66	0,64	0,00

Tabela A.56: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “left” obtidos nas diferentes repetições.

A.8 *Script Right*

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	39	14	16	17	1	39	4	39	39	56
2	39	0	29	29	28	38	0	37	0	0	17
3	14	29	0	2	3	13	29	10	29	29	42
4	16	29	2	0	1	15	29	12	29	29	40
5	17	28	3	1	0	16	28	13	28	28	39
6	1	38	13	15	16	0	38	3	38	38	55
7	39	0	29	29	28	38	0	37	0	0	17
8	4	37	10	12	13	3	37	0	37	37	52
9	39	0	29	29	28	38	0	37	0	0	17
10	39	0	29	29	28	38	0	37	0	0	17

Tabela A.57: Diferenças absolutas entre os histogramas de tamanhos (TCP) de “right” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,70	0,25	0,29	0,30	0,02	0,70	0,07	0,70	0,70
2	2,29	0,00	1,71	1,71	1,65	2,24	0,00	2,18	0,00	0,00
3	0,33	0,69	0,00	0,05	0,07	0,31	0,69	0,24	0,69	0,69
4	0,40	0,73	0,05	0,00	0,03	0,38	0,73	0,30	0,73	0,73
5	0,44	0,72	0,08	0,03	0,00	0,41	0,72	0,33	0,72	0,72
6	0,02	0,69	0,24	0,27	0,29	0,00	0,69	0,05	0,69	0,69
7	2,29	0,00	1,71	1,71	1,65	2,24	0,00	2,18	0,00	0,00
8	0,08	0,71	0,19	0,23	0,25	0,06	0,71	0,00	0,71	0,71
9	2,29	0,00	1,71	1,71	1,65	2,24	0,00	2,18	0,00	0,00
10	2,29	0,00	1,71	1,71	1,65	2,24	0,00	2,18	0,00	0,00

Tabela A.58: Diferenças relativas entre os histogramas de tamanhos (TCP) de “right” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	47	40	36	41	41	49	42	47	49	56
2	47	0	39	31	32	48	12	45	12	12	17
3	40	39	0	26	33	41	39	38	41	39	42
4	36	31	26	0	29	43	37	46	35	35	40
5	41	32	33	29	0	40	34	43	36	34	39
6	41	48	41	43	40	0	50	47	48	50	55
7	49	12	39	37	34	50	0	43	12	6	17
8	42	45	38	46	43	47	43	0	49	45	52
9	47	12	41	35	36	48	12	49	0	10	17
10	49	12	39	35	34	50	6	45	10	0	17

Tabela A.59: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “right” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,84	0,71	0,64	0,73	0,73	0,88	0,75	0,84	0,88
2	2,76	0,00	2,29	1,82	1,88	2,82	0,71	2,65	0,71	0,71
3	0,95	0,93	0,00	0,62	0,79	0,98	0,93	0,90	0,98	0,93
4	0,90	0,78	0,65	0,00	0,73	1,08	0,93	1,15	0,88	0,88
5	1,05	0,82	0,85	0,74	0,00	1,03	0,87	1,10	0,92	0,87
6	0,75	0,87	0,75	0,78	0,73	0,00	0,91	0,85	0,87	0,91
7	2,88	0,71	2,29	2,18	2,00	2,94	0,00	2,53	0,71	0,35
8	0,81	0,87	0,73	0,88	0,83	0,90	0,83	0,00	0,94	0,87
9	2,76	0,71	2,41	2,06	2,12	2,82	0,71	2,88	0,00	0,59
10	2,88	0,71	2,29	2,06	2,00	2,94	0,35	2,65	0,59	0,00

Tabela A.60: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (TCP) de “right” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	375	11	265	253	19	379	368	380	374	523
2	375	0	372	602	598	370	16	63	15	15	706
3	11	372	0	260	250	14	374	359	375	369	520
4	265	602	260	0	20	262	612	615	613	603	528
5	253	598	250	20	0	250	604	599	605	597	522
6	19	370	14	262	250	0	368	361	371	367	524
7	379	16	374	612	604	368	0	55	13	13	712
8	368	63	359	615	599	361	55	0	56	52	757
9	380	15	375	613	605	371	13	56	0	16	709
10	374	15	369	603	597	367	13	52	16	0	713

Tabela A.61: Diferenças absolutas entre os histogramas de tamanhos (UDP) de “right” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	0,72	0,02	0,51	0,48	0,04	0,72	0,70	0,73	0,72
2	0,53	0,00	0,53	0,85	0,85	0,52	0,02	0,09	0,02	0,02
3	0,02	0,72	0,00	0,50	0,48	0,03	0,72	0,69	0,72	0,71
4	0,50	1,14	0,49	0,00	0,04	0,50	1,16	1,16	1,16	1,14
5	0,48	1,15	0,48	0,04	0,00	0,48	1,16	1,15	1,16	1,14
6	0,04	0,71	0,03	0,50	0,48	0,00	0,70	0,69	0,71	0,70
7	0,53	0,02	0,53	0,86	0,85	0,52	0,00	0,08	0,02	0,02
8	0,49	0,08	0,47	0,81	0,79	0,48	0,07	0,00	0,07	0,07
9	0,54	0,02	0,53	0,86	0,85	0,52	0,02	0,08	0,00	0,02
10	0,52	0,02	0,52	0,85	0,84	0,51	0,02	0,07	0,02	0,00

Tabela A.62: Diferenças relativas entre os histogramas de tamanhos (UDP) de “right” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10	Total de ocorrências
1	0	701	543	489	507	507	657	712	644	674	523
2	701	0	630	678	674	702	490	463	463	425	706
3	543	630	0	496	486	550	676	695	699	659	520
4	489	678	496	0	482	474	656	687	639	653	528
5	507	674	486	482	0	506	628	671	667	645	522
6	507	702	550	474	506	0	680	709	647	681	524
7	657	490	676	656	628	680	0	421	421	439	712
8	712	463	695	687	671	709	421	0	444	412	757
9	644	463	699	639	667	647	421	444	0	382	709
10	674	425	659	653	645	681	439	412	382	0	713

Tabela A.63: Diferenças absolutas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “right” obtidos nas diferentes repetições.

Repetição do teste	1	2	3	4	5	6	7	8	9	10
1	0,00	1,34	1,04	0,93	0,97	0,97	1,26	1,36	1,23	1,29
2	0,99	0,00	0,89	0,96	0,95	0,99	0,69	0,66	0,66	0,60
3	1,04	1,21	0,00	0,95	0,93	1,06	1,30	1,34	1,34	1,27
4	0,93	1,28	0,94	0,00	0,91	0,90	1,24	1,30	1,21	1,24
5	0,97	1,29	0,93	0,92	0,00	0,97	1,20	1,29	1,28	1,24
6	0,97	1,34	1,05	0,90	0,97	0,00	1,30	1,35	1,23	1,30
7	0,92	0,69	0,95	0,92	0,88	0,96	0,00	0,59	0,59	0,62
8	0,94	0,61	0,92	0,91	0,89	0,94	0,56	0,00	0,59	0,54
9	0,91	0,65	0,99	0,90	0,94	0,91	0,59	0,63	0,00	0,54
10	0,95	0,60	0,92	0,92	0,90	0,96	0,62	0,58	0,54	0,00

Tabela A.64: Diferenças relativas entre os histogramas dos intervalos temporais, entre chegada de *frames*, (UDP) de “right” obtidos nas diferentes repetições.

Referências

- Opensimulator, 2009. Disponível em <http://opensimulator.org>, acessido a última vez em Junho de 2009.
- Mindark PE AB. Entropia universe portal, 2009. Disponível em <http://www.entropiauniverse.com>, acessido a última vez em Fevereiro de 2009.
- Inc. Blizzard Entertainment. World of warcraft community site, 2009. Disponível em <http://www.worldofwarcraft.com>, acessido a última vez em Fevereiro de 2009.
- Mark Claypool. The effect of latency on user performance in real-time strategy games. Technical report, Worcester Polytechnic Institute, USA, 2005.
- Douglas E. Comer. *Internetworking with TCP/IP*. Upper Saddle River, NJ : Prentice-Hall, Fourth edition, 2000. ISBN 0-13-018380-6.
- NCsoft Corporation. Lineage ii celebrates five-year anniversary with largest content expansion to date | ncsoft, 2009. Disponível em <http://us.ncsoft.com/en/news/press-releases/lineage-ii-celebrates-five-year-anniversary-with-largest-content-expansion-to-date.html>, acessido a última vez em Junho de 2009.
- Centro de competência malha atlântica. O que é a internet?, 2005. Disponível em <http://www.malha.net/index.php?option=content&task=view&id=29&Itemid=>, acessido a última vez em Fevereiro de 2009.
- Blizzard Entertainment. Blizzard entertainment - media alert, 2009. Disponível em <http://www.blizzard.com/us/press/081028.html>, acessido a última vez em Junho de 2009.
- Open Metaverse Foundation. libsecondlife, 2009a. Disponível em <http://lib.openmetaverse.org/>, acessido a última vez em Junho de 2009.
- Open Metaverse Foundation. Libopenmetaverse | open metaverse foundation, 2009b. Disponível em <http://www.openmetaverse.org/projects/libopenmetaverse>, acessido a última vez em Junho de 2009.
- Wireshark Foundation. Wireshark: Go deep., 2009c. Disponível em <http://www.wireshark.org/>, acessido a última vez em Junho de 2009.
- Inc. Free Software Foundation. Gawk - gnu project - free software foundation (fsf), 2008. Disponível em <http://www.gnu.org/software/gawk/>, acessido a última vez em Junho de 2009.

- Dukhyun Chang-Taekyoung Kwon Yanghee Choi Eungsu Yuk Jaecheol Kim, Jaeyoung Choi. Traffic characteristics of a massively multi-player online role playing game. Technical report, Seoul National University, Korea. NCsoft Corp., 2005.
- Mark Claypool James Kinicki. Traffic analysis of avatars in second life. Technical report, University of Michigan, Worcester Polytechnic Institute - USA, 2008.
- Mark Claypool James Nichols. The effects of latency on online madden nflr football. Technical report, Worcester Polytechnic Institute, USA, 2004.
- Daniel Zappala Jared Jardine. A hybrid architecture for massively multiplayer online games. Technical report, Brigham Young University, 2008.
- Chun-Ying Huang Chin-Laung Lei Kuan-Ta Chen, Polly Huang. Game traffic analysis: An mmorpg perspective. Technical report, National Taiwan University, 2005.
- Second Life. Second life official site, 2009. Disponível em <http://secondlife.com/>, acessado a última vez em Junho de 2009.
- Inc. Linden Research. The second life economy - first quarter 2009 in detail - features - second life blogs, 2009. Disponível em <https://blogs.secondlife.com/community/features/blog/2009/04/16/the-second-life-economy--first-quarter-2009-in-detail>, acessado a última vez em Junho de 2009.
- Seth Borg-Mark Claypool Emmanuel Agu Nathan Sheldon, Eric Girard. The effect of latency on user performance in warcraft iii. Technical report, Worcester Polytechnic Institute, USA, 2003.
- NCSoft. Lineage ii: The chaotic throne - top fantasy mmorpg by ncsoft, 2009. Disponível em <http://www.lineage2.com>, acessado a última vez em Fevereiro de 2009.
- Josilene Moreira-Paulo Cunha Carlos Kamienski Djamel Sadok Rafael Antonello, Stênio Fernandes. Traffic analysis and synthetic models of second life. Technical report, Federal University of Pernambuco, Federal Center for Education in Technology, Federal University of the ABC - Brazil, 2008.
- Josilene Moreira-Djamel Sadok Carlos Kamienski Stênio Fernandes, Rafael Antonello. Traffic analysis beyond this world: the case of second life. Technical report, Centro Federal de Educação Tecnológica de Alagoas, Universidade Federal de Pernambuco, Universidade Federal do ABC - Brazil, 2007.
- Grenville Armitage Tanja Lang, Philip Branch. A synthetic traffic model for quake3. Technical report, Swinburne University of Technology, Melbourne, Australia, 2004.
- Inc. The Croquet Consortium. Croquet consortium, 2009. Disponível em <http://www.opencroquet.org/>, acessado a última vez em Junho de 2009.
- Corey Lusher-John Plunkett Emmanuel Agu Mark Claypool Tom Beigbeder, Rory Coughlan. The effects of loss and latency on user performance in unreal tournament 2003. Technical report, Worcester Polytechnic Institute, USA, 2004.
- Duke University et al. Cobalt, 2009. Disponível em <http://opencobalt.org/>, acessado a última vez em Junho de 2009.

- Wikipédia. Awk - wikipédia, 2009. Disponível em <http://pt.wikipedia.org/wiki/AWK>,
acedido a última vez em Junho de 2009.
- Wikipedia. Internet, 2009. Disponível em <http://pt.wikipedia.org/wiki/Internet>,
acedido a última vez em Fevereiro de 2009.
- Wu-chi Feng Jonathan Walpole Wu-chang Feng, Francis Chang. A traffic characterization of
popular on-line games. Technical report, Portland State University, 2005.