UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

Ingeniería Técnica de Telecomunicación, especialidad Telemática

Trabajo de Fin de Carrera

Online Gaming Traffic Evaluation through Developing an Online Multiplayer Game

Autor: David Vela Tirado

Proyecto realizado en la Universidad de: Portsmouth
País: INGLATERRA
Profesor Tutor: Dr. Salem Aljareh

FIRMAS:

Secretario de la Escuela: D. Francisco Álvarez García

Coordinador Internacional: D. Antonio Guerrero Baquero

Tutor Académico: D. Enrique de la Hoz de la Hoz

CALIFICACIÓN: .....

FECHA: .....

Name of student	**David Vela Tirado**
University reference	634033
Date of birth	24 August 1990
HESA reference	1110746340339

Registered on	**C2396F Exchange Programme Eng**
Awarding Institution	University of Portsmouth
Teaching Institution	University of Portsmouth

Record of Learning and Achievement

2011/12	**Level 1**	**Mark**	**Credit**	**Decision**
U18352	English for Academic Purposes 1 (Credit)	64	10	Pass

2011/12	**Level 3**	**Mark**	**Credit**	**Decision**
P09165	Individual Project	80	40	Pass
P15888	Network Security	78	10	Pass
U16001	Open Distributed Systems 10	53	10	Pass

2011/12	**Level M**	**Mark**	**Credit**	**Decision**
P10191B	Client Server Network Architecture	54	15	Pass

Units Taken for Interest Only

2011/12		**Mark**	**Credit**	**Decision**
U18355	English for Academic Purposes 2 (Non-Credit)	N.A.	0	
U17329	Portuguese Language Grade 1A	68	10	Pass
U17330	Portuguese Language Grade 1B	N.A.	10	

Total Credits Gained	**85**
Total Credits Gained (ECTS)	**43**

Date Transcript Printed: *4 July 2012*

See overleaf for notes
To check the validity of this transcript you may contact the University of Portsmouth by any of the following methods:

By Post: School of Engineering
Anglesea Building
Anglesea Road

Portsmouth
Hampshire
PO1 3DJ

By Telephone: 023 92846451

By E-mail: eng-office@port.ac.uk

END OF TRANSCRIPT

* Denotes Pass/Fail Unit

RESUMEN

El estudio sobre el tráfico de los juegos online ha estado en gran medida olvidado; sin embargo, debido a la reciente popularidad de los juegos online, las publicaciones sobre dichos juegos han aumentado considerablemente. Este proyecto trata de abordar el análisis del tráfico de los juegos online de una manera diferente a la seguida hasta ahora. En primer lugar, una investigación sobre los juegos online es completada para el posterior desarrollo de un juego online y análisis de su tráfico de red. El resultado esperado es una lista de recomendaciones que optimicen el desarrollo de los juegos online.

RESUMEN EXTENDIDO

La industria de los juegos se está convirtiendo en una industria tan influyente como la industria de la música o el cine. Sin embargo, la calidad de servicio (QoS) de los juegos online es una dura y continua tarea, la cual tiene que ser tomada en consideración para desarrollar juegos online. Para mantener y respaldar la calidad de servicio de los juegos a través del continuo crecimiento de su tráfico en Internet y la limitación de su ancho de banda, este documento esta centrado en el estudio de los mayores problemas encontrados por los desarrolladores y diseñadores de juegos online durante el proceso de creación del juego o en el cumplimiento de los requisitos de conexión por parte de los usuarios.

El enfoque escogido para la satisfacción de este proyecto esta dividido en tres fases que pueden ser clasificadas en: Investigación (Sección 2: Background); Desarrollo (Sección 3: Methodology) y Análisis (Sección 4: Game traffic Analysis).

En primer lugar, una investigación sobre la progresión de las técnicas para limitar el tráfico de red y la motivación de los desarrolladores de juegos es realizada. A través de esta investigación es posible observar que existen un gran número de clasificaciones para agrupar los distintos tipos de juegos, los cuales tienen diferentes requisitos de red en función de sus características como pueden ser el número de jugadores, el modo de juego, la versión del juego, o la configuración del juego. Además la motivación de los desarrolladores de juegos y los efectos de la tecnología de banda ancha son estudiados con el fin de entender la progresión de técnicas para limitar el tráfico de red en los juegos online. Estas técnicas se han incrementado con el paso del tiempo haciéndose cada vez más complejas. Sin embargo, la mayoría de las publicaciones suelen coincidir en un diseño sencillo y la exclusiva transmisión de la información esencial como factores clave para la reducción del ancho de banda.

A la vista de los resultados obtenidos en la exploración de las propiedades de los juegos online, dicha investigación es finalizada con una lista de características que pueden influir en el tráfico de juegos online:

- Conexión de red
    - Protocolos: TCP/UDP
    - Tecnología de banda ancha: Ethernet/ Inalámbrica/ Redes móviles
- Número de jugadores
- Arquitectura de red: Cliente/Servidor, P2P, Hibrida.
- Efectos Gráficos y Sonoros.

Más adelante es incluida también el tipo de red (LAN /WAN) como factor fundamental en la calidad de servicio ofrecida a los jugadores.

Tras la clasificación de las características que pueden afectar al tráfico de red, la fase de desarrollo es comenzada. Antes de empezar con el desarrollo del juego online, un diseño preliminar es realizado con el fin de elegir el juego a desarrollar y las herramientas empleadas.

En primer lugar el juego a desarrollar es un juego arcade debido a que este tipo de juegos son los que tienen los requisitos de conexión más elevados por su característica de conexión en tiempo real. La duración del proyecto es limitada por lo que otra característica básica para el desarrollo del juego es la simplicidad. Con dichas características en mente, los primeros juegos en la historia fueron examinados en busca de posibles opciones de diseño y así fue como surgió la idea de rescatar el "Pong" creado por Nolan Bushnell en 1972. Con el juego escogido el siguiente paso es la elección del lenguaje de programación y el sniffer para desarrollar el juego y capturar su tráfico de red. El lenguaje escogido es Java y el Sniffer Wireshark. Esta decisión fue tomada a través de una exhaustiva comparación considerando la simplicidad, portabilidad, rendimiento y open source de las alternativas como características esenciales para la elección de las herramientas.

Cuando el análisis preliminar es concluido, la evaluación del hardware y el software es examinado y el aprendizaje del lenguaje de programación escogido es estudiado para la elaboración de un primer acercamiento en el desarrollo del juego. El cual es implementado con sockets para un mayor estudio de las cualidades de la red a diferencia de los applets de java que limitan el estudio al tráfico transmitido por el navegador.

El juego desarrollado tiene como objetivo evaluar el mayor número de características identificadas por lo que un portal es implementado para favorecer la coordinación entre jugadores y permitir varios jugadores jugar simultáneamente. Asimismo otro aspecto visible para los jugadores son los distintos modos de juego que tienen como fin evaluar los efectos gráficos y sonoros del juego en red. Por otra parte, el mismo juego es implementado tres veces usando distintos protocolos y arquitectura de red:

- Versión 1: protocolo TCP; Cliente – Servidor.
- Versión 2: protocolo UDP; Cliente – Servidor.
- Versión 3: protocolo UDP; P2P.

Cuando las tres versiones del juego son desarrolladas y probadas, la fase de Análisis es comenzada en la cual el tráfico de las tres versiones será capturado y analizado dando como resultado una lista de recomendaciones que tienen como propósito optimizar y asistir a futuros desarrolladores / diseñadores de juegos en el proceso de creación de juegos online multijugador.

Previamente a la captura del tráfico del juego, dos perspectivas son planificadas para evaluar las diferentes materias vistas a lo largo del proyecto y que pueden condicionar la calidad de servicio de los usuarios y el tráfico transmitido por el juego. Los dos enfoques son los siguientes:

- Análisis técnico.
- Análisis de Calidad de Servicio.

El análisis de la calidad de servicio es realizado a través de una encuesta que es completada por los jugadores la cual permite estudiar la impresión y preferencias de los jugadores con respecto

al juego desarrollado. Los resultados de esta encuesta coinciden con los resultados obtenidos en el análisis técnico y por lo general los jugadores están satisfechos con los colores y estilo elegido en el juego. Sin embargo es curioso destacar el hecho de que aunque las compañías actualmente dediquen gran parte de su presupuesto en gráficos, la encuesta rellenada por los jugadores demuestra que los jugadores prefieren un juego divertido y original a un juego con los últimos efectos gráficos del mercado.

El análisis técnico comienza con una clasificación de las medidas que serán abordadas durante el análisis del tráfico del juego. Estas medidas son en primer lugar:

- El ancho de banda [Kbps]
- Throughput [Bytes/s]
- Tasa de Paquetes [PPS]
- Retraso [ms]

Adicionalmente, también es necesario tener en cuenta las siguientes medidas para evaluar y observar posibles patrones en el tráfico del juego.

- Número de paquetes [Numero]
- Perdida de paquetes [Porcentaje]
- Longitud de los paquetes [bytes]
- IAT (tiempo entre envió/recepción de paquetes) [ms]

El cálculo y la evaluación de estas medidas se realizaran mayormente a través de las herramientas proporcionadas por el sniffer Wireshark. No obstante, las herramientas de Wireshark a veces no son suficientemente precisas; por lo que para complementar los resultados obtenidos con Wireshark dos programas proporcionados por el Dr. Savage de la universidad de Portsmouth son usados para representar estadísticas del tiempo transcurrido entre envió/recepción paquetes y tamaño de los paquetes.

Para la evaluación de las anteriores medidas son programados cuatro experimentos constituidos por una o varias pruebas. La siguiente tabla resume los experimentos mencionados:

EXPERIMENTOS	PRUEBAS	Cuestión evaluada
Experimento 1	**Prueba 1**	Protocolos TCP & UDP
Experimento 2	**Pruebas 2 y 3**	Características del juego
Experimento 3	**Pruebas 4 y 5**	Arquitectura de red
Experimento 4	**Prueba 6**	Tipo de redes

El primer experimento evalúa los protocolos TCP y UDP. Los cuales a través de un análisis en profundidad es demostrado que este parámetro condiciona enormemente la calidad de la comunicación entre jugadores. El protocolo TCP, por una parte resulta mucho más fiable; sin embargo también es más lento debido a dicha fiabilidad y esta excesivamente limitado por

tiempo RTT (Round Trip Time) de los paquetes. Por lo que los dispositivos más rápidos reciben mayor número de paquetes creando una desigualdad entre clientes y discriminando una parte de los jugadores. Por otro lado, el protocolo UDP muestra una gran superioridad frente al protocolo UDP ya que es independiente de la red y por lo tanto los paquetes son enviados uniformemente y equitativamente a ambos clientes favoreciendo el cumplimiento del requisito de comunicación en tiempo real exigido por este tipo de juegos.

Por otra parte, la diferencia entre protocolos no es el único parámetro que condiciona la calidad de la comunicación, la diferencia entre tecnologías y redes es una posible fuente de retrasos en los paquetes transmitidos. El último experimento demuestra que el tipo de red en conjunto con el protocolo escogido puede suponer la diferencia entre una aceptable calidad de servicio o una demora excesiva en la comunicación. El protocolo UDP funciona adecuadamente en las redes LAN y WAN. Sin embargo para que el protocolo TCP tenga la misma calidad de servicio el juego debe estar funcionando en red local. Además el rendimiento de este protocolo no es adecuado para juegos arcade donde la comunicación entre clientes debe ser lo más ligera posible sin importar la perdida de paquetes.

El segundo experimento trata de evaluar las características de los juegos a través de la prueba número dos (Efectos gráficos/sonoros y la velocidad del juego) y la prueba número tres (número de jugadores simultáneos). Mientras que los efectos gráficos y sonoros no suponen una gran variación en el rendimiento del juego, la velocidad de la bola del juego y el número de jugadores elevan considerablemente el número de paquetes enviados.

Anteriormente fue comentada la gran dependencia que existe entre el protocolo TCP y los parámetros de la red. Esta dependencia limita el número de paquetes transmitidos, debido a que hasta que el servidor no recibe el ACK del cliente no transmite el siguiente paquete. Por lo que la demora entre paquetes es mayor al igual que el tamaño de los paquetes, los cuales viajan cargados con información retrasada que no es útil para los clientes. Además comparando los dos protocolos, el protocolo TCP consume mucho más ancho de banda que el protocolo UDP debido al tráfico de subida de los clientes (paquetes ACK).

El tercer experimento en contraste con el resto de experimentos trata de evaluar el rendimiento de la arquitectura P2P, la cual demuestra propiedades muy deseables para los juegos arcade como un alto rendimiento (protocolo UDP) y un bajo ancho de banda consumido. Sin embargo, esta arquitectura también tiene ciertos problemas. Uno delos más graves es la posibilidad de engaño por parte de los usuarios, debido a que es el cliente el que genera la comunicación y no el servidor. Además, la comunicación entre clientes debe ser posible y este requerimiento muchas veces no es admisible debido a ciertos cortafuegos o diferencias entre redes.

Como conclusión no es posible establecer unas recomendaciones estándar para el desarrollo de juegos online ya que cada juego dispone de sus propias necesidades pero de forma general. Sin embargo, en vista de los resultados obtenidos es posible argumentar que a no ser que los datos transmitidos sean absolutamente imprescindibles es aconsejable la utilización del protocolo UDP. En cuanto a la arquitectura, la típica es Cliente – Servidor, pero P2P puede ofrecer ciertas

ventajas en las redes locales. Sin embargo, la seguridad de esta arquitectura es un aspecto delicado. En referente a las características de los juegos, los efectos gráficos y sonoros no suponen una gran carga en la red y rendimiento de los juegos. No obstante, para la administración de la velocidad del juego y número de jugadores simultáneos es necesario tener un completo control de la situación de la red. También es posible argumentar que la tecnología Ethernet suele dar más prestaciones que las redes inalámbricas. Sin embargo las diferencias son mínimas y las redes inalámbricas permiten más movilidad. El análisis de la calidad del servicio completado con las opiniones de los usuarios da como resultado principal la clara preferencia de los jugadores por la originalidad y lógica del juego. Por lo que, pese a que es necesario cierto interés por las necesidades de la red en los juegos online también es importante cierta pasión por los videojuegos y originalidad para que el juego desarrollado tenga éxito.

Por último este proyecto recoge en los apéndices información adicional que respalda los datos incluidos a lo largo del proyecto y ayuda a comprender el desarrollo y entendimiento del mismo. Además, la sección 7 (Future work) recoge varias formas posible para continuar este proyecto a través de la mejora del juego, análisis de las tecnologías móviles u otro tipo e arquitecturas como pueden ser el almacenamiento de información en la nube (Cloud computing).

Abstract

Gaming industry is becoming as influential and important as music and movie industries. However, the quality of service on online games is a hard and continuous task, which has to be taken into consideration for developing online games. This document attempts to solve some of the main concerns which developers and designers of online games have to deal in the creation process of a game. The approach to satisfy the requirements of this project consists of three parts. Firstly, a study is completed about the progression of the techniques for limiting network traffic and the motivations of the game developers. This investigation is concluded with an overview of the most important game features which has an effect on the network traffic. Secondly, after the research, a simple online-multiplayer game is developed using the more valuable tools in order to evaluate game features identified. Finally, the project concludes with a comparative analysis of the results obtained from capturing gaming traffic of the different game versions deployed. This analysis is concluded with a list of recommendations that may help to future game developers in the implementation and design of online games in order to meet the network requirements accurately.

Keywords

Gaming traffic evaluation, development of online multiplayer games, measurement, techniques to limit network traffic, game features.

ACKNOWLEDGMENT

I would like to thanks everybody who read this report for use your time in this piece of work; in particular I would like to thanks the people who always were with me, both the people who are with me here in Portsmouth and the people who are more than 700miles away from me. This project would not be possible without their support. Concretely, I want to thanks my family for encouraging me to continue studying; without them I would not have been able to achieve any of my goals.

To finish and not less important, I'd like to thanks all my new and old friends who always believed in me. Specially, I would like to highlight Johanna who without her peculiar way of supporting me, this residence in England would not have been the same, and Julio who did not doubt for a second in helping me with the experiments in Spain.

Table of Contents

RESUMEN i
RESUMEN EXTENDIDO ii
Abstract vii
Keywords vii
ACKNOWLEDGMENT viii
Table of Contents ix
Nomenclature xi
Document Format xii
List of figures xiii
List of Tables xv
Screenshots xvi
1. INTRODUCTION 1
1.1. Objectives 3
1.2. Results 4
1.3. Report Overview 5
2. BACKGROUND 7
2.1. Classification generic of Online Multiplayer Games 7
2.2. Progression of the Game Publications 9
2.3. Broadband Technology 13
2.4. Techniques for limiting Network Traffic 15
2.4.1. Basic Data Reduction 16
2.4.2. Time Delay and Time warp 16
2.4.3. Compression techniques 16
2.4.4. Prediction 17
2.4.5. Other methods 18
2.5. Game Features which Affect to the Network Traffic 21
3. METHODOLOGY 24
3.1. Preliminary Design 24
3.1.1. Game Selection 25
3.1.2. Tools Selection 27
3.2. Development Process 35
3.2.1. Available Resources 35

3.3. Initiation into Java 41
3.4. First approach in the development of the game 45
3.5. Expansion Process 56
3.5.1. UDP Connection 56
3.5.2. Sound Effects 57
3.5.3. Graphics Effects 58
3.5.4. Others - P2P architecture - 60
4. GAME TRAFFIC ANALYSIS 64
4.1. Introduction 64
4.2. Methodology of the Game Traffic Analysis 65
4.3. Technical Analysis 70
4.3.1. First Experiment 71
4.3.2. Second Experiment 87
4.3.3. Third Experiment 99
4.3.4. Fourth Experiment 107
4.4. Quality of Service Analysis 117
5. CONCLUSIONS 119
6. FUTURE WORK 124
Bibliography 127
Appendix A – APDF form 132
Appendix B – Statistical and forecast for 2010 – 2015 from Cisco 133
Appendix C – Classification Generic (by Game Style) 135
Appendix D – Acquisition of the specifications 136
Appendix E – Planning: Project initiation and planning tables (Gantt charts) 138
Appendix F – Code: First approach 144
Appendix G – UML Diagrams 147
Appendix H – Survey to evaluate the preferences of the users 150
Appendix I – Wireshark General Information 151
Appendix J – Network Abstraction 155
Appendix K – Presentation 156
Appendix L – CD content 158

Nomenclature

AI	Artificial Intelligence
API	Application Programming Interface
Bytes/s	Bytes per second
C/S	Client – Server (Network Architecture)
FPS	First Person Shooters
GUI	Graphic User Interface
HW or H/W	Hardware
I / O	Input / Output
IDE	Integrated Development Environment
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
JVM	Java Virtual Machine
KB/s	Kilobytes per second
Kbps	Kilobits per second
M2M	Machine to Machine
MB/s	Megabytes per second
Mb/s or	Megabits per second
MG	Multiplayer Games
MMOG	Massively Multiplayer Online Game
MMOFPS	Massively Multiplayer Online First Person Shooter
MMORTS	Massively Multiplayer Online Real Time Strategy
MMORPG	Massively Multiplayer Online Role Playing Game
ms	milliseconds
MUD	Multi User Dungeon Games
NS	Network Simulator
OOP	Object Oriented Programming
OS	Operating System
P2P	Peer to Peer
PC	Personal Computer
QoS	Quality of Service
RTS	Real Time Strategy games
RPG	Role Playing Game
s	seconds
SW or S/W	Software
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UML	Unified Modelling Language
VoIP	Voice over IP
VPN	Virtual Private Network
WoW	World of Warcraft
WAN	Wide Area Network

Document Format

This document adheres to certain typographical rules in order to maintain consistency throughout the whole report. Therefore, the following formats are described to make the document more readable.

Type	Font	Size	Weight	Information
Body	Times New Roman	12	Normal	Any text which is not a header
Code	Helvetica	10	Normal	All code or any pseudo-code
Heading one	Times New Roman	14	Bold	Top-level heading indexed with a number e.g. **1 Primary**
Heading two	Times New Roman	12	Bold	Second-level heading indexed with the number of the previous headings and a new number. e.g. **1.1 Secondary**
Heading three	Times New Roman	12	Normal	Third-level heading indexed with a roman number in lower case: e.g. 1.1.1 Subheading
Subtitle	Times New Roman	12	Italics	Any sublevel under the Heading three. e.g. *Subtitle*

This document was created in Microsoft Word 2010 with the compatibility Mode activated.

List of figures

Figure 1: Access network 13
Figure 2: Time Delay on the Server. 16
Figure 3: Arcade Pong Game Machine. 25
Figure 4: Capture from the Arcade Pong Game Machine 26
Figure 5: Scheme of the game 26
Figure 6: Sniffer Selection. Cain & Abel screenshot 31
Figure 7: Sniffer Selection. Tcpdump man page 32
Figure 8: Sniffer Selection. Wireshark screenshot. 33
Figure 9: Sniffer Selection. Wireshark logo 34
Figure 10: First approach 46
Figure 11: First approach. Control packets 48
Figure 12: First approach. Game process 49
Figure 13: Utility timer ball (Davison, 2005, p.40) 54
Figure 14: Improvements P2P. Game Process. 61
Figure 15: The Performance of a Fast Ethernet Connection under Best-Effort Conditions. . 68
Figure 16: Experiments summary. 70
Figure 17: Test 1. UDP server. Packet rate [PPS] 74
Figure 18: Test 1. Server TCP. Packet rate [PPS] 74
Figure 19: Test 1. TCP Server. Flow Graph 76
Figure 20: Test 1. UDP Server. Flow Graph 77
Figure 21: Test 1. UDP version. Packet Lengths 79
Figure 22: Test 1. TCP version. Packet Lengths 79
Figure 23: Test 1. TCP server. The largest packet 80
Figure 24: Test 1. TCP Server. Throughput 82
Figure 25: Test 1. TCP Server. RTT. 82
Figure 26: Test 1. Packet Loss rate in the clients 83
Figure 27: Test 1. UDP version. IAT 85
Figure 28: Test 1. TCP version. IAT 85
Figure 29: Test 2. Server TCP. Packet rate [PPS] 91
Figure 30: Test 2. Server UDP. Packet rate [PPS] 91
Figure 31: Test 2. Laptop UDP. Packet rate [PPS] 91
Figure 32: Test 2. Chipmunk UDP. Packet rate [PPS] 91
Figure 33: Test 2. Chipmunk TCP. Packet rate [PPS] 91
Figure 34: Test 2. Laptop TCP. Packet rate [PPS] 91
Figure 35: Test 3. Server TCP. Packet rate [PPS] 92
Figure 36: Test 3. Server UDP. Packet rate [PPS] 92
Figure 37: Test 2. Output Laptop RTT. G 1 (simplex) 93
Figure 38: Test 2. Output Laptop RTT. G 2 (water) 93
Figure 39: Test 2. Output traffic server. Game 3 94
Figure 40: Test 2. Packet Lengths. Game 3 TCP. 95
Figure 41: Test 2. Packet loss rate. 96
Figure 42: Test 3. Evaluation of packet loss rate 97

Figure 43: Test 4. Server P2P. Packet rate [PPS] 103
Figure 44: Test 4. Chipmunk P2P. Packet rate [PPS] 103
Figure 45: Test 4. Deer P2P. Packet rate [PPS] 103
Figure 46: Test 5. Packet rate [PPS] 104
Figure 47: Test 4 and 5. Packet loss rate. G1 & G2 106
Figure 48: Test 6. Madrid Server TCP. Packet rate [PPS] 110
Figure 49: Test 6. Laptop TCP. Packet rate [PPS] 110
Figure 50: Test 6. Madrid_laptopTCP . Packet rate [PPS] 110
Figure 51: Test 6. Madrid Server UDP. Packet rate [PPS] 111
Figure 52: Test 6. Madrid_Laptop UDP. Packet rate [PPS] 111
Figure 53: Test 6. Laptop UDP. Packet rate [PPS] 111
Figure 54: Test 6.1. Madrid_Server TCP. Packet rate [PPS] 112
Figure 55: Test 6.1. Tenerife_Computer TCP. Packet rate [PPS] 112
Figure 56: Test 6.TCP flow 113
Figure 57: Test 6.1. TCP flow 113
Figure 58: Test 6. TCP Laptop. RTT 114
Figure 59: Test 6. TCP Madrid_Laptop. RTT 114
Figure 60: Test 6.1. Tenerife Computer. RTT 114
Figure 61:Test 6. Packet loss rate. Global 115

List of Tables

Table 1: Measures for Captured Packet. 10
Table 2: Measures for Access Technologies. 14
Table 3: Language of Programming. 28
Table 4: Sniffer comparison 34
Table 5: Resources HW. Personal Laptop 36
Table 6: Resources HW. Computer of the room A2.3 37
Table 7: Resources HW. Server 38
Table 8: Resources HW. Network 39
Table 9: Resources SW 40
Table 10: Game versions 62
Table 11: Test 1. Resume. 71
Table 12: Test 1. Network Trace Information 72
Table 13: Test 1. Trace Network Statistics. 73
Table 14: Test 1. Server Protocol Hierarchy Statistics 75
Table 15: Test 1. Total packet Lengths in the servers 78
Table 16: Test 1. Statistics of the Packet Length in the servers 78
Table 17: Test 1. Evaluation of the packets sent and received. 83
Table 18: Test 1. IAT Statistics 84
Table 19: Test 2/3. Resume 87
Table 20: Test 2. Network Trace Information 88
Table 21: Test 3. Network Trace Information 88
Table 22: Test 2. Statistics of the Packet Size. Game 3 TCP 95
Table 23: Test 2. Evaluation of packet loss rate 96
Table 24: Test 3. Evaluation of packet loss rate 97
Table 25: Summary Servers. Test 2_ G3; Test 3 98
Table 26: Test 4. Resume 99
Table 27: Test 5. Resume 100
Table 28. Test 4. Network Trace Information 100
Table 29: Test 5. Network Trace Information 101
Table 30: Test 4. Trace Network Statistics. 102
Table 31: Test 4 and 5. Packet loss rate. G1 105
Table 32: Test 4 and 5. Packet loss rate. G2 105
Table 33: Test 4 and 5. Summary servers G1 & G2 106
Table 34: Test 6. Resume 107
Table 35: Test 6. Network Trace Information 108
Table 36: Test 6. Packet loss rate. Global 115
Table 37: Test 6. Summary Servers. G1 & G2 116

Screenshots

Screenshot 1: Graphic design. Initial…….…....….....68
Screenshot 2: Graphic design. Football……...….……..…68
Screenshot 3: Graphic design. Water…...……..……..…….69
Screenshot 4: Graphic design. Game Portal…….……...……..69

This page intentionally left blank

1. INTRODUCTION

Online gaming traffic represents an important percentage in the Internet traffic statistics at the present, and will subsequently proliferate in the next few years. The Cisco Visual Networking Index Forecast Methodology estimates and forecasts that not only the gaming traffic will increase, but also that the global IP traffic, which has augmented over the past five years, will continue increasing fourfold over the next five years. Non-PC devices (TV's, tablets, smartphones, and machine to machine (M2M) modules) which are increasingly relevant to the Internet traffic and most of these devices can store online games are in the same situation (Cisco and/or its affiliates, 2011), without mentioning the game consoles as Xbox or PlayStation 3, which are generating a significant portion of the gaming traffic at the moment (Onlinemba, 2011). (The tables which corroborate the above data and the following are included in the appendix B)

This report focuses on the gaming traffic generated by computers on online multiplayer games. These games use a high quantity of bandwidth; with the expansion on Internet traffic, which was seen above as a big industry, the game industry must ensure and maintain the Quality of Service (QoS) of games at all times. Therefore, a good administration of game bandwidth and network resources such as the latency, jitter and loss of connexion is vital. Several companies such as Plusnet offer high QoS on gaming traffic through the prioritizing of client traffic based on the users' preferences (Plusnet, 2012).

However client QoS is not the only challenge related with the bandwidth or latency. Although bandwidth in the development of online games is a subtler constraint, game features and development are affected (Hall & Novak, 2008, p. 42), because the bandwidth available between the client and server is limited (Russell, Donaldson & Sheepard, 2008). Latency is believed to be very closely related to the bandwidth. Therefore solving the bandwidth issues may solve latency issues (Armitage, Claypool & Branch, 2006, p.96). Furthermore, all these issues depend on the type of game, different games have different needs. For instance, if the game has many interactions between players, the latency must be low (Lecky-Thompson, 2008, p.91).

It is essential to enhance the importance of videogames in the worldwide industry. However, the most impressive data is the duration of this achievement, games have changed from a simple hobby and a little niche in the market to an industry which generates billions of dollars per year in just twenty five years, and the present situation seems not slowing this overgrowing (Hall & Novak, 2008, p. 6). Nae, Prodan, Iosup & Fahringer, (2011) present a brief overview of the companies most impressive on the market where it is possible to find ESA (Entertainment Software Association) which estimated seven billion US dollars (USD) with an avid growth over 300% in the last 10 years. In contrast, the MPAA (Motion Picture Association of America) reports a size of 8.99 billion USD and RIAA (Recording Industry Association of America) with a size of

12.3 billion USD. However, this sudden upsurge is not only in America, the Chinese market also presents predictions of billions USD (Lin, Cao & Yin, 2010). Therefore, this data seems to prognosticate that the gaming industry will be soon larger than both movie and music market sizes (Nae et al.).

Despite all the benefits shown, the bandwidth is considered an important cost on this industry; because Internet service providers' (ISP) charge is based on the highest data rate achieved during the billing period. (Hall & Novak, 2008, p. 213).Therefore, if the number of users playing games is very high, the traffic transmitted in the same time could also be very high and reduce the capital allocated for other purposes, such as maintain the graphics and game effects according to the market necessities; a cost sometimes overvalued. (Hall & Novak, 2008, p. 39)

As a consequence, the preservation of QoS users, the limitation of network resources and the cost make necessary to measure, understand, and analyse network traffic. In large networks, the previous purposes are still a challenge for achieving and are a vital task for network management and diagnosis of issues, according to Zhang, Wang, Shi, Lian & Feng, (2012, p. 4). In other words, games are evaluated in order to achieve the management and control of the network traffic; and therefore success on the market. (Lecky-Thompson, 2008, p. 158)

This project will provide other points of view regarding to online multiplayer games evaluation. It is focused on the network requirements, researching, and analysing the potential tools, techniques and technologies which allow developing a game complying with the network specifications. To achieve this goal, a study on the motivation and the concerns game developers and designs is performed. The study is deepened at the end in the main aspects which affect the network traffic, such as access technology –broadband–, measures of the network traffic and gaming features. As the requirements of each game are different, it is helpful to understand the characterization of the kinds of games according to their purposes and necessities. Hence, this research must choose the adequate game to evaluate the network widely, subsequently to the proper identification of the kinds of games.

Finally, a simple online multiplayer game belonged to the genre selected will be developed in order to test the features in the network traffic generated by several prototypes of the developed software.

1.1. Objectives

The main objective is the evaluation of the Network Traffic generated by the games on the Internet and its possible effects on the Network. To achieve this objective with success, it is necessary to obtain a list of recommendations for an efficient development of online games, to take into consideration the network and to find a balance between network and game quality.

For a proper conduction of the project, it is essential to adopt the following specified aims:

1. Research the progression of techniques used within games to limit network traffic of the game.

2. Investigate how broadband may aid online gaming.

3. Identify game features that may have an impact on network traffic.

4. Develop a simple multi-players game to experiment and evaluate the different effects of the identified game features.

5. Establish measures which allow an objectively way to compare the results obtained from applying several modifications to the game.

6. Draft a final report based on the results, recommendations and conclusions obtained throughout the project.

1.2. Results

The main result will be the accomplishment of a final report with the following contents:

- A list of the techniques used within games for limiting the network traffic.
- An analysis of how the broadband changed the world of the online games.
- A detailed description of the features which affect to the Network Traffic.
- A comparative analysis of the results obtained from the combinations of several features implemented in the software prototype (game) previously developed.
- A critical evaluation of the methodologies and techniques used in light of the results obtained, indicating which is or are the most adequate, and a proposal for improve it in each instance.

As mentioned above, a set of software prototypes of a simple multiplayer game will be developed in order to explore the game features on the Network. This software prototype will be accomplished during the evaluation process together with the next elements:

- Source code.
- Engineering documentation of the software associated to the development, including the detailed instructions for the running of the prototype.
- Executed programs and/or compiled elements ready for the installation.

This documentation will be added into the final report appendix.

1.3. Report Overview

This report is divided into six sections. This first section, the Introduction, is intended to give a general overview about the objectives, expected outcomes, and main motivations for the execution of this project. Furthermore, the implementation of this project involves the satisfaction of three processes: the research process, the development process, and finally the analysis process. These three phases are organized in the following three sections.

Section 2–Background: The aim of this chapter is to gain more insight and knowledge into the online-games area through the research of the previous publications and the progression of techniques used within games to limit network traffic. Moreover, this section also shows the broadband technology effects over gaming traffic and concerns of game developers to result in the identification of game features that may effect on gaming traffic.

Section 3–Methodology: Based on the game features identified at the end of the last section, the design and development of a game is performed through the election of the accuracy tools and evaluation of available resources. The last results of this section are the implementation of three versions of one game designed in order to test the previous game features.

Section 4–Game traffic analysis: The results of the last chapter is evaluated and analysed in this section through the accomplishment of two different analyses, one based on the technical parameters and organized in four experiments; and other focused on the final users' preferences.

Consequently, this report is finalized with the presentation of the 'conclusions' – section 5– obtained through the fulfilment of the previous sections and 'future work' –section 6–which can be accomplished in future projects.

This page intentionally left blank

2. BACKGROUND

Several studies which constitute the area of "Gaming Traffic Evaluation" had been published; this section of the report tries to collect some of the most important documents regarding this field. Focusing first on the progression of the studies over the time, it is possible to observe the motivations and concerns which the developers and designers of online games may find. This study deepens in the techniques used within games to limit the network traffic –bandwidth– of the game, and thus to deal with the bandwidth limitations efficiently. Furthermore, this section considers the latency problems because this issue is very closely related to the bandwidth problem (Armitage et al., 2006).

Before starting the progression of game development over time, it is necessary to make a brief overview of the different kinds of games which exist on the current market, because throughout this report appears these terms several times. The next classification is one of the many classifications which are in the world of the video games.

2.1. Classification generic of Online Multiplayer Games

The games can be classified according to purpose of the game and number of players. The four main genres according to the above criteria are:

- MUD (Multi User Dungeon Games)
- RPG (Role Playing Game)
- FPS (First Person Shooters
- RTS (Real Time Strategy games)

(Further information can be obtained in the Appendix C)

Apart from these four genres, there an extra one called MMOG (Massively Multiplayers Online Games), which involves all the above genres with an extension of the number of its users. The MMOG games will be the focus of this piece of writing as a result of the ideas from Rabin who demonstrates that "the Bandwidth cost increase linearly with the number of players" (Rabin, 2010, p. 629).

The big difference with other genres, as was mentioned previously, is the number of players who are playing simultaneously; it can be supposed when the word "Massively" appears at the start of the title. However, how many players a game has to support simultaneously to be considered a MMO game? This is a question which still is not resolved, and hence the first MMOG on the history (Hall & Novak, 2008, p. 6). However, it is not relevant to the approach of this report; the important data is the progressive increasing of the number of players over time, until one day, this number of

players was an amazing amount. Hence, the term "massively multiplayer online game" suddenly seemed appropriate. (Hall & Novak, 2008, p. 6)

MMOG games type include: World of Warcraft (MMORPG); Counter Strike (MMOFPS); Age of Empires (MMORTS). The identification of these three popular games is suitable due to the arguments of Färber (2002) who points out that three to four per cent of all packets are generated with only six popular games (Färber, 2002). This is the main reason of the numerous studious about these three games; highlight the first one, World of Warcraft is considered one of the MMO with larger number of players; furthermore this game has attracted, according to Wang et al (2012), more than ten million players around the world.

Within of the MMOG – the genre which usually has the highest number of users– is the MMORPG. The players of this kind of games are identified with virtual avatars, chosen and modified according the preferences of each user. In these games avatars can interact with the rest of players, speaking, chatting, and sharing experiences within a virtual world, full of fantastic and mysterious stories (Wang et al, 2012). One of the reasons of the high number of users may be the big addiction generated by MMORPG games between teenagers and some adults. As it was discussed above, the number of player affects directly to the network resources, as a consequence this report will focus on this kind of games due to their number of players and its repercussion on the network.

2.2. Progression of the Game Publications

The Evaluation of Gaming Traffic is currently very popular; it is possible to find evaluations of several acclaimed high number players games like World of Warcraft or Counter Strike. However, until some years ago questions related to the games' effect on the network has been largely ignored in both academic and industry publications. (LaPointe & Winslow, 2001, p. 11)

The study of LaPointe & Winslow describes the situation of the research ten years ago. The industry publications were more interested on setting milestones and deliverables correctly in order to meet the deadlines planned and include realistic effects and spectacular graphics into the game. These considerations are very important; however issues like developing a robust network layer or reduce network load are also necessary, but due to the limited budgets, were ignored. (LaPointe & Winslow, 2001, p. 11)

On the other hand, academic articles were focused on playing issues created by new network protocols; instead of evaluating the performance of the protocols used in the gaming traffic. This lack of interest could cause certain delay in the acquisition of knowledge regarding to traffic patterns, bandwidth usage, and gaming protocols. (LaPointe & Winslow, 2001, p. 11)

Nowadays, it is still possible to perceive some of the motivations of the past. The game industry is very competitive, and the best games are the usually first to arrive the market (Lin et al, 2010, p.1725). A possible reason to speed the deadlines is that the second game arriving to the market does not usually get the recognition deserved. The situation of gaming publications has increased over the time. However, the majority of the contents of these publications are related to the evaluation of the most popular games.

Some of the most relevant publications about the study on games are:

- "Characterizing the Gaming Traffic of World of Warcraft: From Game Scenarios to Network Access Technologies" by Wang et al (2012).

- "A Survey of First-Person Shooter Gaming Traffic on the Internet by Ratti, Hariri & Shirmohammadi (2010).

- "A Traffic Characterization of Popular On-Line Games" by Feng et al (2005).

These publications often coincide in the measures selected to analyses the gaming traffic; and in the identification of Traffic Patterns. However, the methodology and the conclusions of each document are very different. It could be due to the intervals of years and the difference of genres. For this reason, the numerical data are omitted in this section of the report.

Ratti et al (2010) state that the main measures to evaluate the gaming traffic are the measures represented in the following table, which express the characteristics along with their standard unit values.

Table 1: Measures for Captured Packet.
Retrieved from (Ratti et al, 2010)

Characteristic	Standard unit value
Packet interarrival time (IAT)	Milliseconds (ms)
Packet length	Bytes
Bandwidth/data rate	Kilobits per second (kbps)
Packet rate	Packets per second (PPS)

Although they have not the same name in the different publications, and they are calculated with a different procedure; finally all the publications deal the same issues of the gaming traffic: the latency and the bandwidth. Specifically, the identification of the next characteristics: "the packet size" and "packet IAT" is decisive to calculate the packet rates and the data rate; furthermore these characteristics have connection with other important parameters regarding the evaluation of the network (Ratti et al, 2010).

Last features are mainly to study and support the high bandwidth requirements of some games. However, it is also interesting to consider the performance of the network through the analysis of the latency and the packet loss in the gaming traffic (Wang et al 2012).

- Delay or real-time interaction. – Latency –
  Delay is defined as the time passed since a user send a request to the server and the server answers to the user. Games need this value to be as lower as possible, and with any variation in the arrival of packets (jitter) in order to have the appearance a real time games. Therefore, the calculation and control of this element, which can be calculated with the round-trip time (RTT), are vital tasks for the games' success. (Wang et al 2012).

- Packet loss.
  This parameter affects directly the gaming performance. For instance, whereas the game traffic is TCP and the packet is lost, the system has to resend the packet. This process consumes time; therefore if this fact is often repeated, the QoS may reduce significantly. (Wang, W et al 2012). Despite of this fact, it depends on the requirements of each game, and its network connection. Consequently, the knowledge of the game requirements is essential for the establishment of game measures.

Finally, each study is evaluated separately in order to conclude the analysis of these publications after the exposition of the most pertinent problems to the main academic

publications. Through this evaluation is possible to learn some techniques very useful for the future evaluation forecasted in this project. Continuing with the publication of Ratti et al. (2010), this study divides the process of Gaming Traffic evaluation into three basic phases: Data capture, Data scrubbing, and analysis and modeling.

The researchers capture larger and longer set of data or traces in the data capture phase. Although the size of the traces depends on the study, the acquisition of these packets can vary according to several parameters as:

- Game version: The new versions of games tend to bring patch which raise the performance; therefore the capture of small set of data may be enough for the analysis.

- Technical environment: It is referred to the use of a dedicated/nondedicated server and, server access technology. The capture traffic in nondedicated server is often more complicated due to the different traffic found.

- Game configuration: Several games allow configuring the characteristics that affect the network. Thus, it is necessary to know the configuration effects in order to evaluate the gaming of traffic.

- Match configuration: The gaming parameters can usually affect the traffic; for instance the game map and the number of players may increase considerably the packet number.

(Ratti et al., 2010)

The researchers must clean the traces captured from packets which are not relate with the game analysis after the capture of the game traces. Finally, the investigators are in conditions to investigate the game packets; this step is called "analysis and modeling". In this step the research extract, tabulate, plot, and analyze the traffic characteristic identified before and through simulations or comparison with other games, a conclusion and a summary are drawn in order to clarify the results obtained (Ratti et al., 2010).

The study of different access technologies (traditional wired links, wireless networks, and mobile networks) from different WoW gaming scenarios for player's interactions evaluation is the major innovation of the document of Wang et al (2012).

The results of this document are fairly interesting. Strengths and weakness of different WoW access technologies which use TCP connections are evaluated and compared among themselves. The three popular access technologies examined are: Ethernet, WiFi and WiMAX. Ethernet technology was the most stable with the highest bit rate. Nevertheless, wireless networks are very similar in performance, and offer the possibility of playing the game in different places. In contrast, the mobile technology is

the technology with more deficiencies, especially in the packet loss rate (Wang et al 2012). Hence, the UDP protocol may be a suitable option for games in mobile networks; as the UDP protocol does not have retransmission, and thus the latency is lower.

Finally, Feng et al (2005) states that the FPS games analyzed exhibit a predictable and periodic behavior over long periods of time. Hence, simulators of gaming traffic may be an effective tool for understanding the performance of the online games through testing and experimenting on several points of the network. However, this idea was also seen in the dissertation of LaPointe & Winslow (2001); in which the traffic from two of the most popular games at that moment (WoW and Counter strike) was analyzed in terms of bandwidth distribution between players, number and size of packet, and packet interarrival time. A game traffic simulator was subsequently developed depending on the results obtained. Some variations and differences are observed through the comparison of the simulator traffic with the real gaming traffic. Thus, these results may indicate that the development of game simulator had to be improved; a fact that has been executed in posterior studies.

It appears that the most severe issue found in these publications is the use of a style which it is known as "Postmortems". It means that the analysis is developed after the game development cycle. These analyses content the problems and motivations of game developers who are more interesting in the purpose and game effects on the final users, such as the graphics design; instead of focus on issues related with networks resources or tools used for game development (LaPointe & Winslow, 2001, p. 12). This approach may difficult the acquisition of information for technical studies about the techniques used in the game development; or information related with the design methodologies for the correct use of network resources during the gaming development.

In addition, most of post-mortems publications are often written by a non-technical producer; in other words somebody without technical knowledge about parameters or network elements. LaPointe & Winslow (2001, p. 12) argued that the majority of games developers choose their profession because of their love towards video games; and it is a good point to develop games. However, it is necessary something more, those specialists also have to be interested in technical aspect of the game development, such as bandwidth limitations of or delay issues to become successful in their profession.

As the cycle of development to deploy and launch games has been reduced, the majority of the researches are devoted to graphics, an area that changes radically every six or eight months with each new generation of video hardware (LaPointe & Winslow, 2001, p. 12). Hall & Novak (2008, p. 39) note that "No matter how cutting-edge your technology might be, it won't take long for the rest of industry to duplicate it". Therefore, this part of the research can be summarized as a lack of interest in the technical aspects of game development by game designers and current publications.

2.3. Broadband Technology

The evolution and increase in the gaming traffic studies could be triggered by the evolution of broadband technology; specifically the evolution of the broadband access network.The Section 706 of the Telecommunications Act of 1996, defines advanced telecommunications capability (i.e., broadband) as "high-speed, switched, broadband telecommunications capability that enables users to originate and receive high-quality voice, data, graphics, and video telecommunications using any technology." (Federal Communications Commission, 2010, p.3). In other words, it is the bit rate of the user connections. Without agreement on how many bit rates is broadband access (Armitage et al, 2006, p. 121), in the 2010 Sixth Broadband Deployment Report, the Commission took the step of raising the minimum speed threshold for broadband from services in "excess of 200 kilobits per second (kbps) in both directions" (Federal Communications Commission, 2010, p.3). However Armitage et al, (2006), refers to broadband at least 256kbps.

As it was observed in the publication of Wang et al (2012), it is possible to obtain a different range of values in network features depending on the access technology. The figure 1.0 shows in simplified form the connection between players and game server.

Retrieved from (Armitage et al, 2006, p.122)

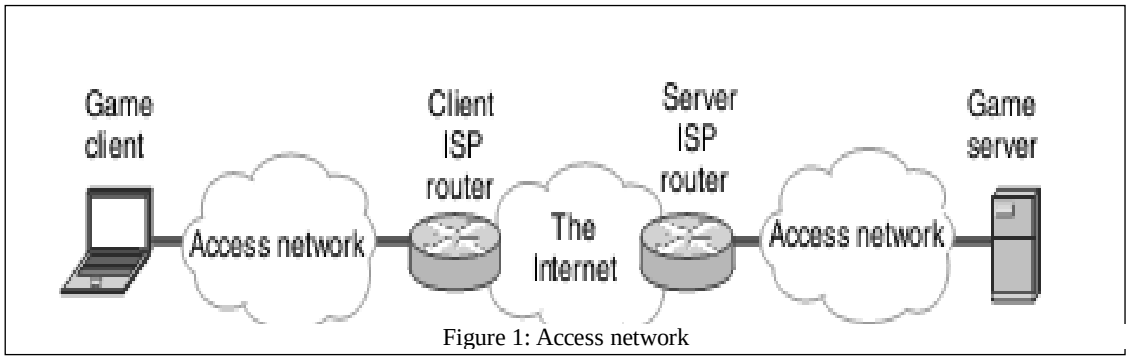


Figure 1: Access network

The concerns on the connection between server and the Internet are based on network conditions (i.e. congestion), which it is impossible to control them from the point of view of game developers. Therefore, this study is focused on the characteristics of broadband access network from game client side. A great diversity of broadband access technology can be found in the market, this situation allows increased the number of player, and therefore the benefits in the gaming industry. However it also means that it is necessary to understand the strengths and weaknesses of each technology (Armitage et al, 2006, p.136).

The use of the following features which are very similar to ones that help to measure the quality of the gaming traffic are usually necessary to make a classification about the qualities of the different broadband technology.

Table 2: Measures for Access Technologies.
Retrieved from (Armitage et al, 2006, pp.135 – 136)

Characteristic	Measured using:
Capacity	Bandwidth (kbps)
Delay	RTT (ms)
Delay variations (jitter)	IAT (ms)
Coverage	Frequency (Hz) (mobile networks)

Taking as example the data of the evaluation realised by Wang et al (2012) about the game WoW, it is possible to establish the following broadband access technologies:

- Ethernet or ADSL:
- Wireless LANs (WiFi):
- Cellular Networks: Generally is to have similar features
    - WiMAX
    - LTE

Regarding the fact that the study is executed about the game WoW, it is relevant to remember that this game is one of the most important games on the Internet which has millions of user and brings very high revenues. Therefore, the user's requirements are essentials to meet as it was reflected in the study of Wang et al (2012) in terms of Ethernet, WiFi or WiMAX connections. The only problem was found in the latency and the packet loss of WiMAX, which is a very recent technology; thus it could be argued that these negatives results are excusable. However, the resources of WoW both economic and materials are not available to everyone. The majority of online games have to usually satisfy the users with poor resources in terms of servers and bandwidth.

The access types used by the clients are necessary to analyse for developing and deployment games in order to solve issues like random variations in capacity and delay. Hence, game designers have to work on minimising the traffic transmitted as much as possible (Armitage et al, 2006, p.136). There are some considerations and techniques which may be helpful to limit network traffic and obtain the best performance of the used resources.

2.4. Techniques for limiting Network Traffic

Some of the techniques to mitigate issues of latency, jitter, packet loss and bandwidth, in order to achieve the success of a network will be presented. However, it is essential to know first what the sources of these problems are:

- The three main causes of delay or latency are:
  - Large distances on the network
  - Serialisation delays
  - Congestion on the network

- The mechanisms which introduce jitter are: the variations on path length, packet size and congestion of the network.

- The packet losses are often provoked by: overflow of the network or errors in the link.

(Armitage et al, 2006, p.70)

- High bandwidth is usually generated when there are too many users playing at the same time and when the frequency of retransmission of packets is very high. (Lecky-Thompson, 2008, pg. 179)

The majority of these causes are beyond the control of game developers, and despite the high rates of the broadband, the clients may still suffer high latencies during packets congestion periods (Armitage et al, 2006, p.83). The only way to solve this is to create mechanism or techniques able to reduce these problems connected to high latencies or packet loss.

Through the reduction in the size of the messages sent between client and server may decrease the latency due to the fact that a small packet has a shorter transmission time than a large packet (Armitage et al, 2006, p.96). Therefore, using network bandwidth efficiently becomes possible to reduce delay on the packets.

The "Principle of Minimum Data in Transit" is based in two rules to obtain the best performance on the development of games. These rules are:

- Data transmitted are just the essentials.
- Data are created in order to cover the maximum meaning.

(Lecky-Thompson, 2008, p. 179)

It is important to keep these rules in mind when the game is in the development process. However, sometimes the amount of necessary information is too high and cannot be reduced. Therefore the next techniques can be used for helping in that process.

2.4.1. Basic Data Reduction

The first obvious technique is to relay only the absolutely necessary data (Lecky-Thompson, 2008, p. 180). The method of "Basic Data Reduction not only selects the data that is absolutely necessary but reduces the refresh rate (Lecky-Thompson, 2008, p. 180), adding a delay in the processing in the server. In addition this reduces the bandwidth and solves problems of latency avoiding the situation in which the faster users take advantage over the slower users. This kind of "time manipulation has to be managed and control it, in order not to slow down the game. (Armitage et al, 2006, p.93).

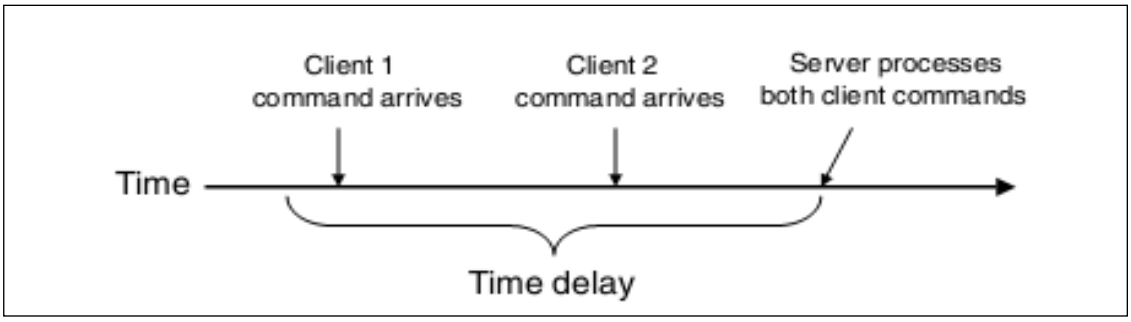


Figure 2: Time Delay on the Server.
Retrieve from (Armitage et al, 2006, p.94).

2.4.2. Time Delay and Time warp

The management of the time is very important in real time games, and the delays of the events can produce inconsistences. The inclusion of the time in the packets - a common time between all the players – may solve it. This may be necessary depending on the kind of game; for instance if the user_1 shoots to the user_2 in time_0 in a FPS game, but the event arrives to the server at time_1; in the time passed between time_0 and time_1, the user_2 could evade the shot and this event may arrive before time_1. This would cause a significant inconsistence. However including the time in the packets and checking the times on the server is possible to solve this situation. This technique is called "Time warp". (Armitage et al, 2006, p.94).

2.4.3. Compression techniques

Along with the reducing techniques, the compression techniques might be a good alternative to reduce the data transmitted. These techniques require post processing of data; therefore the processor load increases consuming more resources and accumulating delays. It is one of the reasons of why these techniques are avoided and are used as last option. A kind of compression technique is for instance, to draw a rectangle instead of sending all the pixels one by one, sending only one corner, the

colour, and width and height of the rectangle. Furthermore, compression increases the efficiency - reducing the amount of data- and the obscurity - it is more difficult that an attacker discovers what the meaning of the data transmitted is - (Lecky-Thompson, 2008, pp. 182- 183).

An inconvenience of the compression is the incompatibility with some algorithms; moreover the use of compression techniques increases the value of the packets, and then the possible damages or mistakes if the packets are lost. This fact is extremely alarming in the case of UDP due to its lack of retransmission in case of lost. (Lecky-Thompson, 2008, p. 183).Therefore, in contrast to the other algorithms used to compress audio, video and images; these algorithms must avoid the loss of information, because all the compressed bits must be restored when uncompressed (Armitage et al, 2006, p.96).

Lecky-Thompson (2008, p.184) maintains a weak, but efficient solution, using a run-length encoding (RLE). This method consists of reducing the number of bytes for the representation of objects, which can be a single character or a structure, through the replacement of multiple objects in the data stream with numbers which represent the replications of these objects. Consequently, it seems likely that these kinds of techniques are not available for beginner due both the difficulty for its management and the high level of knowledge required for its development.

2.4.4. Prediction

The last technique describes how to solve the problems of the "latency"," jitter" and "reduction packet send" is the use of predictions. Lecky-Thompson (2008, p. 185) identifies the next three areas where is possible to apply the prediction:

- Player Environment.
  In this case, prediction techniques try to guess the next move of the player.

- Non-player in game objects.
  This area is focused on the artefacts which are not controlled by the clients; for instance, if one player fires a bullet, the direction and velocity of the bullet can be known. Therefore, it is possible to calculate its path.

- The Game environment.
  It is the easiest area to predict, because the predicted objects are usually fixed or with very few movements, such as planets in a space game or tracks in race games.

The biggest issue of the prediction is when the data is permanently lost; this could be due to several factors, such as disconnection of the client or congestion in the network. Two approaches are used in order to solve this problem: one is the use of AI (Artificial

Intelligence) to substitute the real player with an AI equivalent. The other option is the freezing of the players. However, the use of one or other option depends on the game requirements. For example in Street Fighter IV, where small glitches in network connectivity cannot be tolerated, it is necessary to use AI; but in Counter Strike is use the frozen technique, because the players' movements cannot be predicted (Lecky-Thompson, 2008, p. 185).

2.4.5. Other methods

The previous four techniques may be the most popular and decisive techniques for solving the main problems in the network. However, it is possible to find a large list of techniques less known but at the same time useful to reduce the gaming traffic on the network.

The first one is called aura. This method requires a careful management and control of the information, because the game sends different information to each player depending on their position. For instance, if one player is hidden on a place without light, this player does not receive any information because he does not see anything. The name of this technique – aura – also defines the region that each player can see. (Armitage et al., 2006, p.96)
Another approach which helps to reduce the traffic between clients and server is the use of the architecture peer to peer (P2P) (Armitage et al., 2006, p.97). The use of this network architecture instead of the typical client-server architecture can offer remarkable features like the scalability, and the fault tolerance; although it increases exponentially the traffic on the rest of the network in concordance with the number of clients (Almashor & Khalil, 2010, p. 1). Some studies, as the publication of Almashor & Khalil (2011), attempt to reduce this issue by means of Two-dimensional Voronoi Diagrams (2D-VD) in DVE (Distributed Virtual Environments). However, these techniques, which are very sophisticated, are based on parameters of the network and distribution of the users; therefore this project does not consider them.

The techniques given above are deployed through the design and development of games; however it is possible to find other variety of techniques to limit the network traffic such as the use of firewalls or the use of tunnelling. The firewalls are exclusively limited to the dropping of packets; although the current technique of tunnelling which multiplex all the small packets in normal packets transmitted from the client to the server is more advanced. This technique has been created due to the apparition of the Cybercafés where the game players in local area networks (LAN) play together. (Saldaña Medina et al. 2011)

The use of one technique or another depends on the type of game because each game has its own requirements. A classification of network games according to its network necessities is shown below. This classification modifies the game classification based

on purposes of games, suggested by Lakkakorpi, Heiner & Ruutu (2002), combing two of its genres and adding a new to include the browser games.

Types of Network Games:

- Arcade games. - action games and simulators -
  The client in the majority of these games manage a single object, the object can be a character, a car, or even a football team, where the purpose of the game is to beat the others objects in "real time" using a virtual environment (Lakkakorpi, et al, 2002) and through a set of rules. The architecture used is usually the basic client/server architecture; but with extreme performance, reliability, and data consistence due to the real time requirements (Lecky-Thompson, 2008, p.43). These conditions require critical evaluation and a competent management of the game design, although sometimes it is not enough; the use of techniques to limit the network traffic in order to satisfy the QoS of the users without spending excessive resources is mandatory. Lecky-Thompson (2008, p.44) cites Greg Costikyan in his report "The Future of Online Gaming," where he states: "An online game's success or failure is largely determined by how the players are treated. In other words, the customer experience—in this case, the player experience—is the key driver of online success."
  This section covers mainly the genres FPS, RPG and sports; the FPS often uses the client/Server Architecture. However, the RPG games due to the large amount of users need to use the P2P architecture to reduce the traffic between client and server. Regarding the type of packet transmitted, both FPS and RPG usually coincide in the use of TCP for the establishment of connexion; and periodic and small UDP packets during the game execution. (Feng et al, 2005)

- Turn by turn games.
  These games incorporate an interaction between players which follows the statement that: a client cannot make movements even though the previous players have not executed their movements before; in other words the movements are sequential (Lakkakorpi et al, 2002, p.1). Therefore, in this kind of game the requirements of the network are not as high as the arcade games; it may be because this kind of game allows certain delay in the change of turn. Some games belonging to this group are the typical board games implemented on the network such as the chess or the Tic Tac Toe.

- Browser games.
  The browser games have increased its popularity in the last years, firstly with "the advent of the World Wide Web" and secondly with the emergence of the social networks. These kinds of games offer an attractive alternative in the development of online games (Lecky-Thompson, 2008, p.49), through the reduction of the implementation time and the creation of a new kind of player which only wants to play simple games (to have fun for a while). However, it

raises some concerns with security as a weakness of HTTP protocol due to its condition of open source; therefore the logic of the game must be stored in the server to avoid possible_misleading information provided by the users (Lecky-Thompson, 2008, p.89). It is very significant the evolution of the games in the social networks, focusing the attention on a particular game like FarmVille developed by Zinga. This game has become the most popular application of Facebook, with over 62 million active users and over 24.6 million Facebook fans since September 2010. Furthermore, the types of players who play this game are not only men, in fact the 60 percent of Farmville players are women, while the 40 left percent are men. Regarding the revenues, some researchers affirm that FarmVille generates over $1 billion of revenue every day (Online Marketing Trends, 2011).

As summary of the previous classification it is possible to argue that all the genres seem to agree in the importance of the users' satisfaction and the adaptation to the current trends as the social network; therefore the platform in which the game is developed has a fundamental role in the game design. However, focusing on the technical concepts the arcade games is the genre which requires more resources and has the higher limitations in terms of synchronization and latency. This is likely the primary reason that the majority of researches are about this genre.

2.5. Game Features which Affect to the Network Traffic

The distribution established by Lakkakorpi et al., (2002) is very important in the identification of the main game features which affect directly to the gaming traffic. The next outline exposes the indicated features in a simplified way to be used as reference in the subsequent analysis.

- Network connection type. It could be referred to:
  - Protocols:
    - TCP (Transport Control Protocol).
      The TCP protocol may cause delay if the network has a high loss packet rate. However, it is very reliable in the delivery of packets.

    - UDP (User Datagram Protocol).
      The UDP protocol is the opposite of TCP protocol; it offers a faster sending of information through connectionless.

  - Broadband Access technology
    - Wired Links (Ethernet).
    - Wireless Networks (WiFi).
    - Mobile Networks (WiMAX, LTE).

- Number of players.
  There is evidence to suggest that if the number of players increases, the gaming traffic also increases.

- Client - Server Architecture.
  - Client/Server architecture
  - P2P
  - Hybrid

In addition, the incorporation of sounds and graphics in the games will be considered in order to evaluate the network parameters and the performance of the game. At the end of the project will be verified if the hypothesis executed in this section is true or not.

- Sound effects
  There are numerous types of audio in the games such as the music of the background or the sound effects when a player shoots, or when the commentator in a football match game explains a movement; this audio usually is loaded and running from the client machine. However, if the game is running online, the performance of the game can be reduced considerably. Furthermore, the current online games offer the possibility of speaking in real time with the other players – audio in streaming- this kind of audio, which is not triggered from the client machine, may increase significantly the amount of traffic transmitted between

clients and server; it is the main reason to transmit this information through P2P channels.

- Graphic effects
  The graphic effects would be in a similar situation to sound effects; they are usually loaded in the client machine. However, this does not indicate that the graphics not suggest a problem for the network performance; if the client machine has several threads running, the process of reading the packets of the server can be locked and can cause a failure in the game system. In consequence, it is necessary to evaluate the effects of the graphics in an online environment.

The previous classification established can be taken as a summary of the main points which will be studied during the development and analysis process. Furthermore, this section of the report has been able to demonstrate theoretically the motivations for the execution of this project and several techniques used in the limitation of gaming traffic. Therefore, the following sections deal with a more practical approach through the implementation of a simple online multiplayer game.

This page intentionally left blank

3. METHODOLOGY

This section is divided into two different parts. Firstly, it is introduced planning and the first choices which have to be executed to start the development process explained in the second part. The second part, in addition to evaluating the obstacles and design consideration found during the project, will also describe and compare the several version of the game in order to analyse the network performance. This is the vital section when it comes to understand the approaches adopted in the next sections and the conclusions of this project.

Then, after the preliminary design, which consists of election the game and tools for the completion of the project, the evaluation of the available Hardware and Software is executed in order to know the characteristic of the resources and help to understand the posterior results. The development process depends on the tools selected; hence this process cannot be started until the proper control and considerable knowledge in the tools is acquired. As a result of this section a set of versions is obtained and it will be evaluated in the next section of this project.

3.1. Preliminary Design

The first step to develop a game is to plan the game design where it is necessary to decide:

- The game which will be developed.

- The tools used to develop and evaluate the game.

3.1.1. Game Selection

Before the network technology and the programming language for developing the game are chosen, it is necessary to know the game which will be developed and how it will be implemented. The platform for implementing the game is a computer, because it is specified in the project requirements; this way it is possible to avoid the cost which implies the acquisition of a console such as Xbox or PlayStation.

The kind of game which is thought for the project evaluation is an arcade game. The reasons are exposed in the background where it was demonstrated that this genre of games are the games with the highest requirements in terms of bandwidth and latency because of their condition of real-time and continuous updating. Currently, it is possible to develop a large list of games with a little imagination; however the majority of them are non-viable due to the time limitation. Therefore, a simplistic approach is opted in order to reduce the time of development.

The first games in the history were inspected for the acquisition of ideas; the first multiplayer game found was the "tennis for two" (1962), created by William Higinbotham and some years later the first version of "PONG" (1972) created by Nolan Bushnell, which also is one of the creator of the current Atari (Rabin, 2010, pp.4-8). Its popularity may have been triggered by their simple and conventional design, quality which the game developed in this project must have. Secondly via the deepening of the Higinbotham version, a big amount of manuals available on the Internet for the development of this game was found. Consequently, the time of the development process will not be very long, time which can be used for testing the network and applied changes over the game. It is how the idea of rescuing the PONG appears in order to evaluate the network through successive modifications and changes on the game.

In the following pictures can be observer the first arcade game commercialized:

Figure 3: Arcade Pong Game Machine.
Retrieved from Universidad Politécnica de Madrid: https://www.eui.upm.es/museo_virtual/3g/pong
through the next reference:http://www.androidguys.com/wp-content/uploads/2011/02/Pong.jpg

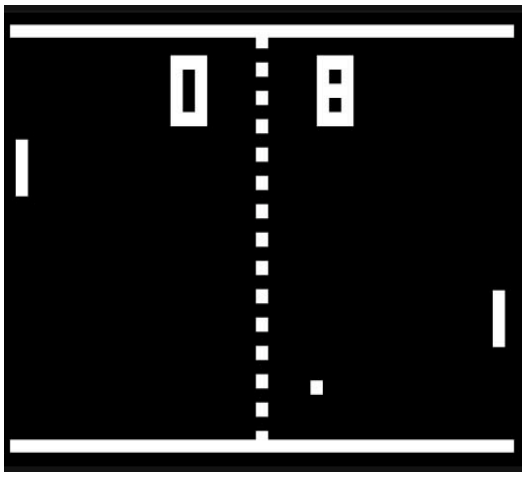


Figure 4: Capture from the Arcade Pong Game Machine
Retrieved from Universidad Politécnica de Madrid: https://www.eui.upm.es/museo_virtual/3g/pong
through the next reference: http://allthingsfadra.com/wp-content/uploads/2010/10/Pong01.jpg

Although the development of the arcade pong as an online and multiplayer game sis an innovative plan, it is not enough to evaluate all the necessities of the online games. Hence, the following mechanisms will be implanted in the game in order to cover all the issues which can be found in the development of online games.

- Multiple games at the same time.
- Inclusion of improved graphics.
- Addition of sound.
- Modification of the kind of connection.

The possibility of playing multiple games at the same time can be achieved through the implementation of a portal that allows the creation of several simultaneous games and the players' communication with instant messages. The preliminary approach can be watched in the Figure 4.0. Furthermore, graphics, sounds and the alteration of the kind of connection are crucial tests for the confirmation of the hypothesis expressed in the literature review, and then elaborating a list of recommendations for the optimally online games development.

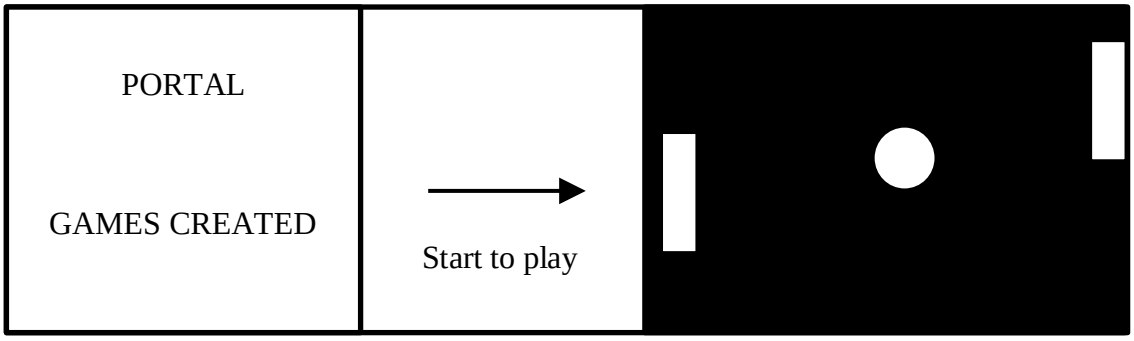


Figure 5: Scheme of the game

After the game election and the elaboration of the preliminary plan, it is necessary to choose the most effective tools for the development of the game and the implantation of the improvements.

3.1.2. Tools Selection

The development of the project requires the use of several tools. Two of the most relevant in the duration of the project are the programming language to develop the game and the sniffer to evaluate the traffic of the game.

Programming Language

It is possible to develop the game with any language of programming. However, there exist languages which are more appropriate than others. Therefore, with the accurate choice the process will be faster and comfortable (Rabin, 2010, p.189). The programming languages more common in the game development are: C++, C#, Java, and Scripting Languages such as Lua or Pynthon. As it is necessary to choose only one, the next four features are compared in order to select the best option.

- Performance.
  This project attempts to find the highest performance of the network with online games. Therefore, the game must be deployed with an efficient language of programming.

- High-level.
  The high-level offers an ease development through the use of resources already implemented; in other words the development time can be reduced.

- Expansion.
  The inclusion of a wide range of libraries is an interesting feature to facilitate the programming and allow the integration of different effects in the game, such as graphics or sound.

- Independent-Platform.
  This feature allows the use of any operating system not only to develop the game but also to play it; consequently these characteristics will reflect in an increment of the game players.

The table 3 shows the previous features matched by the programming languages in a simplified way which will be extended subsequently. Nevertheless, the advantage of Java over the rest of programming language in this project is clear without need of explication.

Table 3: Language of Programming.
Information retrieved from (Rabin, 2010, pp. 189 – 206)

Properties	Performance	High-level	Expansion	Independent P
Languages	/	/	/	/
C++	Very High	Low level	V	V
C#	Normal	V	support XNA	Need .Net (windows)
Java	High	V	V	V
Scripting Languages (Lua or Pynthon)	10 times worse than C++	V	V	V

C++ is one of the preferred languages for games; but it was not always, previously used to be "C". However, with the apparition of its high-level features - such as templates, and classes with different level of encapsulation- and its compatibility with C, C++ substituted rapidly to C. Furthermore, C++ has a wide number of libraries and a high performance. Nevertheless, C++ has several lacks, the most serious, which make it incompatible for this project, is its low level; and although the memory management of C++, created by the low level of programing, is a powerful tool for improving the performance, it is also hard to manage. (Rabin, 2010, pp. 189 – 194)

The game development of this project has to be fast and efficient. Rabin (2010, p.195) argue that "Writing in C++ is a bit like painting a wall with a paintbrush instead of a roller. You can do it, but it'll be more time consuming, and you'll probably do a worse job even though you have more control over every stroke and can reach the corners better". Therefore, observing the features of languages of high level such as Java and C#, it is possible to find several similarities. Both have a lower performance than C++; however both allow the rapid development of games through easy structures and functions. The main difference between C# and Java is in terms of Platforms. "Java is compiled into a special byte code and then interpreted on the fly by the Java Virtual Machine (JVM)" (Rabin, 2010, pp.195-196). This abstraction of hardware makes it possible to run in different platforms and increases its security, because the program is running in an independent virtual machine. This property and its rise in performance have done to Java a hard competitor of C++, especially in browser games, downloadable games and mobile games.
In the other side is C# which requires the libraries and the environment .NET for running in Windows; Linux and Mac have available a version called "Mono" which imitates .NET. However, it is still far from reach JVM. An advantage of C# is the availability of the extension XNA, which is a professionally-developed programming suite for Microsoft Visual Studio, which in addition of wining the prize in the category of "Hall of fame" of the front awards line 2011 competition, allows the development of games for three separate platform: PC, Windows Mobile and Xbox 360 (Sheffield, 2012, pp. 7 - 8). Stressing the last platform and looking at Java language, it is possible to observe the inferiority of Java in the "most popular gaming platform", the game

consoles (Rabin, 2010, p. 198). However, this project is focus on the pc development, so Java has advantage over C# due to its portability.
Currently, it is not strange to find part of games written in scripting language. The main services of this language are the high number of functions and the rapid interactions with the users, because the compilation of the code for reload parts of the program is not necessary. Against, the scripting language has an abysmal lack in the performance and tools support for the correct development of games (Rabin, 2010, pp. 199 – 203). One of the most popular scripting languages apart of python is LUA due to its portability with C/C++ and simplicity which have made winner in the category of "programming" of the front awards line 2011 competition (Sheffield, 2012, p. 11). However, due to the lack in performance, it is not possible to develop a whole game with a scripting language.

After comparing different languages, it seems obvious that Java is the suitable language for the development of the game in this project. This task is put in practice through a wide number of java libraries which smoothed the development process. Some of them even allow increase the performance using "hardware-accelerated 3D graphics through OpenGL bindings, or have access to sound hardware operations through OpenAL" (Rabin, 2010, p. 197). Therefore, the next step in the development process is the evaluation of the several libraries of Java in order to evaluate the network in the way more optimal.

Before the development process starts, the tool necessary for the analysis of the gaming traffic has to be chosen: The sniffer.

Sniffer Selection

The experiment will be executed through the capture of the gaming packets in the different versions of the game using a packet sniffer. Therefore, it is necessary to know what the packet sniffer is and what features are useful to perform a critical analysis of the gaming traffic.

A packet sniffer according to Qadeer, Iqbal, Zahid, & Siddiqui (2010, p.313), "is a program running in a network attached device that passively receives all data link layer frames passing through the device's network adapter". In other words, with the network card correctly configured this application is able to capture all the traffic of the network. The last fact is the main reason why the packet sniffer, also known as Network or Protocol Analyzer or Ethernet Sniffer, is used in terms related to the security (Qadeer et al, 2010, p.313). The administrator of the network uses these programs to manage the local networks through the identification of possible intruder, weaknesses of the system, or possible bottlenecks. On the other hand, the crackers control this software through techniques of eavesdropping in order to intercept keys or valuable information (Hannah, 2011). However, the objective of this document is not the security directly; it is the analysis of the traffic in order to create statistics about the traffic of the game depending on the changes implanted.

Currently a big variety of packet sniffers are accessible in the market, some of them are open source to the platforms more known such as UNIX and Windows; others are commercials. Some of the most popular sniffers suggested by Gordon Lyon (2010) are: Wireshark, Tcpdump, and Cain and Abel. However; before tackling the salient features of these sniffers which are very similar, the requirements which should satisfy the sniffers for the adequate fulfilling this project are rigorously selected.

Sniffer Requirements

In order to choose the suitable sniffer the following features are highly assessed:

- Availability
- Final Statistics
- GUI
- Management
- Open Source

The availability of the software for different platforms such as UNIX and Windows is an advisable feature, because the second computer dedicated to capture packets has the operating system: windows 7. Furthermore, it is essential that the sniffer allows management operations as the open, save, search and filtering of packets and set of frames in order to be able of compare the traffic of the different games versions and facilitate the process of packet reading.

Other interesting characteristic to expedite the analysis process are the incorporation of a Graphic User Interface (GUI) and the capability of generation of final summaries with the packets statistics. Finally, the packet sniffer selected has to be open source due to the economic limitation which involved a student project.

A set of requirements which were not indicated because the majority of the sniffers must satisfy to be called sniffer is the capability to display the information of TCP/IP traffic which involves the packets IP, TCP and UDP. Furthermore, it would be helpful if the sniffer captures the packet time accurately in order to reduce the time of game implementation, and then analysing easily the latency and delays between packets.

After the feature selection, the previous packet sniffers are analysed through the handbooks provided by the software manufactures, and the users' opinions.

Cain & Abel

Cain and Abel, often abbreviated to Cain, is a password recovery tool for Microsoft Operating Systems which allows inter alia, the recovery of several kinds of passwords by sniffing the network. Furthermore, it is freeware and was developed in the hope to be useful for networks administrators, teachers and everyone else that plans to use it for ethical reasons (Montoro, 2011). However, even being one of the most popular sniffers, Cain & Abel is focused more in specifying topics related to the security. Therefore, it will not be used in the analysis process of this project.

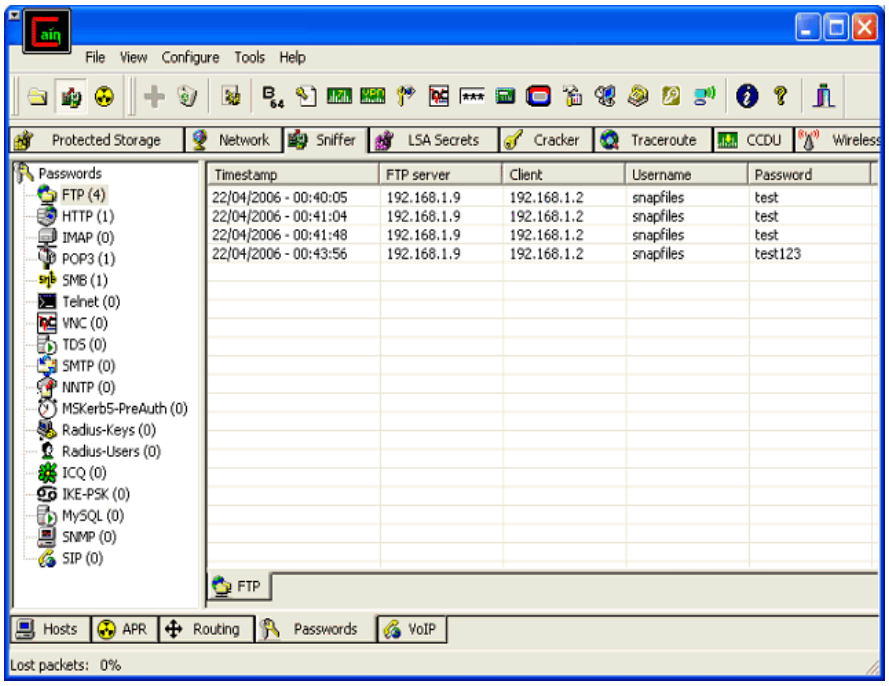
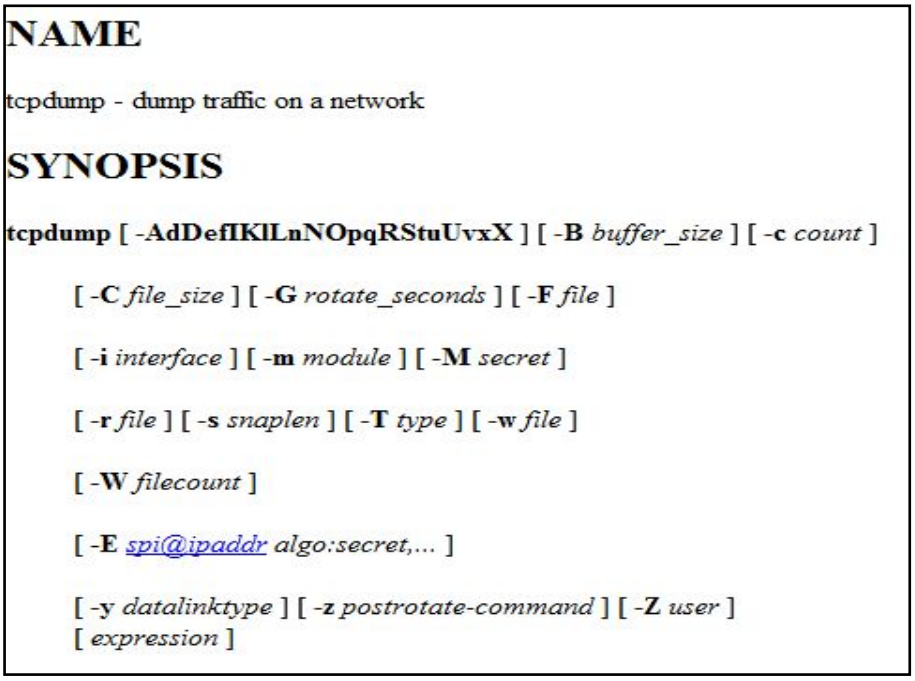


Figure 6: Sniffer Selection. Cain & Abel screenshot

Retrieved from: http://rajhackingarticles.blogspot.co.uk/2010_07_11_archive.html

Tcpdump

Tcpdump is a powerful command-line packet analyzer which uses libpcap, a portable C/C++ library for network traffic capture (Martin Garcia, 2010). This software is the IP sniffer used before the apparition of Ethereal (Wireshark). Despite it have not the GUI or parsing code for hundreds of application protocols such as Wireshark, it still continues to be used frequently. Some of the reasons of this are the high level of security offered and the fewer system resources needed to run it. Moreover, Tcpdump is an open source program available for Linux, and separately there is a version for Windows named WinDump with the same functions (Lyon, 2010).



NAME

tcpdump - dump traffic on a network

SYNOPSIS

tcpdump [**-AdDefIKlLnNOpqRStuUvxX**] [**-B** *buffer_size*] [**-c** *count*]

[**-C** *file_size*] [**-G** *rotate_seconds*] [**-F** *file*]

[**-i** *interface*] [**-m** *module*] [**-M** *secret*]

[**-r** *file*] [**-s** *snaplen*] [**-T** *type*] [**-w** *file*]

[**-W** *filecount*]

[**-E** *spi@ipaddr algo:secret,...*]

[**-y** *datalinktype*] [**-z** *postrotate-command*] [**-Z** *user*]
[*expression*]

Figure 7: Sniffer Selection. Tcpdump man page

Retrieved from: http://www.tcpdump.org/tcpdump_man.html

Wireshark

Wireshark – known as Ethereal until summer 2006 – is a network packet analyser for UNIX, Windows, and Mac OS X. Also, it is open source software and provides the possibility to capture live packet data from a network interface with very detailed protocol information and operations for opening, saving, filtering, searching, colorizing data captured and creating statistics. Moreover, the most powerful feature of Wireshark, Which also includes a Tcpdump as console named Tethereal, is the vast array of display filters - over 103000 fields in 1100 protocols as of version 1.6.5- (Sharpe, 2011).

However, the only disadvantage of Wireshark is in terms of security. Gordon Lyon (2010) points out that "Ethereal has suffered from dozens of remotely exploitable security holes". Therefore, it is not preferable to use it on untrusted or hostile networks (Lyon, 2010). This caution does not affect the development of this project because the programs are run within the VPN of the University of Portsmouth, where a firewall is in the entrance of this VPN dropping dangerous traffic and connection with the server without secure connection –SSH–.

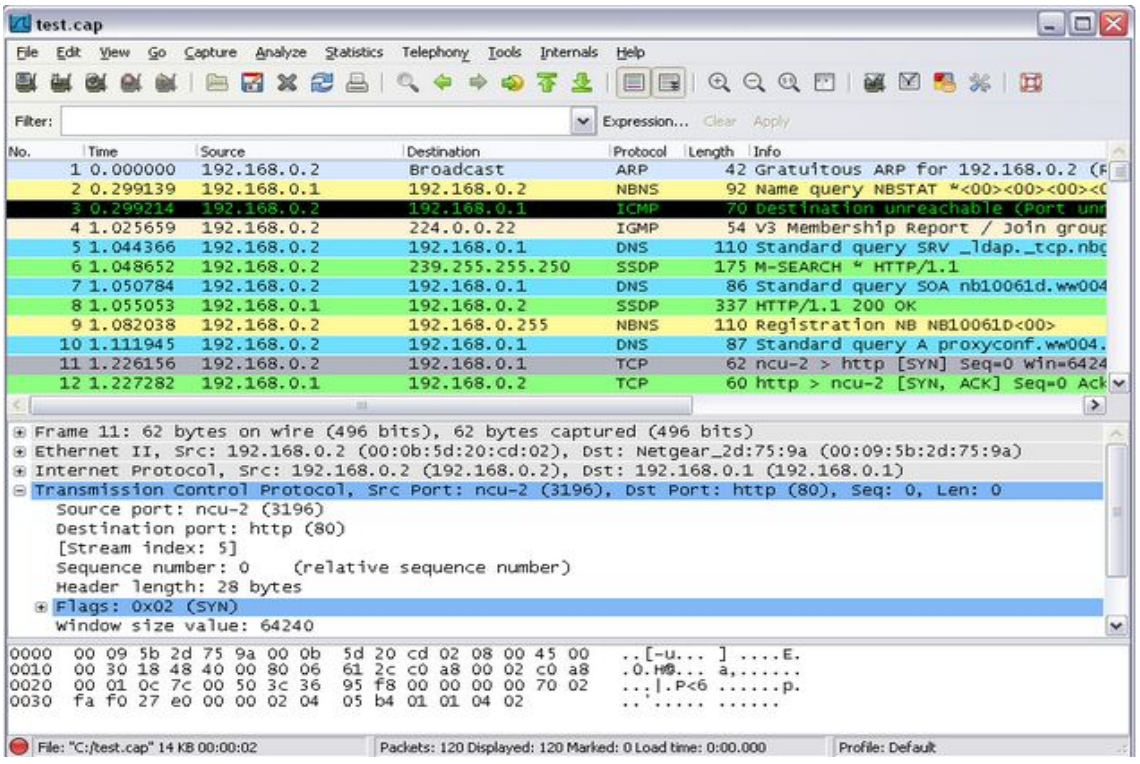


Figure 8: Sniffer Selection. Wireshark screenshot.

Retrieved from: http://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs

In order to facilitate the comparison between sniffers, the next table shows the main characteristic and features of each sniffer.

Table 4: Sniffer comparison

Sniffers	Cain and Abel	Tcpdump	Wireshark
Properties	/	/	/
Availability	Microsoft	Tcpmdum (Linux) WinDump (windows)	All the platforms
Final Statistics	V	V	V
GUI	V	X	V
Management	V	V	V
Open Source	Freeware	V	V

Analysing the last table, it appears likely that the suitable sniffer is Wireshark because of the lacks in GUI of Tcpdump which would increase the time of analysis and the exclusivity of Cain & Abel for windows, without mentioning the aspects of security which this project is not studied in detail. Furthermore, William Stalling (2011) maintains that "Wireshark is ideal for allowing students to study the behaviour of protocol not only because of its many features and multiplatform capability but also because students may subsequently use wireshark in their professional life". Therefore, the election of Wireshark as the sniffer used in the analysis process is correct. Finally, the website of Wireshark contains a very useful video tutorial showing how to use of this program. (Stalling, 2011, p. 836)

Figure 9: Sniffer Selection. Wireshark logo

Retrieved from: http://www.linux-mag.com/id/7896/

3.2. Development Process

In the last phase, the game and the tools which are used in the development process of the game and the evaluation of its traffic were chosen. However, the previous tools are not the only tools which will be used during the development of the project.

3.2.1. Available Resources

The resources needed to execute the project can be separated in two types: Software and Hardware. Some of these resources can be chosen, others are established by the university and the requirements of the project.

Hardware resources

The hardware resources are the most limited due to the conditions of security and costs. It is necessary to use at least two devices – PC's – to achieve an analysis in depth of the traffic. One of these devices works as server running the server program and the other computer works as client which can play the game through the server.

The available devices and their functions are the following:

- A personal laptop. – Client –

- An allocated computer in the room A2.3 of the Anglesea Building: chipmunk. – Client –

- Server.

The traffic is captured in clients and server through the sniffer Wireshark. However, last devices are not the only clients; the software is tested with several clients in order to perform a detailed analysis. These clients have similar characteristics to the computer: "chipmunk", because the majority of these clients are in the same laboratory than the computer chipmunk, the A2.3 laboratory.

The technical specifications of each device can be separated into three groups:

- Processing specifications

  This kind of specifications shows the power of processing of the system. It depends on the System type – 64 or 32 bits- ; microprocessor - frequency, type- ; memory RAM; and Operating system – the OS have different performance based on the distribution of memory.

- Game specifications

  The Game specifications used to be the elements which improved the quality of the game such as the graphic card, sound card or the display – screen resolution- .

- Network specifications

  These specifications often depend on the network and the connexion in the clients. However, the configuration of the network features in the server is very important to determine the quality of the server and the traffic transmitted.

All the specifications are relevant when it comes to understand the results. The tables below show the previous specifications.

Table 5: Resources HW. Personal Laptop

Asus A53SD-SX199V – personal laptop –	
Public IP address	**148.197.67.100 (Wi-Fi)**
Connection	Wireless (802.11 b/g/n)
Display	15.6" LED backlight: Glare HD (1366x768/16:9)
Direct rendering	YES
Graphic card	Intel® HD Graphics 3000
	NVIDIA GeForce GT 520MX 1Gb (dedicate)
memory (RAM)	4,00GB
Operating System	Windows 7 Home Premium 64 bits Service Pack 1
Processor:	Intel® Core™ i5-2430M CPU @ 2.40GHz,3MB (L3) Cache
Sound card	Realtek High Definition Audio
System type	64-bit Operating System

Table 6: Resources HW. Computer of the room A2.3

Linux chipmunk.ee.port.ac.uk – Computer of the room A2.3 –	
Public IP address	**148.197.27.14**
Connection	Ethernet (eth0)
Display	1366 x768 pixels
Direct rendering	NO
Graphic card	-
memory (RAM)	8,00GB
Operating System	CentOS release 5.5
Processor:	(Intel® Core™ i7 CPU 860 @ 2.80GHz.)* 4 Quad core
Sound card	No sound card found
System type	64-bit Operating System

Table 7: Resources HW. Server

Linux project1.ee.port.ac.uk – Server –	
Public IP address	**148.197.34.67**
Connection	Ethernet (eth0)
Display	-
Direct rendering	NO
Graphic card	VGA compatible controller nVidia Corporation C51G [Geforce 6100]
memory (RAM)	2,00GB
Operating System	CentOS release 6.6 (Final)
Processor:	AMD Athlon™ 64 Processor 3500 + / 1000.000MHz
Sound card	2 sound card ICH [Intel ICH]
System type	64-bit Operating System
Settings for eth0	
Properties	**Values**
Supported ports:	[MII]
Supported link modes:	10baseT/Half 10baseT/Full
	100baseT/Half 100baseT/Full
Supports auto-negotiation:	Yes
Advertised link modes:	10baseT/Half 10baseT/Full
	100baseT/Half 100baseT/Full
Advertised auto-negotiation:	Yes
Speed:	100Mb/s
Duplex:	Full
Port:	MII
PHYAD:	1
Transceiver:	external
Auto-negotiation:	on
Supports Wake-on:	g
Wake-on:	d
Link detected:	Yes

The network is also considered as a hardware resource; therefore it is analysed before starting the development and analysis process. The previous server is inside the LAN of the University of Portsmouth. This network may suppose a limitation to analyse the traffic in real conditions due to an internal firewall which protects the network from possible intruders through external packet dropping. As a consequence, the communication with the server is only attainable for users inside of the university LAN.

The last sections of the report comment on some advantages in terms of security for VPN (Virtual Private Network) – LAN - and the firewall available in the network of the University of Portsmouth. However, it has been proved that there are also drawbacks such as the impossibility of analysing the network in the same conditions than the game players on the Internet with the typical delays that seldom appear in LANs.

The firewall can be avoided through the use of secure communications as SSH protocol provides; nevertheless it is hard to implement the SSH protocol in Java. Therefore, the other possibility, which provide the use of this protocol, is to establish a remote connexion with one of the computers inside of the LAN, and then to run the client in that computer. This section will be explored subsequently in order to find solutions to the problems presented by the firewall.

The specifications of network are as following:

Table 8: Resources HW. Network

Network of the University of Portsmouth	
DNS	197.148.5.64
Proxy	Lion Server: lion.ee.port.ac.uk (197.148. 27.206)
Bit rate	100 Mbytes/s per each device

Software resources

The previous devices in addition to be hardware resources also contain software resources which are necessary for the project development. The software that is used throughout this project is:

- Language of programing: Java Development Kit (JDK) – tool –
- Libraries of Java
- IDE (Integrated Development Environment) NetBeans – program –
- Sniffer: Wireshark – program –

The language of programming and the sniffer was chosen in the last sections. The IDE selection, which is a program to improve the development process, was executed depending on the available programs of the allocated computer. Moreover, Netbeans is a free, open-source IDE with all the tools and functions needed to create applications in several platforms – web, Windows, Linux, Mac, mobile – with a large list of functions in Java platform, as well as with, PHP, C/ C++ and more. (Oracle Corporation and/or its affiliates, 2012)
The table below provides the version and the specifications of the last software in summary.

Table 9: Resources SW

SOFTWARE	Personal Laptop	A2.3 PCs	Server
Features	/	/	/
Operating System	Windows	Linux	Linux
Java version	1.7.0_03	1.6.0_21	1.7.0_03
Wireshark version	1.6.7	1.0.15	1.0.15
NetBeans version	7.1	6.9.1	-
Libraries	Libraries	Libraries	Libraries

The different versions in the last software involved several confusions at the start of the project because although the programming language – java – runs in different platforms. Therefore, the methodology which is used in NetBeans to create projects is different. When a Windows Netbeans project is opened in Linux Netbeans, it is likely to find mistakes and errors in the compilation process. This is because, even it is easier to write portable programs –run on different types OS- in java than in other programming languages; differences between compilers, JVMs and computers can make portability difficult to achieve. Consequently, simple programs in Java do not guarantee portability. (Deitel, P. & Deitel,H., 2012, p.18)

Furthermore, the version of Java in the server was not always the 1.7.0_03; at the start the version 1.4, which is incompatible with the inherences of the new Java versions, was installed in the server. Consequently, the current version was downloaded and installed through the execution of remote commands, and then set as the default version in the server system.

Libraries and packages used in the game development are detailed over the course of the project.

The websites used to make the installation of the current version of Java were:
http://docs.oracle.com/javase/7/docs/webnotes/install/linux/linux-jdk.html
http://www.linuxnix.com/2010/12/how-to-change-the-java-version-in-linux.html

At this time the project is well on the road to the start the development of the software and the analysis of the results; however a careful planning is required to meet the deadlines of the project. The planning is showed in the appendix E.

3.3. Initiation into Java

In parallel to the evaluation process of the resources, Java programming language was learnt successfully. Below an analysis in-depth and an objective evaluation about this programming language is showed.

Paul and Harvey Deitel (2012, p.2) claim that Java is used today in more than a billion general-purpose computers and billions of mobile devices. Therefore, several editions are reachable through the website of "Oracle" in order to satisfy the necessities of each device depending on the purpose of the application developed.

The most known are: Java SE, EE and ME

- Java Standard Edition 6 (Java SE 6).
  Currently, the version 7 is available and it has new functionalities in terms of strings, resources management – Objects, input, output- , and event handlers. Furthermore, both versions sixth and seventh are entirely compatible.

- Java Enterprise Edition (Java EE)
  This version is mainly used in order to meet the requirements of large-scale development, distributed networking applications and web-based applications.

- Java Micro-Edition (Java ME)
  Java ME is geared towards the development of applications for devices with low power, such as smartphones or tablets.

(Deitel, P & Deitel,H, 2012, p. 2 - 3)

The objective of this project is to research the network through the development of a computer Java application. Therefore, the edition used is Java Standard Edition, as this project does not have commercial or industrial purposes.

Lewis Carrol (1871, cited by Savitch, 2012, p.21) wrote in his book "through the looking glass":

> 'The time has come,' the Walrus said,
> 'To talk of many things:
> Of shoes–and ships–and sealing wax–
> Of cabbages–and kings . . .'

Previously to the development of the Java applications, it is necessary to know the principles of Java which is a programing OOP (Object Oriented Programming):

- Encapsulation .
  Encapsulation means hiding the details of the objects and showing only the relevant information.

- Polymorphisms.
  As it can be presumed through the name of this property, which means in Greek: "many forms", the polymorphism allows that a program instruction has different outputs depending on the input. In others words, it offers different object behaviour for different context.

- Inheritance.
  Inheritance is a way to coordinate and to classify the objects based on a hierarchy where the main class is called superclass and the classes that are resulting from the superclass subclasses. The names are 'base class' and 'derived class' in C++. However, the names are not the only difference; Java does not support multiple-inheritance - which occurs when a class is derived from more than one direct superclass-. The alternative which Java proposes is to use interfaces. This mechanism based on polymorphism provides many benefits of multiple-inheritance while avoiding the associated problems.

(Savitch, 2012, p.23 - 25)

The first property is very useful to keep the security and control on the game in order to avoid possible cheating and holes in the game; whereas the polymorphisms and inheritance are very helpful to reduce program developing time through the code reuse.

The design of the game will be focused on the available resources optimization, particularly the time which is very limited. Therefore, the inheritance and the polymorphism are the major contributions of Java together with the rich set of predefined classes available in the Java APIs (Application Programming Interface) or Java class library that can be reused instead of "reinventing the wheel". These classes are contained in packages that can be used through the commands: 'import' – to import the data – or 'extend' – to use inheritance–. (Savitch, 2012, p.73)

The next representation forms are used in order to facilitate the Java game understanding developed throughout this project:

- Pseudocode

- The UML (Unified Modelling Language) Diagram

The Pseudocode is a simplified representation of the code using an informal language which can be useful to develop and understand algorithms, avoiding the strict details of Java language syntax. (Deitel, P & Deitel,H, 2012, p. 140)

The UML is the "OMG's most-used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure" (Object Management Group, Inc., 2012). The UML Diagram, which is used in this project, is a graphic scheme to understand the methods, properties, attributes and inheritance of the objects.

The first time that a Java application is studied by an engineer with knowledge about C# and C++, it seem likely that this person find several similarities between these two programming languages and Java. However, there are marked differences in the threads and graphic management.

Beginning with the graphics, one of prime considerations is the differentiation of the two main Java packets to create a GUI (Graphic User Interface):

- AWT – Abstract Windows Toolkit – (java.awt)

- Swing (java.swing)

The AWT components look like the native GUI components of the platform on which Java is executed. However, the Swing GUI components allow the specification of a uniform 'look-and-feel' - interaction and appearance of components with the users- for the applications because these components does not depend on the platform or OS.

Most Swing components are lightweight components: they are written, manipulated and displayed completely in JVM – Java Virtual machine-, whereas AWT components are heavyweight components, because they rely on the local platforms. (Deitel, P & Deitel,H, 2012, p. 555 - 556). Therefore, this project attempts to use Swing components in order to ensure the compatibility between different OS. Nevertheless, the total separation of an application from the operating system is not always possible. Furthermore, the AWT package is also the packet which manages the majority of the system events; hence it has to be used.

The threads in Java are a popular and powerful tool which allows the execution of concurrent task simulating the effect of real time in the system. Java was designed with threads in mind; therefore it is likely easier to work with threads in Java than in many other languages. (Bracket, Barker & Vanhelsuwé, 2003, p.4). The online games need a correct use of the threads in order to make more dynamic system and administer adequately the system resources.

Anne Frank (1929 – 1945, cited by Deitel, P & Deitel,H, 2012, p. 1) have stated in 'The Diary of a Young Girl' (1947): "How wonderful it is that nobody need wait a single moment before starting to improve the world"

Through the use of threads nobody will need to wait until the termination of the previous task, because the tasks are executed concurrently; therefore the loss of information and waits are reduced.

Java provides three ways to use threads:

- Extends the threat class
- Implement the runnable interface
- Use anonymous inner classes

The preferred way of thread implementation in this project is the use of anonymous inner classes, but the other ways are also valid. Despite all the advantages, the threads may cause severe problems such as unnecessary delays – Oversynchronize- or deadlocks. (Bracket, Barker & Vanhelsuwé, 2003, p.10 - 11). Thus, it is necessary to synchronize the threads and to know when to use threads in order to solve and avoid these problems.

For example the threads can be used in these situations:

- When doing any network communication
- When doing any massive calculation
- When using files to load or store information

(Bracket, Barker & Vanhelsuwé, 2003, p.14)

It could be argued that not all the games need threads; however this is not true. All the graphical applications need at least one thread apart to dispatch the events of the users. Furthermore, this project requires an efficient synchronization in order to evaluate the network in the best way possible.

Finally, the game developed is online and multiplayer; therefore it is necessary to implement a channel to make possible the communication between users through Internet. There are two ways to achieve the previous objective: through the use of applets and the browser or sockets and system applications.

3.4. First approach in the development of the game

Through the revision of the tutorials and PONG versions in order to evaluate the development of the game using applets or sockets it was found that there exist several webpages which offer the possibility to play PONG online like the website pippinbarr: http://www.pippinbarr.com/games/pongs/Pongs.html. However, the applications found for computers lack the capacity of online connection; hence these offline PONGs do not provide any help when x the best kind of communication to analyse the network is decided.

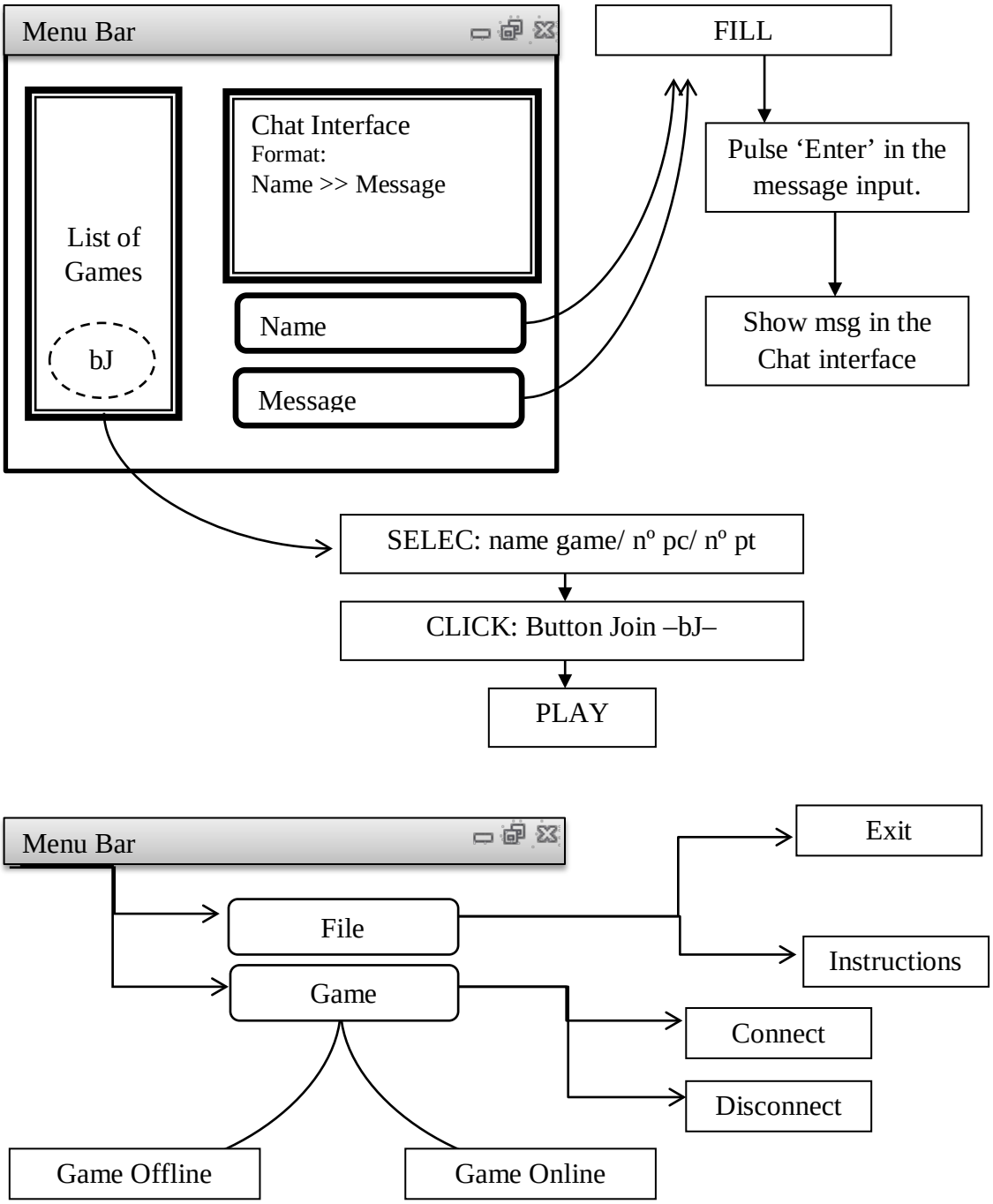
Researching the two possibilities of connexion together with the game features, Walter Savitch (2012, p.30) states that "We typically will display the graphics inside of an applet because it is easier to do, especially for beginners". To run an applet, the code has to be compiled in the same way that any other Java class. However, the applet must be executed differently. The normal way to run it is as part of a Web page (Savitch, 2012, p.37). Consequently, this fact involves the drawback that the analysis of the gaming traffic would be limited to the HTTP protocol and the port 80; whereas the development of a Java application using sockets allows the use of different ports and types of connexions – UDP or TCP.

The sockets are a low-level programming interface for networked communications, where Java provides a simplified model that makes network communications easier than the other language programs (Niemeyer & Knudsen, 2005, p. 436). Moreover, this mechanism – the socket implementation- enables more detailed study about the network which is often characterized by the five attributes: topology, bandwidth, latency, reliability, and protocol (Davison, 2005, p. 770). Therefore, the mechanism chosen to execute the communication between server and clients is the sockets (because of its proper control over the network parameters).

After the study and learning about the tools and possibilities that Java offers to develop applications, the following approach, which appears represented through the scheme of the figure 7.0, is conceived in order to accomplish the first objective of the game: to allow playing multiple sets of games simultaneously. It is achieved through the development of a portal with the following functions:

- A chat that can be used to coordinate the games.
- A panel where the players can join games created previously.
- A menu with the options:
    - Create a game offline
    - Create a game Online
    - Connect/Disconnect with the server
    - Instructions which provides information about the game.

First approach:



Number of current players: nº pc
Scheme Total number of players: nº p
Message: msg

Figure 10: First approach

The above diagram shows the structure and the main functions of the game; however it is not enough to understand the procedure completely. Therefore, the following Pseudocode is developed to gain a better understanding of the performance of the chat and game creation.

- Initiate the program
  Connect with the server
    If the client is already connected => error
    Else connect and enable the chat send.

- Create the game Online
  Wait for players
    If the waiting time is too long close the game
    Else start the game (server)
      Send periodically the position of the ball
      Listen to the movements of the players and send it to the other player.
    When the time indicated at the start of the game is over finish the game.

- Write the name and the message in the input box.
  Push 'enter' key when the message is finished.
    If the name box is empty add the text "anonymous"
  Send the packet to the server
  Receive the packet and send it to all the clients connected (server)
  Receive the message and show it in the chat interface.

- To join a game, one of them has to be selected
  Push the button 'join'
  (Server)
    If the game is available
      And
    If the number of current players is lower than the number of total players
    Return ACK with the port of the game,
    Else Return NACK with the error message

The communication between clients and server is necessary to meet the previous design. It can be done by the following flows of packets –Basic Sequence of communication–. These flows represent two possible types of communication:

- The management communication: Control packets

- The game communication: Game process.

Control packets

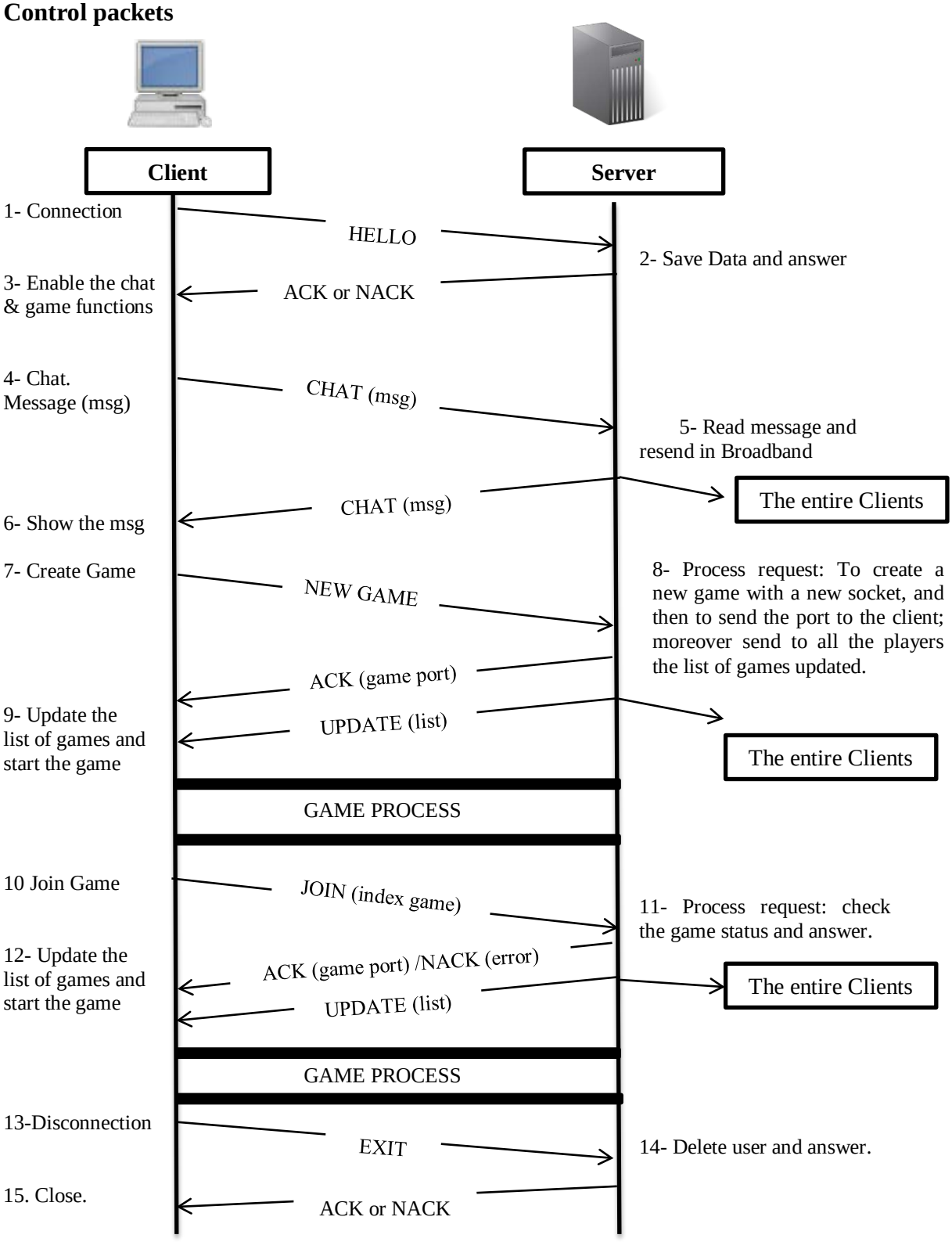


Figure 11: First approach. Control packets

Game process

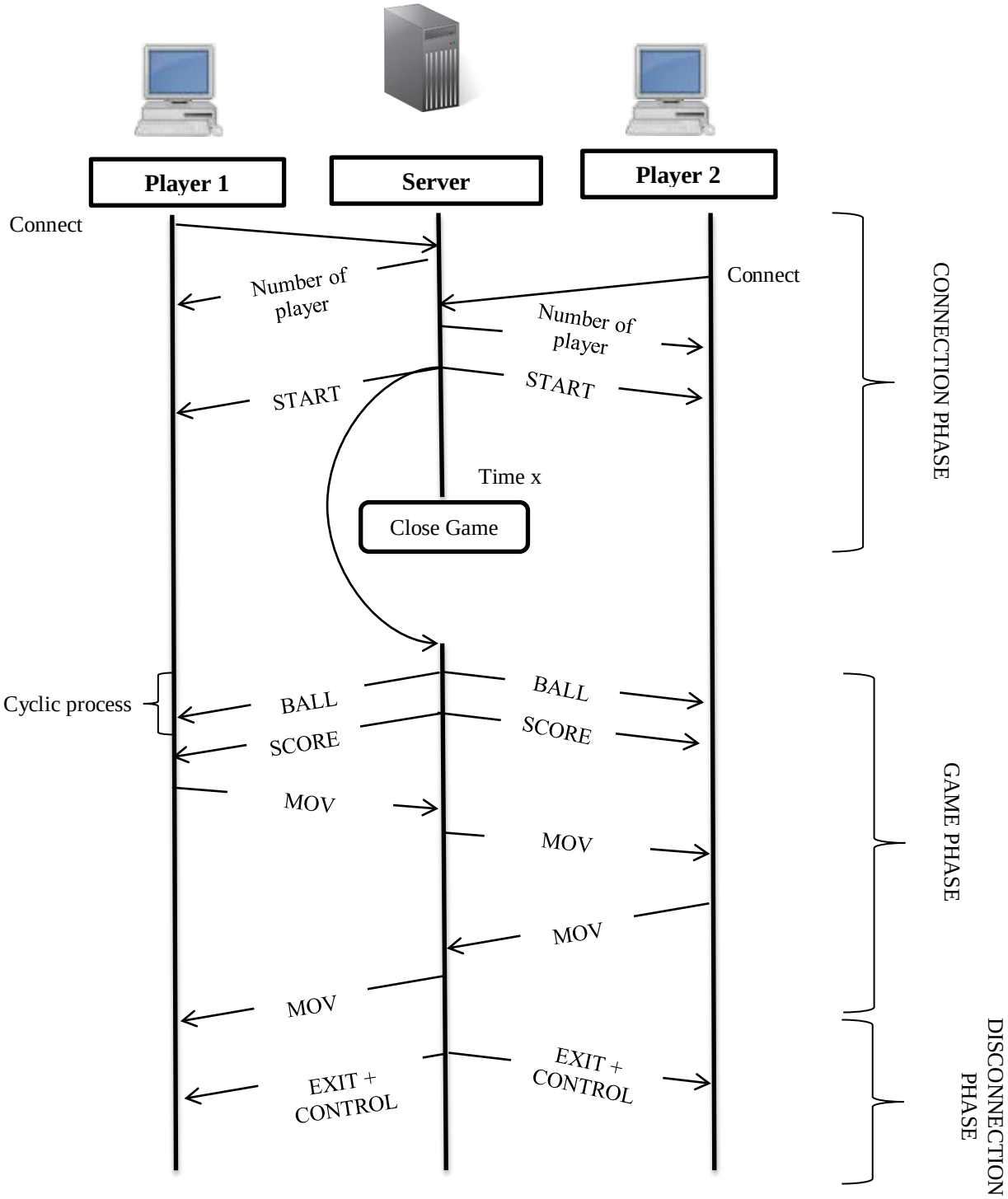


Figure 12: First approach. Game process

It is possible to find some details and considerations in the communication diagrams that can be valuable for the future designs and election of protocols. However, the behaviour of the topology is not possible to assurance until the first capture of traffic. Therefore, all the information about the communication protocols and game performance are hypothetic, which will be verified in the following section –Chapter IV –

The control packets represent an architecture in which for each client request there is at least one response. This fact provides a control of errors similar to the TCP protocol; moreover the messages do not have a big amount of information. Thus, the selection of the UDP protocol for the deployment of this communication client – server in order to assurance the highest performance of the game will be announced in the following sections.

The game process appears more complex due to the synchronization and control of the connectivity between players. The first approach divides into three phases the game process:

- Connection phase
- Game phase
- Disconnection phase

The first phase is important because it highlights the control of server resources through the timer, which deletes the game if the number of players is not complete in thirty seconds. After the user request to connect, the server sends the number of players, which can be cero – player 0 – or one –player 1–. As soon as the player 1pins the match, the server sends a notification to both players and the game can begin.

Subsequently to the accomplishment of the number of players the game is started. During the game phase the server sends the positions of the ball periodically; in addition to receive the movements of the players and update their new scores.

The disconnection phase can be executed from the server when the game time is over or from the client when one of the players closes the game. When the timer is over, the server compares the players' score and then notifies the winner of the game. When one of the players leaves the game, the other player receives a notification indicating that he is the winner because the other player left the game.

Before it was stated the preliminary design of the application which will be developed throughout this project. However, this project deals with the traffic analysis; therefore the technical considerations are pushed aside and the following sections are focused on the essential concepts for the understanding the network traffic, commenting upon the problems which can affect the available resources such as bandwidth, delays, QoS, or

even time of development. If the code is wished to be analysed, the information about the access to the code is in the appendix F of this report. The game is divided into two parts: the server and the clients; moreover in each part, two different types of communications have to be developed, one for each kinds of operations.

- Administration operations.
  - Connect/Disconnect
  - Create/Join games
  - Send/Receive chat messages

- Game operations
  - Send/Receive game information – Ball position, score updates
  - Send/Receive players' movements

The communication in the game portal for the administration operations is executed through the use of UDP sockets due to the briefness and the content of the messages. Moreover, the UDP connection allows to the server to provide a service faster and more robust because there is no retransmission if the packets are lost due to the connectionless of the UDP protocol. (Stalling, 2011, p.759)

The research and evaluation of the gaming traffic is focused on the game operations; therefore this part of the game will be static over the development of the additional improvements and the different versions in order to provide stability to the development process.

The first step executed in the development process is the elaboration of the graphic interface represented in the figure 7.0 through the GUI of NetBeans in order to save time in the development process. After the creation of client interface, the next phase is to design the packets which will be transmitted between client and server following the previous sequences of communications.

The packets are internal classes of a higher class which store the operations of serialization/deserialization, marshalling/unmarshalling and relevant information for the running of the application. This class is called 'Information' and the variables are the same in both side of the game – server and client -; however the methods are not the same. The UML class diagrams and other useful information are available in the appendix G.

The methods of serialization and deserialization involve a change from class type to String type and vice versa in order to facilitate the information transfer through the network. The implementation of these methods caused some problems due to the high level of Java. The programming language C++ provides a function called 'memcpy' which change the type of variables in byte level. However, Java does not have this

function because of its high level condition; thus the information transfer is more difficult to develop.

The serialization is implemented changing the divergent variables into a common format – String type- with high level functions and adding the character 'new line' - \n – at the end of the object transmitted. The process of deserialization involves the inverse operation, it change the Strings received into the corresponding variables or objects. These types of conversions entail to know exactly the type of packet which is sent and received in the process of serialization and deserialization. Therefore, the methods marshalling and unmarshalling are not meaningful and relevant to the game, because these methods are implemented to separate the process of serialization from the packets through the use of a common packet which could be implemented using an abstract class in Java. Consequently, the communication process is only based on two methods: serialization and deserializations which are used both in the parts the client and the server. Nevertheless, the content of these methods in each part is not the same.

In addition to the packets exchanged by clients and server, the class 'Information' of the server needs to manage the clients connected and the games run in the system. It can be done with a large amount of resources that Java provides such as data bases. However, time is saved through the use of simple temporary arrays because the arrays does not need all the control and connexions that a data base needs; however the server may overload if the memory of the arrays is not managed correctly. Consequently, the control of users in the portal is deployed accurately using several methods to connect and disconnect clients.

Finally, Java provides a garbage collect in order to avoid memory leaks; however "Java will not garbage collect an object until there are no remaining references to it" (Deitel, P & Deitel,H, 2012, p. 484). Therefore, in order to enable the garbage collect the elements of the arrays are set on null when they are not needed.

Abstract class
An abstract class is a class with at least one abstract method, which does not admit any object. However, it is possible to create an array with instances of its subclasses. (Deitel, P & Deitel,H, 2012, p. 400 - 401)

Array
An array is a group of variables containing values that all have the same type. (Deitel, P & Deitel,H, 2012, p. 242)

Game operations

Once the administration part has ended, the development process is focused on the PONG game implementation through the use of tutorials which despite of their lack of network connexion, are very useful in order to reinforcing the knowledge learnt about Java, and then to develop the initial version which will be used as a basis for further improvements in the gaming features identified in the literature review – Chapter II - .

The first version of the game is a simple multiplayer pong for two players; therefore it is necessary two pallets and one ball. These components are implemented in a simplified way in order to can reuse it in the following versions of the game. The version executed by Fabiel Casas (2010) was very helpful to the first version development according with a simple and manageable design which allows meeting the previous requirement.

The ball and the pallet of the game are constant components in order to develop a game in a simplest way reducing the amount of traffic through the transmission of screen coordinates.
Although the version of Fabiel Casas (2010) is very useful, also it is far from perfect; the following changes and improvements are applied to this version in order to achieve order and control during the network implementation.

- Globalization of the variables
- Translation of the variables into English
- Addition of a timer.

The implementation of the variables as global variables reduces the development time because the changes in the game features are applied directly in only one section of the game. The timer allows controlling the duration of the games; therefore in the analysis and comparison of the gaming traffic all the games will have the same duration. Finally, this version is in Spanish; therefore the translation into English is executed in order to make easier the understanding of the software.

Moreover, the control elements to manage the game, it is necessary a cyclical process to refresh the graphics and to make the game possible. This process may be achieved using a large list of resources which Java provides such as loops; however this game will use timers to trigger the updating method calls at regular intervals.

Two of the most acclaimed timers which can be implemented in Java are:

- Using the Swing timer
- Using the utility timer from java.util.timer

(Davison, 2005, p. 39)

However, Davison (2005, p. 39) reject the swing timer because does not have the necessary accuracy for graphics representation in games; and the flexibility and scheduling of the utility timer which makes an excellent use of the system resources are highlighted. In addition to the arguments of Davison, Fabiel Casas (2010) also uses this timer in his offline Pong game to run it in a practical way for the user enjoyment. Therefore, this project will be developed through the use of this timer in order to achieve the higher performance.

The following figure captured from Davison (2005, p. 40) is modified to explain one of the main operations that the utility timer allows in the Pong game.

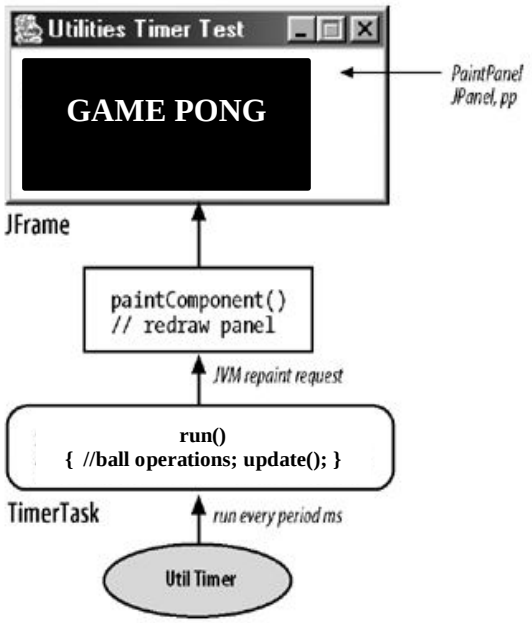


Figure 13: Utility timer ball (Davison, 2005, p.40)

The development of the networking connexion is started with the packet design which it is necessary to establish the communication between players and server. This communication is implemented with reference to the example of the game Tic-Tac-Toe online which appears in the book of Paul Deitel & Harvey Deitel (2012, pp. 1150 – 1165). This example uses a TCP socket and a thread per each player. When the server accepts a request for connection, it assigns a thread for each player, and when the number of players is completed, the game starts.

Although the communication protocol would be the same - TCP - , Tic-Tac-Toe and Pong belong to different genres of games. Consequently, the synchronization between players seems radically different. Pong is an arcade game and Tic-Tac-Toe is a game by turns; these facts involve that the control over synchronization of the Tic-Tac-Toe game is higher than Pong. However, the communication management is more refined in the Pong game due to the higher flow of packets.

A class called 'Information_game' is created in order to make possible the game communication. This class, as the administration operations, is composed with the packets and the game status. Observing the UML class diagrams in the appendix G, it is possible to find similarities between both classes: 'Information_game' and 'Information'.

The basic functions completion of the first version through TCP sockets provides noticeable differences between the TCP and UDP implementations used respectively in the communication of the game and portal. The TCP connection was apparently more complex in the sent and reception of packets, as a sophisticated exception control was necessary to develop in order to avoid mistakes in the packet reception. However, these facts are not definitive due to the discrepancy in the amount of information transmitted in each task; this comparison is finished after the development of the UDP version and the analysis of the gaming traffic.

The initial game version allows the movement of the pallet and the ball in the window of each player, the control of the game duration and game score. Furthermore, a procedure to close the game in the server if the number of players is not completed in thirty seconds is developed in order to optimize the resources of the server. This task is executed through the modification of the game status created in the portal management. The transference of this variable is achieved thanks the passed by reference of Java. It is a curious feature of Java that unlike some other languages, the choice of selecting pass by value or pass by reference is not allowed. This choice is automatically, if the data is a simple element, it is passed by value; however the arrays or the objects are passed by reference (Deitel, P & Deitel,H, 2012, p. 262). Therefore, the status modification is task without a hitch.

3.5. Expansion Process

The first version of the game developed through TCP socket is modified in accordance with two analytical approaches in order to analyse the features identified in the background. Firstly, an approach focuses on network features – UDP protocol –, and secondly other approach focuses on the gaming features – graphics and sounds effects – are designed.

The approach related with game features was conceived thanks to the quote of Peggy Walker "When faced with a decision, I always ask, "What would be the most fun?"" (Deitel, P & Deitel,H, 2012, p. 37), which defines very well the ideology of the games, because a successful game have to be funny. Therefore, it is logical to evaluate the features which make the game funny, such as the innovative graphics and sounds effects.

In contrast, the evaluation of network features are also essential to get a success game; as a result the first version involves the change of TCP protocol for UDP protocol in order to evaluate both protocols and to decide the best game protocol for arcade games.

3.5.1. UDP Connection

The protocol used in the game communication is an important factor when a game is developed. This section shows mainly the technical considerations of the protocols; however in the next section the traffic is analysed and it will be executed a comparison between the protocol TCP and UDP more detailed.

Although the TCP protocol had more errors in the reception of packets that have to be managed using exceptions, through the implementation of the UDP protocol on the game has been possible to observe the strengths of the TCP protocol, such as the synchronization and coordination of clients.

The synchronization in the online games is an essential part of the game. The TCP server can create a thread for each player when the client connection request is accepted; and then managing each connection separately in each thread. Therefore, the TCP structure seems more efficient when the synchronization of the player is complex. However, the player number may be a limitation because each player requires a socket; consequently the server may run slow and usually there is a maximum number of socket that one server can accept.

Regarding the advantages of the UDP connection, the reception and sent of UDP packets is remarkable because of its simplicity and speed. It may be due to its independence between packets. Moreover, the packet rate is also increased, condition that mat be favourable in the arcade games like Pong.

The Pong game does not show differences between connexions in the proofs executed during the development process; nevertheless the previous comments will be increased with the increase of experiments during the analysis process. Moreover, the game process is identically to the TCP process showed in the figure 7.2; hence it is not explained.

3.5.2. Sound Effects

It could be argued that the addition of sounds and music on the games is a deciding factor in the games success. This project with the addition of sounds effects attempts to evaluate the users' preferences and their opinion about the game. In addition the performance and the sounds effects of gaming traffic are analysed in order to obtain a complete list of recommendations for the development of online games. The sounds are loaded and reproduced using applets, which provides the option of load the files

The sound effects can be divided into two sections:

- Background music

- Sound effects - when the ball is hit or the score is increased-

The inventory of the music which can be listened in the game is:

- Football design:
  Background: Sport event. Retrieved from

http://static1.grsites.com/archive/sounds/background/background018.wav

  Hit of the ball: Ball kick a football ball. Retrieved from
  http://efectos-de-sonido.anuncios-radio.com/football52.mp3

  Goal: Soccer Goal in Spanish. Retrieved from
  http://www.audiomicro.com/tracks/dialog/735797

- Water design:
  Background: Ocean. Retrieved from
  http://static1.grsites.com/archive/sounds/nature/nature013.wav

  Hit of the ball: Comical drip sound. Retrieved from
  http://www.audiomicro.com/tracks/dialog/45128

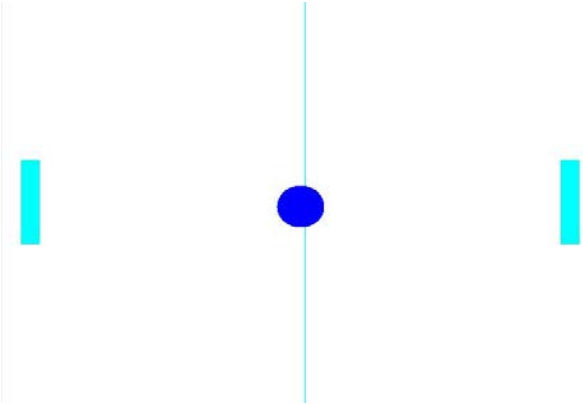
  Goal: Drive warning horn. Retrieved from
  http://static1.grsites.com/archive/sounds/nautical/nautical022.wav

3.5.3. Graphics Effects

The first graphic improvement executed is to replace the rectangle and the circle that represent respectively the pallets and the ball of the pong game with images; moreover the white background of the game panel is changed to an image more attractive for the users. These changes assume a lost in game performance because the representation of an image is slower than a figure. This effect is evaluated in terms of performance and gaming traffic.

Below the different versions of the game are showed.

Initial version
The initial version of the game is designed using the colour blue because the preliminary approach was conceived with the idea of producing the sensation of security and calm. Therefore, the blue colour seems the most adequate for this task.



Screenshot 1: Graphic design. Initial.

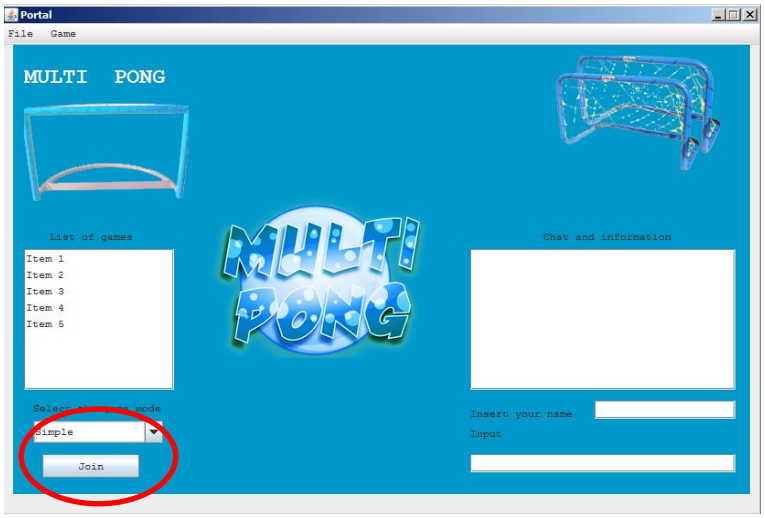
Improved version 1:
The first changed in the game was executed with the intention of connect the pong game with the football style for the purpose of attracting more players due to the popularity of this sport.

Screenshot 2: Graphic design. Football.

Improved version 2:
However, in order to not discriminate any of the players' preferences, other design is produced fulfilling the first idea conceived in the initial version: To produce the security and calm sensation through a water design.

Screenshot 3: Graphic design. Water.

Finally, the portal which appears in all the versions of the Multi Pong game is the following:



Screenshot 4: Graphic design. Game portal.

The font chosen for the portal is Monospaced 13 in order to follow a classical but contemporary style which relax and enjoy to players. Finally, it is remarkable the addition of the input field on the lower left corner of the screenshot 1.3 that allows to each user the selection of a game mode for the online pong. Therefore, the players can play with their favourite game design independently the rest of players.

3.5.4. Others - P2P architecture -

Briefly discussing the previous three improvements implemented in the game, it could be argued that the majority of the gaming aspects have been covered and evaluated; nevertheless all the versions use the same architecture. This is the major argument for the development of a new game using the architecture P2P.

The P2P architecture involves that the clients are communicated directly with each other. The approach selected consists in a server acting as intermediary between the clients through the transmission of the IP address and ports of each player and without interfering in the game course.

The game panel of the server whether or not the number of player is completed closes in thirty seconds; whereas the player who initiated the game would be sending the position of the ball, the updates and its position to the other player through UDP protocol due to the simplicity of this protocol. Moreover, the players' machine usually does not have higher rate than servers. Therefore the UDP protocol seems the adequate choice for this game.

The close of the game in the player part is executed in a similar way to the previous versions, there are two possibilities; firstly the game is always close at the end of the game duration chosen in the game creation. However, whereas one of the players closes the game before time the game is also closed.

Below the P2P architecture flow is represented.

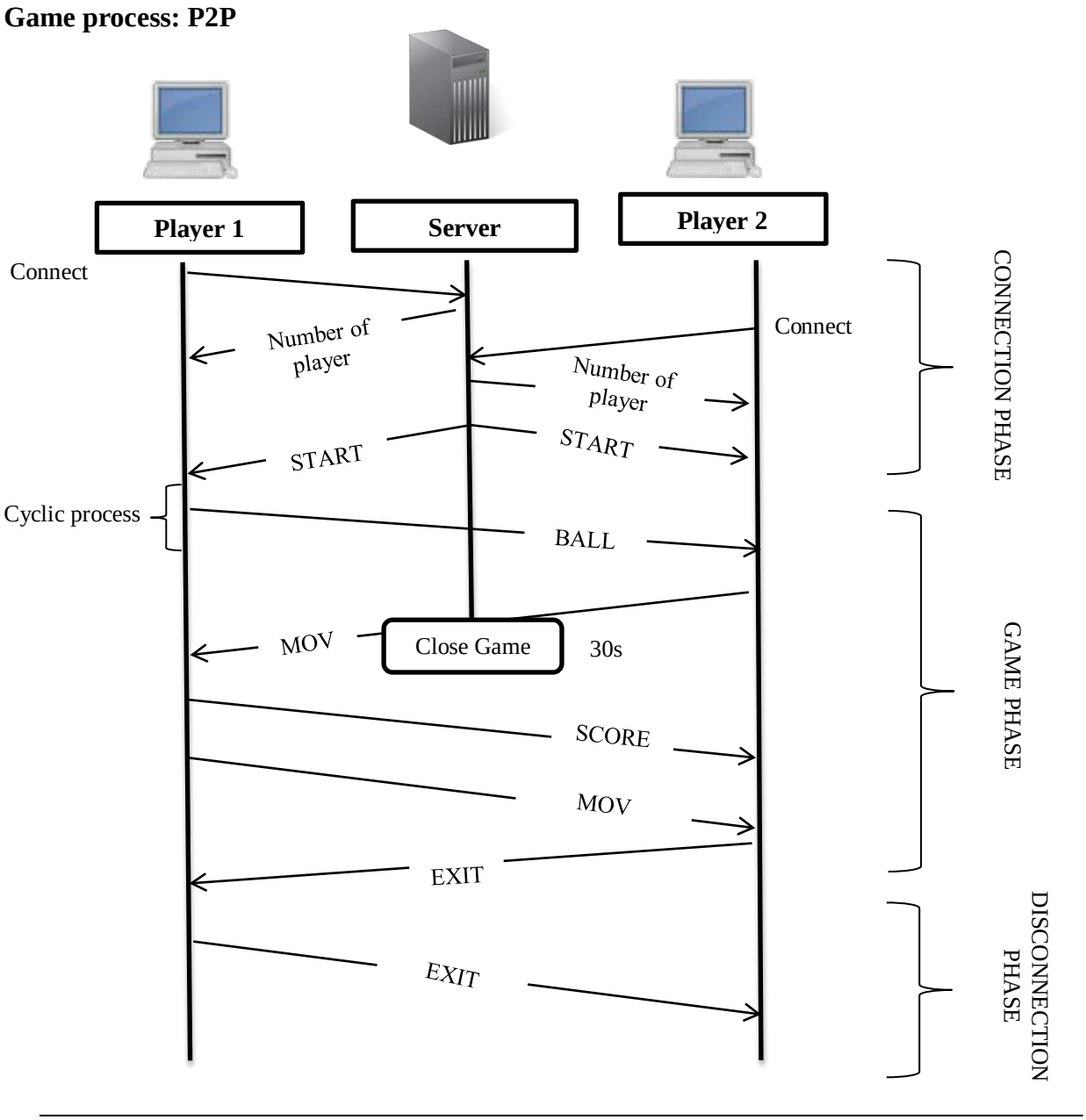


Figure 14: Improvements P2P. Game Process.

Finally, the development process provides as final result three game versions classified by the game connectivity and architecture:

Table 10: Game versions

Version	Game Protocol	Networking Architecture
ClientLabv1	TCP	Client - Server
ClientLabv2	UDP	Client - Server
ClientLabv3	UDP	P2P

Furthermore, each one of these versions involves two game modes. The different modes of game are reduced to two, the simple and the complete mode in order to simplify the evaluation and focus on the network parameters.

- Mode one: Simple.
  The objective of this game is to evaluate the simple traffic and compare it with the traffic performance of the rest of modes.

- Mode two: Complete.
  This mode involves the graphic effects as well as the sound effects. Moreover, this mode implicates two different options:
  - Option 1: Football design
  - Option 2: Water design

The evaluation of the above versions and modes are detailed in the next section.

This page intentionally left blank

4. GAME TRAFFIC ANALYSIS

This section is the central part of the project which involves the evaluation and analysis of the software deployed during the development process; where in it is indicated some strengths and weaknesses related with the implementation of the software. This section reinforces the previous arguments and new considerations are added to the comparison through the study of the traffic and the performance of the network.

4.1. Introduction

The analysis process evaluates three strongly related subjects through the combination of two analytical approaches. In addition, the study is divided into three strongly related subjects:

- Broadband Technology:
  - Ethernet
  - Wireless

- Techniques used in the communication
  - Protocols
    - UDP
    - TCP
  - Client – Server Architecture
    - Client / Server
    - P2P

- Game features
  - Number of players – Different games -
  - Ball speed
  - Sounds effects
  - Graphic effects

- Others
  - Type of networks – LAN / WAN

Two analytical approaches are used in order to evaluate the previous elements; one is focused on technical parameters such as the bandwidth or the delay of the traffic. In contrast, the other approach is focused on assessing the performance to maintain a high Quality of Service – QoS – for the users' satisfaction. The two approaches are summarized below:

- Technical analysis

- Quality of Server (QoS) analysis

As it was commented in the background of this report – section two – through the citation of Greg Costikyan, which appears in the book of Lecky-Thompson (2008, p.44), the customer experience is essential to the success or failure of the games. Therefore, the evaluation of the clients' preferences seems comprehensible in addition to the traffic analysis.

Following the methodology which is executed in each approach is detailed.

4.2. Methodology of the Game Traffic Analysis

The evaluation process of each game is very different. It depends on each feature in which has to be analysed. Firstly, it is indispensable to deploy and to attract players who want to test the game and to have a good time playing to an online game between friends. The process of acquisition of players is performed through the social network known as "Facebook" which allow the communication in real-time by chat or in non-real-time by messages, in addition to add information such as java applications in the messages; thus the synchronization of the players and the game transmission are relatively simple. Still, the addition of the java application in the Facebook messages is a suitable option to transmit the game, the public directories of Dropbox are the option chosen to store and update the game; hence the Facebook messages only contains a link to the application downloaded directly to the players' computer and the instructions to run the game in the different operating systems.

The election of Facebook as the tool to effectuate the experiment planning is also due to the enormously successful of this social network globally. Hence, Facebook provides to this project a dynamism which can be directly translated into save time in the coordination of the players and the tests of the games.

The synchronization process starts in order to initiate some of the players with the Wireshark program run in the background of the system and to capture all the packets when the clients have already received the game. A survey is distributed between the players for the purpose of completing the QoS analysis after the finalization of the game, and then evaluating the technical parameters from the users' perspective.

The survey which is distributed at the end of the game between players can be examined in the appendix H of this document. On the other hand, the methodology used in the Technical analysis and the relation between the subjects indicated are shown below.

The frames captured in client and server are analysed following the indications exposed in the document of Ratti et al. (2010) which divides the process of Gaming Traffic evaluation into three basic phases:

- Data capture: The packets are captured with Wireshark.
- Data scrubbing. The information not relevant is cleaned through filters.
- Analysis and modeling. The technical measures are applied.

The data captured can differ widely according to several parameters, some of the parameters more highlights are the following:

- Game Type: The Pong game is an arcade game. It is an important consideration because this category of games involves continuous updating of the clients and high performance in the packet transmission. Therefore, the traffic captured usually is higher in number of packets and smaller in length than the traffic of the rest of games.

- Game version: Three versions of Pong game have been developed. The three versions are identically with the exception of connection. – See table 10 – The reason of this change is to evaluate the network architecture and protocols.

- Technical environment: The technical environment condition the quality of the game. This project divides into three types the technical environment:
  - The access technology to the network
    - Ethernet / Wireless
  - The type of server:
    - Dedicated host / Run by one of the clients machines
  - The type of network
    - LAN / WAN

- Game configuration: The game configuration depends mainly on two characteristic mainly. The network connection which was showed through the different game versions implemented – TCP, UDP, P2P – and the ball speed that is proportionate to the number of packets sends for updating the ball –packet rate–.

- Match configuration: The pong game shows three different configuration which can be summarized in only two. The simplex mode and the complex mode. The differences between them are principally the addition of sounds and graphics on the game. Nevertheless, these features can decrease the game performance and increase the number of players. Hence, it is important to take the gaming features into consideration during the analysis.

Once the traffic was captured through the packet sniffer, this software also provides the tools to filter and analyse the traffic through filters and statistics tables. These two processes are changed depending on the results obtained. However, the measures do not change significantly. Below is showed the measure used to evaluate the game:

- Bandwidth [kbps]
- Throughput [Bytes/s]
- Packet rate [PPS]
- Delay [ms]

The bandwidth "is the theoretical capacity of one or more network devices or communications links in the system" (McCabe, 2003, p. 41). This measure is calculated through the number of bytes transmitted and the time in which these bytes were transmitted. However, the capacity to transfer information on the system is limited and network traffic grows over time; hence the identification of whom or what is using the available bandwidth may be advantageous for the game management. This task can be executed through the sniffer selected. (Sanders, 2007, p.2) The bandwidth is usually measured in Kbps or Mb/s; however Wireshark use the [Bytes/s] or MB/s. Therefore it is necessary to make the following conversion:

Kbps = (Bytes/s * 8) / 1024

Kbsp = MB/s * 8 * 1024

Bandwidth = Information / Time of transmission.

The throughput "is the realizable capacity of the system or its network devices" (McCabe, 2003, p. 41). Wireshark provides two different tools to measure the throughput. Firstly, Sanders (2007, p.79) claims the use of Wireshark IO Graphs to represent the throughput using packets, bytes or bits sent per second. This graph allows analysing the type of information transmitted and comparing simultaneous data streams. However, this project uses this chart to determine inconsistences in the packet rate, in other words the number of packets send per second [PPS]. Secondly, and only when the protocol is TCP, Wireshark allow the representation of the throughput measured in Bytes per second (B/s). McCabe (2003, p.35) shows the following figure in order to clarify the behaviour of the throughput and the retransmission of TCP packets.

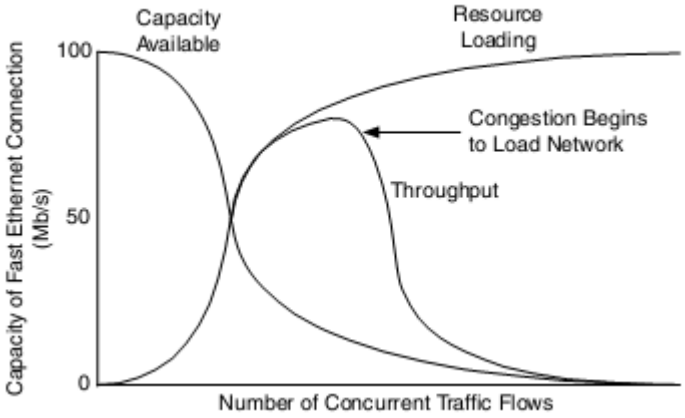


Figure 15: The Performance of a Fast Ethernet Connection under Best-Effort Conditions.

Retrieved from (McCabe, 2003, p.35)

Whether the packet loss is high, the throughput performance is decreased due to the congestion, being able to approach even zero. It is the main cause for the careful study of this characteristic in the TCP protocol because the UDP protocol does not have retransmission of packets; therefore the network performance is not affected by the packet loss rate. Similarly, the delay is focus on the TCP protocol because the major delays in the network are caused by the retransmission of packets. As it was mentioned in the background the delay is the time used in the transmission of information. This feature can be measured using the round trip time –RTT– which is the elapsed time since a packet is sent until the verification of the packet is received.

Wireshark also provides for TCP protocol the possibility to plot these times in a chart for the proper comparison (Sanders, 2007, p.72). McCabe (2003, p.42) proposes other alternative to measure the delay in both directions – RTT- using the "practical and universally available utility" ping; in addition McCabe also describes other way to measure the delay in one direction (end-to-end). However this measure is not compatible with the utility ping.

The previous measurements are the main measures to evaluate the three subjects mentioned in the introduction of this section. However, the game traffic patterns are also necessary to calculate the measures.

- Number of packets [number]
- Packet loss [percent]
- Packet length [bytes]
- IAT [ms]

The number of packets sent by the server is an essential data to calculate several parameters, including the packet loss. The packet loss may be caused by failures in access technology, type of protocol and architecture, or even performance of the HW/SW resource. The real-time applications such as arcade online games vary in the amount of packet loss; hence the real-time requirements are difficult to meet in an environment with variable queuing delays and congestion losses (Stalling, 2011, p.654 - 655). In the case of the Pong game, the packet loss can reduce the quality of service in the clients' machine. Consequently, it will be a decisive factor in the game evaluation.

The packet interarrival time –IAT– is the time between two consecutives packets from client to server or vice versa; and the packet length is the size on bytes of the information transmitted. Both measures are conditioned by the MTU– Maximum Transmission Unit – which usually is 1,500 bytes (Sanders, 2007, p.95). Regarding to the Pong game, this parameter does not affect the gaming traffic due to the small size of the gaming packets. However, these measures also are directly proportional to the delay and bandwidth. Therefore, the study and analysis of these measures may help to understand the rest of measures. The IAT depends mainly on the packet generation rate and hardware performance (Ratti et al., 2010, p.65); therefore the evaluation of this measure may be used to evaluate the clients' hardware.

Although Wireshark is an invaluable tool for the satisfaction of this project, it is not the only tool used; in addition the following utility is used:

C programs provided by the University of Portsmouth, specifically the Dr Savage.

- C++ code for building a histogram of interarrival times. Retrieved from: http://savagen.myweb.port.ac.uk/Histogram_arrival.cpp
- C++ code for building a histogram of packet sizes. Retrieved from: http://savagen.myweb.port.ac.uk/Histogram_size.cpp

4.3. Technical Analysis

Finally, the experiments are designed in order to select the best design for the games developed after the planning of the parameters and the subjects which will be analysed. The experiments are composed of tests in order to simplify the analysis. The test are scheduled depending on the next features.

- Players. The number of players
- Duration. The duration of the games
- Versions. The game version that are evaluated.
- Network environment. The location of the network where the game is run.
- Captures. The host which its traffic is captured.
- Host Identification. This section exposes the devices which are used in the traffic capture; and the network technology which use these devices to connect to the Network.

Below the experiments are listed and selected in order to test in each experiment one of the subject identified in the introduction of this section.

Figure 16: Experiments summary.

EXPERIMENT	TESTS	Subject evaluated
Experiment 1	**Test 1**	TCP & UDP protocols
Experiment 2	**Test 2 and 3**	Game Features
Experiment 3	**Test 4 and 5**	Networking architecture
Experiment 4	**Test 6**	Type of Networks

4.3.1. First Experiment

The first experiment has the objective of understand the behaviour of the game in online mode through two different protocols such are TCP and UDP. This is also useful to plan the following tests and understand the measures selected previously.

The hardware used throughout the entire tests is usually the devices described in the hardware resources, except where it is not possible due to technical causes. In addition, the majority of the tests are focused on the two first versions – TCP and UDP- because these versions generate more traffic in the server environment. Furthermore, the client captures provides information for the evaluation of the access technologies, using wireless connexion in the client one –laptop– and Ethernet connection in the client two – chipmunk –. Therefore, the server captures together with the client captures provide all the information necessary for the accuracy analysis of the TCP and UDP protocols.

The duration of the game is five minutes by default in order to establish a common measure for all the captures.

Table 11: Test 1. Resume.

TEST 1: simple test UDP & TCP	
Players	2
Duration	5 min
Versions	TCP & UDP
Network environment	University of Portsmouth
Captures	All devices
Server	Project1 (dedicated)
Network Technology	Ethernet
Player 1	laptop
Network Technology	Wi-Fi
Player 2	chipmunk
Network Technology	Ethernet

The information about the trace is described before that the traffic is measured in order to be in possession of all the necessary data for understanding the results. The test one results six traces of packets with the following features:

Table 12: Test 1. Network Trace Information

Network Trace Information	TCP	UDP
Server Host	148.197.34.67	148.197.34.67
Start Time	04/05/2012 13:03:24	04/05/2012 13:48:25
Stop Time	04/05/2012 13:13:24	04/05/2012 13:58:25
Total Time of Trace	00:10:00 (600.311s)	00:10:00 (600.304s)
Nº Games activated	1	1
Nº users in the system	2	2
Client - laptop -	148.197.67.100	148.197.67.100
Start Time	04/05/2012 12:51:13	04/05/2012 13:37:22
Stop Time	04/05/2012 12:57:59	04/05/2012 13:43:53
Total Time of Trace	00:06:46 (406.605s)	00:06:31 (391.590s)
Game Mode 1	Simplex	Simplex
Client - chipmunk -	148.197.27.14	148.197.27.14
Start Time	04/05/2012 12:50:52	04/05/2012 13:36:50
Stop Time	04/05/2012 12:57:51	04/05/2012 13:43:51
Total Time of Trace	00:06.58 (418.343s)	00:07:00 (420.975s)
Game Mode 1	Simplex	Simplex

The difference in time between initiation of clients and server may be due to the fact that the server time is ahead twelve minutes. However, this fact does not affect the results. Below the table 13 which presents the network trace statistics provides a general vision about the traffic of each host; hence depending on the table results, the subjects are analysed in order to clarify the traces and obtain logical conclusions. In particular, this table shows the first Wireshark results obtained via filtering the traffic of the previous IP addresses. Consequently, depending on the results the following tables will be more specific.

Although the data of the table 13 are generic, it is possible to find clear differences between the clients and the versions evaluated. The differences more noticeable are the sudden increase in the traffic of the client "chipmunk" in the TCP version and the difference of packets between the two versions.

Whether the number of packets is examined carefully, it is possible to find inconsistences in the statistics results in which the sum total of the packets received by the clients is not the same as the number of packets transmitted by the server. This difference of packets is extremely concerned in the case of UDP due to the high number of lost packets and the lack of retransmission.

Table 13: Test 1. Trace Network Statistics.

Trace Network Statistics	TCP	UDP
Server Host	148.197.34.67	148.197.34.67
Total Packets	12,358	7,529
I / O	6,074 / 6,284	678 / 6,851
Mean Packet Size	74.011 bytes	56.952 bytes
I / O [bytes]	67.120 / 80.684	60.063 / 56.644
Mean Bandwidth	15.92 Kbps	7.88 Kbps
I / O [Kbps]	7.09 / 8.83	0.75 / 7.40
Mean Packets rate	27.557 PPS	12.560 PPS
I / O [PPS]	13.564 / 14.014	1.594 / 16.730
Client - laptop -	148.197.67.100	148.197.67.100
Total Packets	3,634	3,638
I / O	1,891 / 1,743	3,402 / 236
Mean Packet Size	73.125 bytes	59.663 bytes
I / O [bytes]	82.703 / 62.733	60.014 / 54.602
Mean Bandwidth	5.11 Kbps	4.33 Kbps
I / O [Kbps]	3.01 / 2.10	3.01 / 0.26
Mean Packets rate	8.937 PPS	9.290 PPS
I / O [PPS]	4.651 / 4.287	9.055 / 0.603
Client - chipmunk -	148.197.27.14	148.197.27.14
Total Packets	8,672	3,762
I / O	4,342 / 4,330	3,319 / 433
Mean Packet Size	73.734 bytes	59.540 bytes
I / O [bytes]	80.143 / 67.307	60.022 / 55.930
Mean Bandwidth	11.94 Kbps	4.16 Kbps
I / O [Kbps]	6.58 / 5.44	3.80 / 0.46
Mean Packets rate	20.729 PPS	8.936 PPS
I / O [PPS]	10.505 / 10.350	8.106 / 1.052

These differences are studied in the following sections and compared with the rest of tests. However, firstly the traffic of both servers the UDP and the TCP are examined through the next figures in order to understand the game behaviour. These figures show the number of packets sent per second

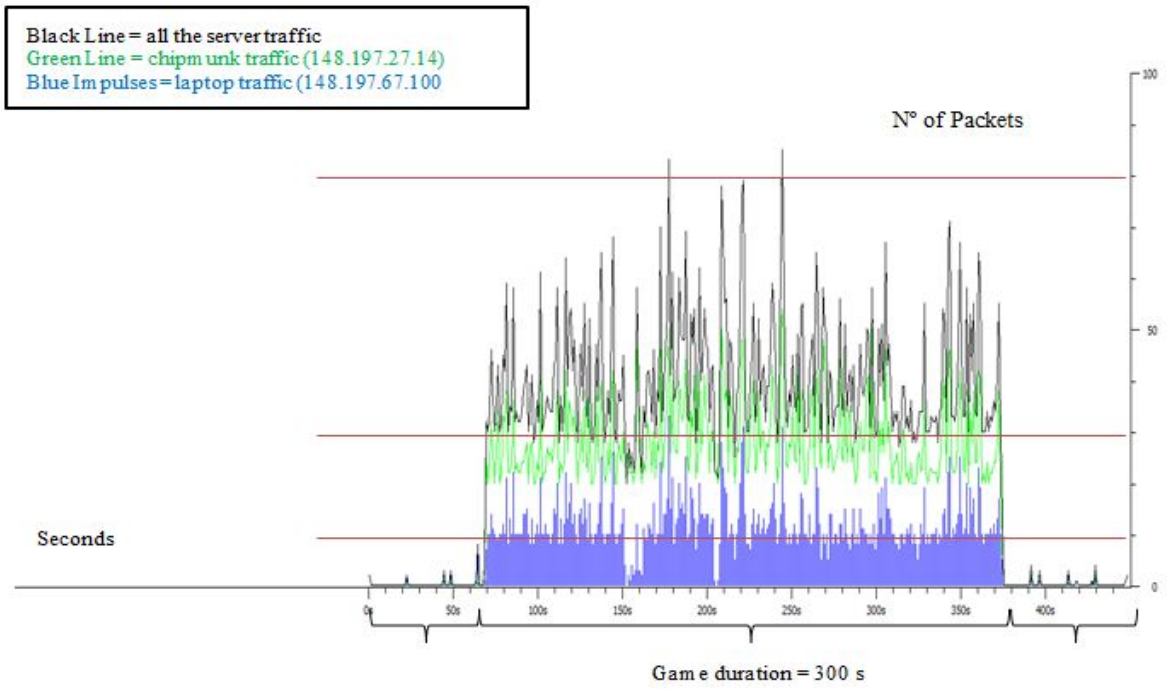


Figure 18: Test 1. Server TCP. Packet rate [PPS]

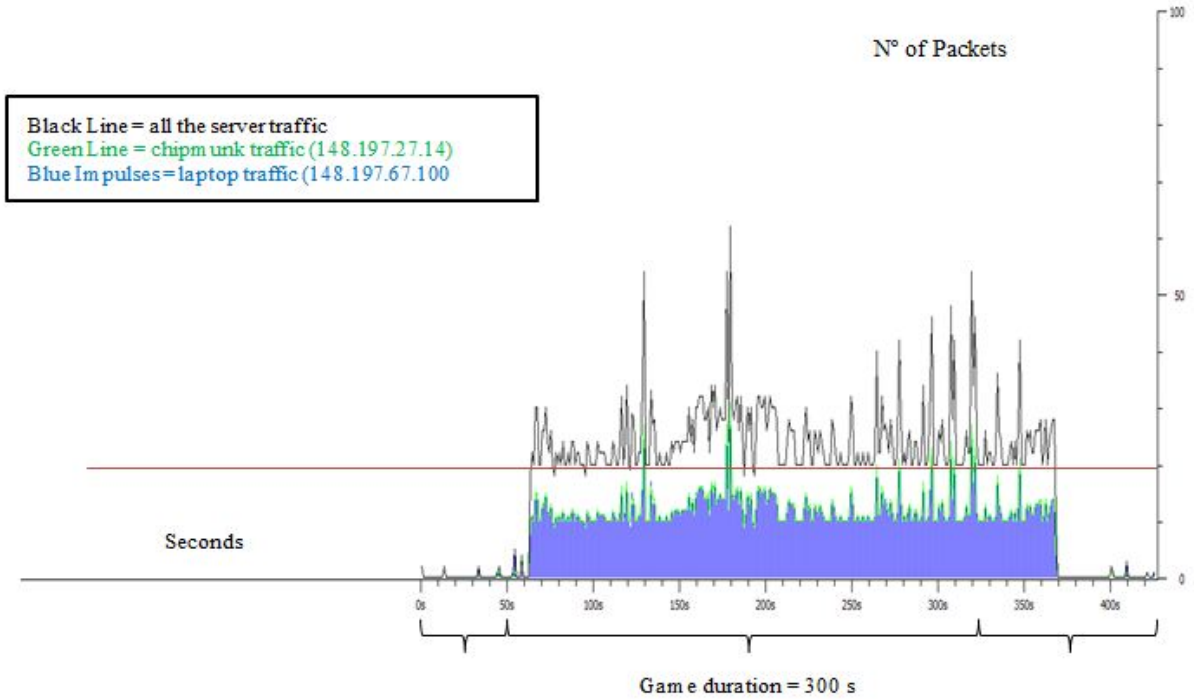


Figure 17: Test 1. UDP server. Packet rate [PPS]

The packets sent by the server to the client laptop which uses wireless are represented in blue and the packets sent to the client chipmunk in green which in addition to use Ethernet, has a processor more powerful and faster than the laptop client. These features may be characteristics that can explain the performance of the figures 18 and 17. Specifically, it is important to understand the performance of the version TCP because the chipmunk client consumes twice of traffic than the laptop client. However, only with the previous representation is complicated to understand.

In contrast with the TCP representation, the UDP representation is more stable and equitable respect to the packet flow. The number of packets in the UDP server is between 10 and 20 PPS with peaks of 50 PPS; whereas the number of packets in the TCP server is 40 and 50 with peaks of 80 PPS. The traffic transmitted between server and clients is limited to the position of the ball, which is update in this capture to a speed of 100ms (10PPS); the movement of the users; and the score update. The last two traffic causes can be considered as random. Thus, this data may cause some of the peaks in the server traffic. However, the most noticeable aspect, as it was mentioned, is the chipmunk traffic. Therefore, the cause of underflow in the laptop client may be due to the differences between TCP and UDP. Consequently, the type of traffic transmitted in the server is exposed below.

Table 14: Test 1. Server Protocol Hierarchy Statistics

Versions	TCP			UDP		
Protocol	**% Packets**	**Packets**	**Bytes**	**% Packets**	**Packets**	**Bytes**
IP v4	100%	12,358	913,792	100%	7,529	428,794
User Datagram Protocol	0.22%	27	1,775	100.00%	7529	428,794
Data UDP	0.22%	27	1,775	100.00%	7,529	428,794
Transmission Control Protocol	99.78%	12,331	912,017	-	-	-
Data TCP	54.68%	6,757	553,447	-	-	-

The above table shows that the TCP version consumes more packets and bytes than UDP. Furthermore, the table 14 allows to calculate the real bandwidth consume during the game process; as the mean bandwidth indicated in the table 13 is calculated with all the traffic captured; and as it can be observed in the figures 18 and 19, the game process only involves 300 seconds.

Consequently, the bandwidth of the TCP version is 23.75 Kbps and the bandwidth of the UDP version is around 11.16Kbps. Comparing this traffic with the previous publications which analysed important games with 800Kbps (Ratti et al, 2010), this bandwidth is insignificantly. This difference is mainly due to the number of players and resources which have the others games.

However, the data of last table does not show the real cause of the higher traffic on the chipmunk client; in addition the UDP traffic should be higher than the TCP traffic due to the connection establishment of TCP protocol. As, the TCP header is smaller than UDP header, because the TCP header does no need connection information (IP address source or destination, ports). Therefore the network traffic interchange between server and clients is analysed in order to understand the gaming performance in the TCP version.

The packet flow graphs of the two versions are shown:

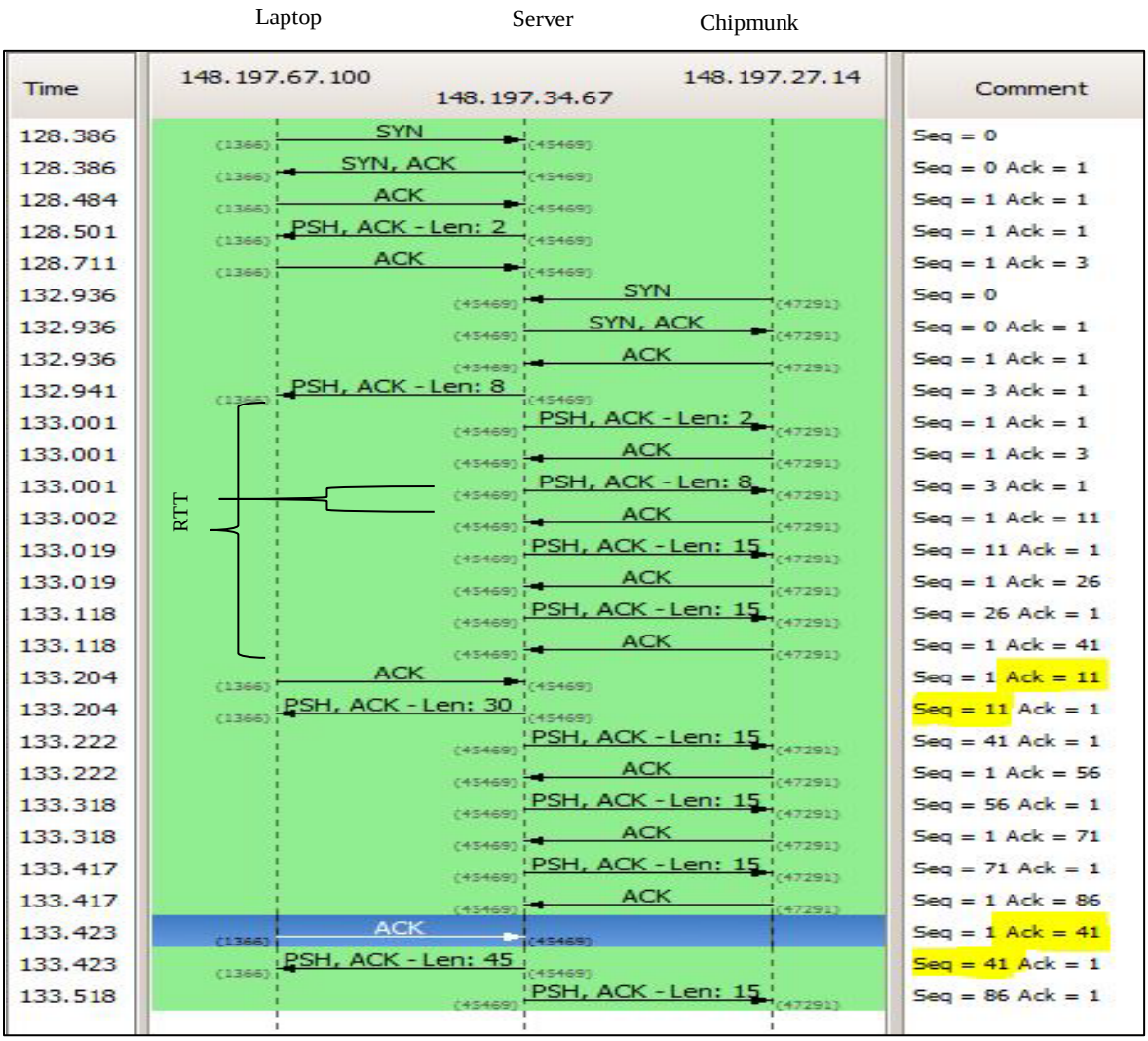


Figure 19: Test 1. TCP Server. Flow Graph

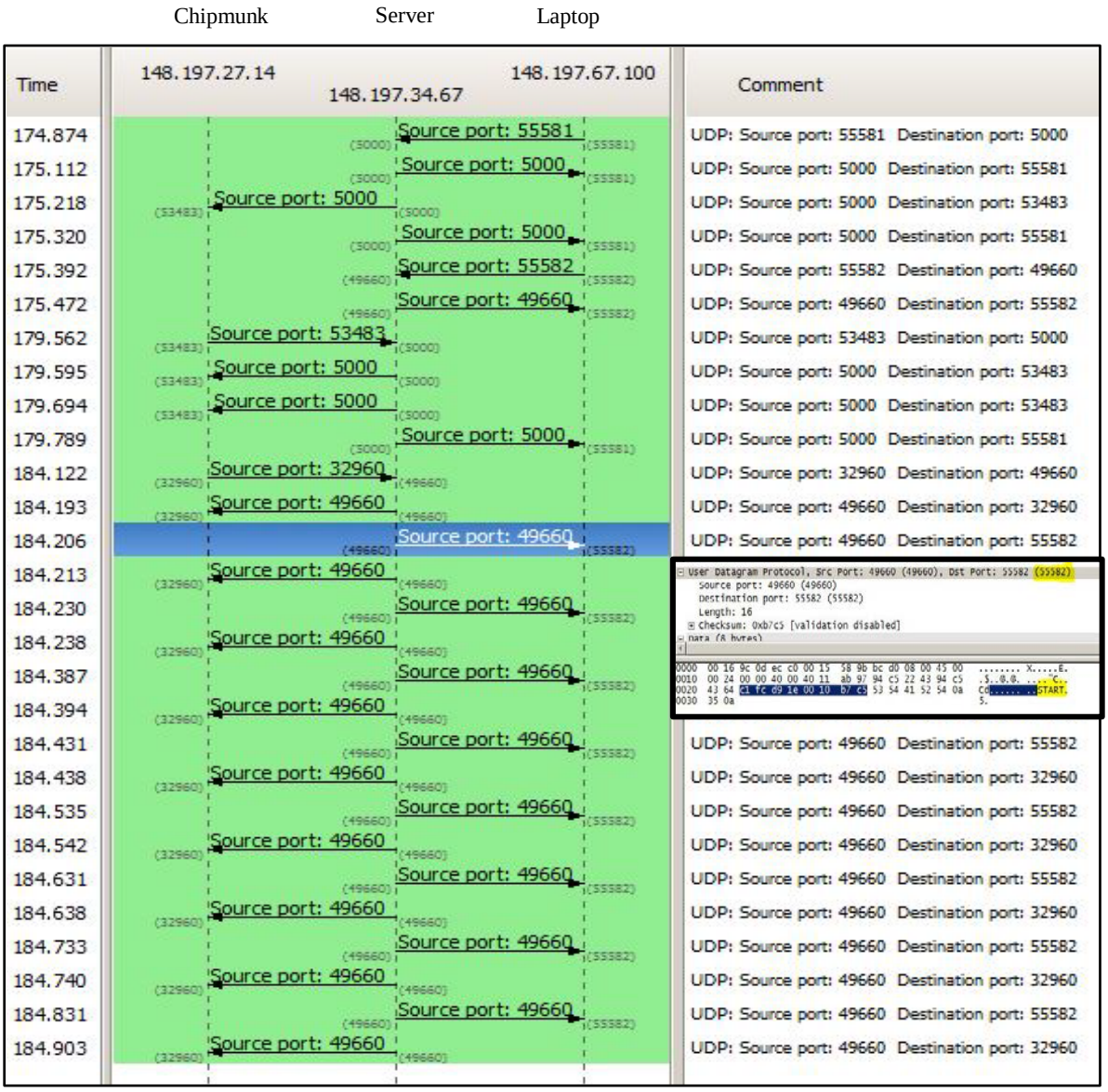


Figure 20: Test 1. UDP Server. Flow Graph

The previous figures reveal the cause of the high traffic in the chipmunk client. The Packet rate in the TCP version is different in each client because the acknowledgment time is different in each client. This causes a high variation in packet length because the TCP protocol is designer to send all the data generated; thus the packets contain the previous information grouped as a single packet. However, the arcade games do not require the delayed packets delivery. As a possible solution to the TCP situation, the management of packet acknowledgment and the improvement in the game synchronization may help to reduce the wide difference between the RTT times. Moreover, this project does not have the sufficient time to make the necessary adjustment; hence these improvements may be executed in future game versions or projects.

In contrast, the UDP communication provides a stable service sending packets periodically without retransmission of last information of loss packets. The loss packet rate also affects the game quality in the user part. Nevertheless, the server packet length is firstly evaluated for the purpose of assessing the gravity of the last issue.

The following tables provide a simplified representation of the data shows in the figures 21 and 22, which also show the relation to the clients.

Table 15: Test 1. Total packet Lengths in the servers

Packets Lengths	**v1. TCP Traffic -Server-**			**v2. UDP Traffic -Server-**		
	count	Rate	Percent	count	Rate	Percent
	12331	0.008996	-	7,529	0.017699	-
0 - 19	0	0.00000	0.00%	0	0.000000	0.00%
20 - 39	0	0.00000	0.00%	0	0.000000	0.00%
40 - 79	6993	0.018214	57.71%	7,529	0.017699	64.86%
80 - 15	5274	0.000143	42.77%	0	0.000000	0.00%
169 - 319	55	0.000143	0.45%	0	0.000000	0.00%
320 - 639	7	0.000000	0.06%	0	0.000000	0.00%
640 - 1279	2	0.000005	0.02%	0	0.000000	0.00%
1280 - 2559	0	0.000000	0.00%	0	0.000000	0.00%

Table 16: Test 1. Statistics of the Packet Length in the servers

	TCP			**UDP**		
Size in Bytes	All	Laptop	Chipmunk	All	Laptop	Chipmunk
MAX	934	934	453	74	74	74
min	54	54	55	47	47	47
Mean	74.01	74.43	73.83	56.95	56.93	56.96
Standard Deviation	18.49	30.25	9.98	1.51	1.44	1.59
Variance	342.11	915.39	99.60	2.30	2.09	2.52

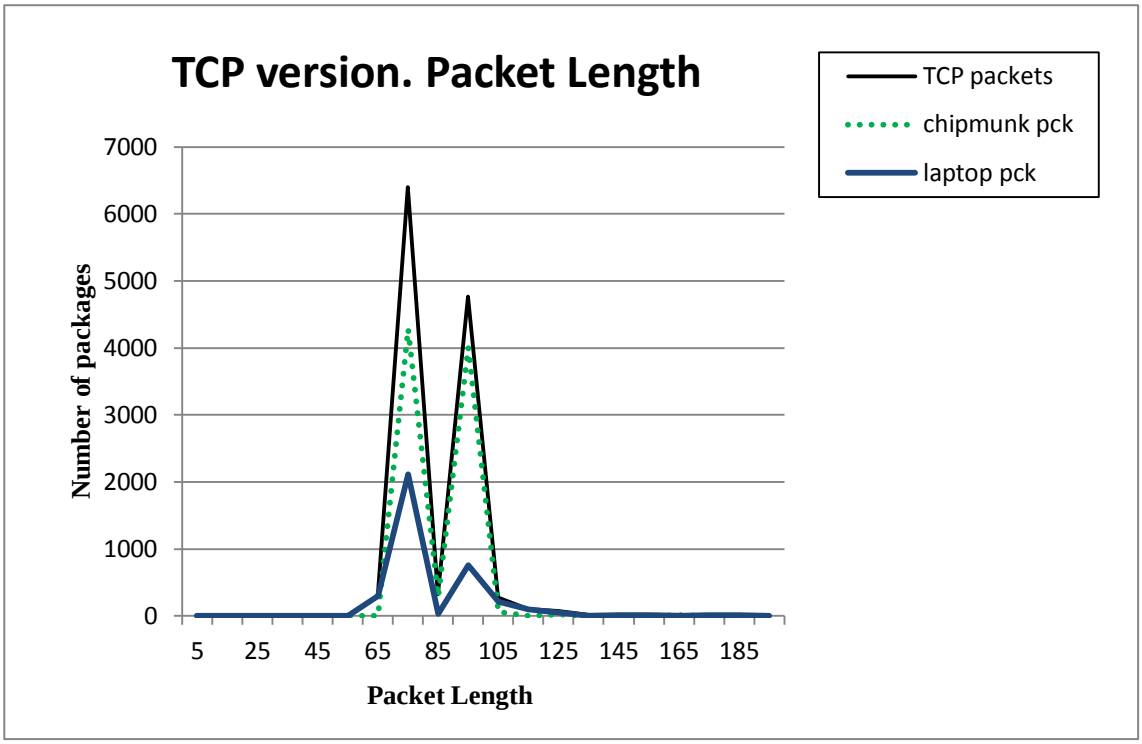


Figure 22: Test 1. TCP version. Packet Lengths

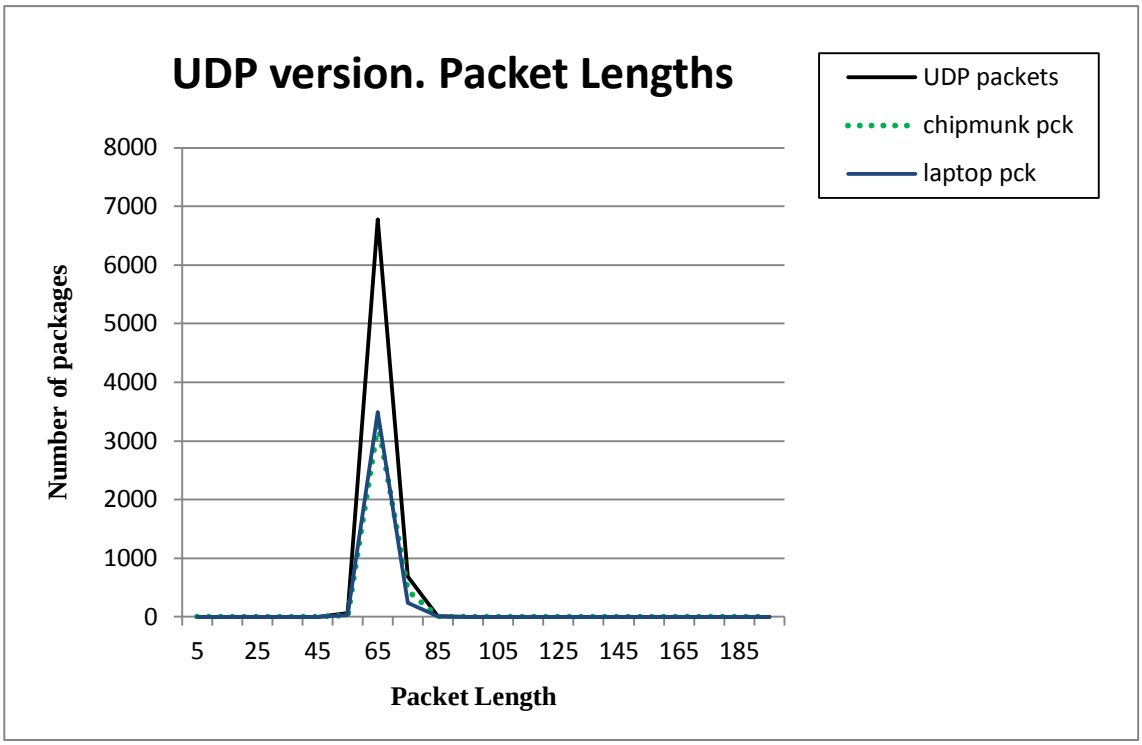


Figure 21: Test 1. UDP version. Packet Lengths

The previous data reinforce the previous arguments and locate the UDP protocol as the favourite for the Pong game implementation; due to the constant stream and equitable flow in the packet sizes. All the packets are within the range of 40 – 79 bytes and both clients receive the same amount of packets with the same average on packet size. On the contrary, the protocol TCP shows an irregular performance which corroborates the last theory about the packet size increased when the client acknowledgment is delayed. This increase in the packet size presents a max of 934 bytes which can be observer in the figure 22. However, the size average is also 74 bytes; although there is other peak in 95 bytes which indicates that a higher percentage of acknowledgment packets is delayed; and in this way increased the size of the following packets.

The largest packet; and thus the worst situation are showed in the following figure. This case, as it was mentioned before, is due to the delay of packet acknowledgment by clients. Therefore, the information is accumulated in the output stream; consequently the server sends packets with a big amount of information which is not useful for clients because most of the information is delayed.

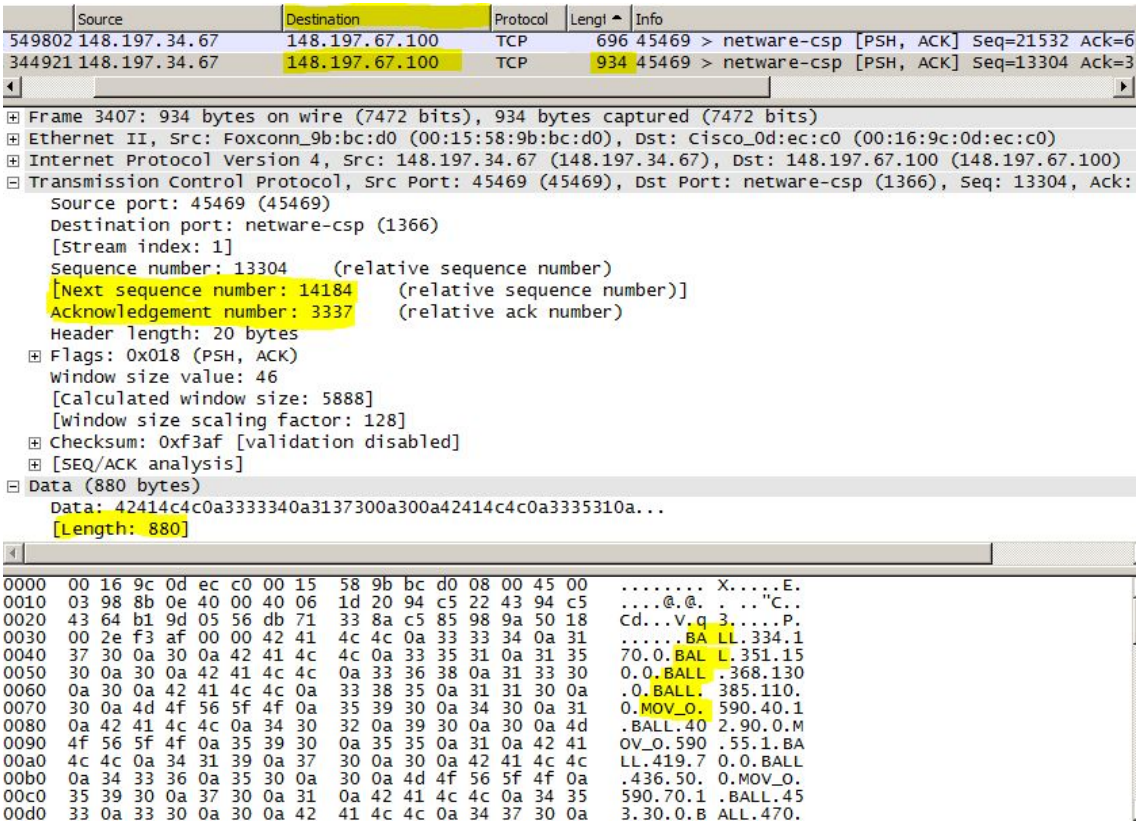


Figure 23: Test 1. TCP server. The largest packet

To conclude the analysis of the servers in the first experiment, the delay and throughput of the packets sent from the server to clients are studied in order to gain a better understanding about the last issues and to find possible solutions to this problems. As, UDP does not present issues in the packet transmission, it is not consider for the following analysis.

The throughput is measured in Bytes/s and the delay in seconds through the RTT tool which Wireshark provides for the TCP protocol. Firstly, the delay is showed in the figure 25 in which it is possible to compare the delay in the packet transmission toward the laptop client (on the right) and chipmunk client (on the left) where the differences are immense.

The red line reflects the limit of RTT time in the chipmunk client (0.04s) in both figures, which in particular this line is at the bottom in the figure of the laptop RTT; and the normal delay of this client which is four times bigger compared to the limit of the chipmunk RTT (0.2s). These facts explain the problems in the reception of packet of the laptop client.

The figure 26 shows the output throughput of the server toward both clients, where in the case of the client laptop the violent fluctuations suggest that the packet loss rate is high and the traffic of this client is congested. However, these fluctuations may be due to the high variation in the packet size. The packet loss in the clients is studied in order to clarify these questions.

Furthermore, a possible cause of the high delay in the laptop client packets may be the differences between access technologies – Wi-Fi / Ethernet – and processing power on the clients. These facts may also explain the low performance in the game when it is run on the laptop client. Therefore, the last part of the first test is focused on the comparison of client technologies through the analysis of the packet loss and IAT.

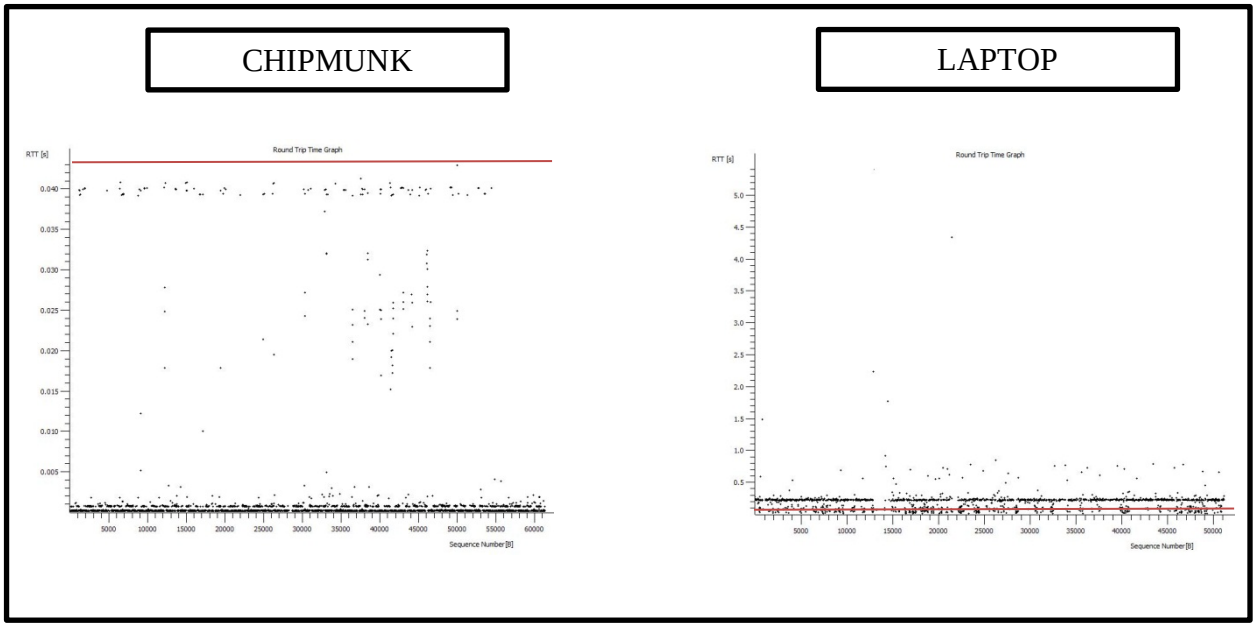


Figure 25: Test 1. TCP Server. RTT

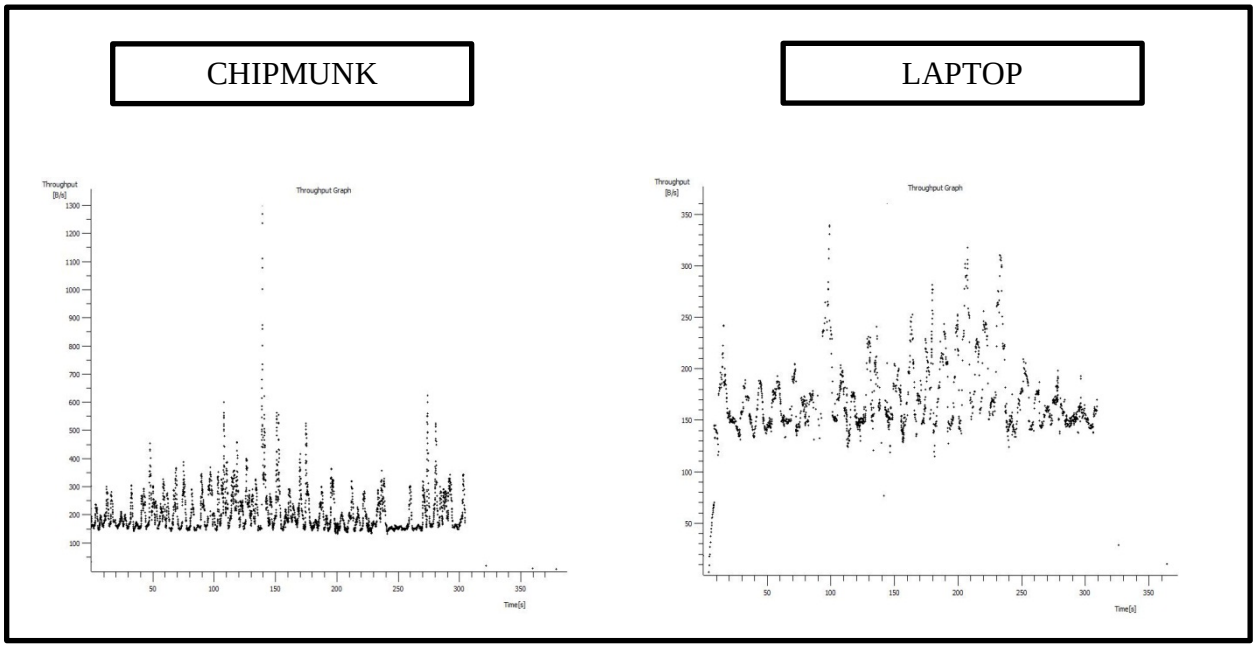


Figure 24: Test 1. TCP Server. Throughput

The following table shows the study about the number of packets sent and received by each device and the difference between them. The packet retransmission also is showed for the case of TCP Protocol. The traffic selected to fulfil the tables is limited only to the traffic belonging to the game process. The traffic was also filtered through the TCP protocol and the devices' IP addresses in the TCP version and through the UDP ports and IP address of the devices in the UDP version (Server = 49660; Laptop = 55582; Chipmunk = 32960).

List of Packet s	**TCP**		**UDP**	
Devices	Laptop	Chipmunk	Laptop	Chipmunk
Packets sent -Server-	1,930	4,338	3,522	3,314
Packets received -Clients -	1883	4338	3,396	3,311
Difference	47	0	126	3
Packets sent -Clients-	1738	4330	231	437
Packets received -Server -	1733	4330	230	437
Difference	5	0	1	0
TOTAL	52	0	127	3
Percent	1.42%	0.00%	3.38%	0.08%
Packets retransmitted sent- Client-	20	0	-	-
Packets retransmitted sent- Server-	50	0	-	-

Table 17: Test 1. Evaluation of the packets sent and received.

The next figure shows the percent of loss packets between both clients in the versions UDP and TCP in order to facilitate the comparison between clients' data.

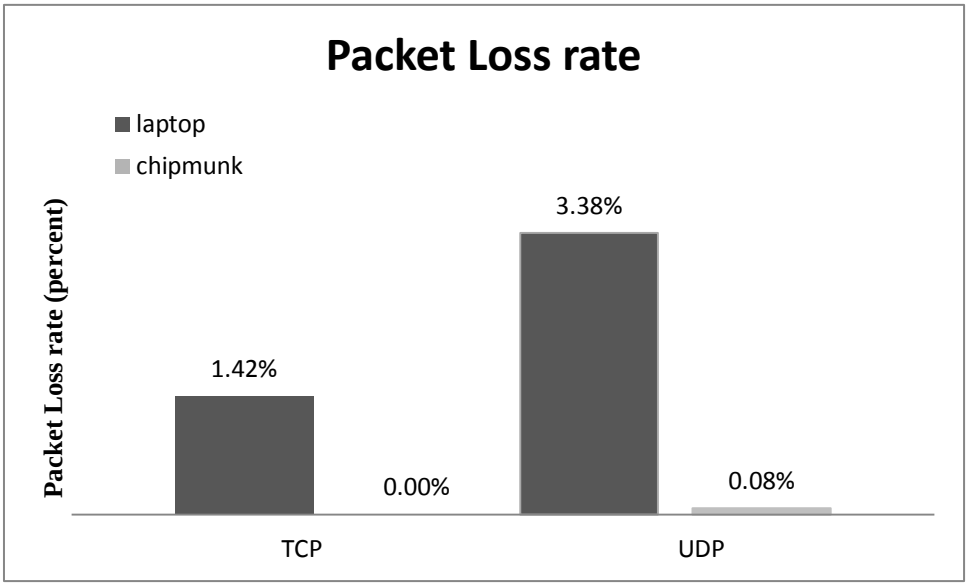


Figure 26: Test 1. Packet Loss rate in the clients

The results of the figure 26 reinforce the superiority and advantage of clients from the A2.3 laboratory against external players which play the Pong game through their laptop. Regarding the network protocols, although the UDP loss packet rate is higher than TCP, this small inconvenience does not affect the game because although the packet loss rate is higher than TCP, the retransmission of packets are not necessary in the arcade games due to the real-time requirements. Therefore, the best technology and protocol to develop the Pong game – arcade game – is through the UDP protocol and the Ethernet connection.

The difference in the loss packet rate between versions is due to the TCP and UDP protocol and the difference between clients is due to the hardware performance. The main cause of the difference between the high packet loss rate of laptop client and chipmunk client is the access technology –Wireless/Ethernet -. However, the RAM and the processor also may influence in clients results. Thus, the analysis of IAT may show the differences between clients' hardware.

Table 18: Test 1. IAT Statistics

IAT Statistics	**TCP**		**UDP**	
Times in ms	Laptop	Chipmunk	Laptop	Chipmunk
MAX I/O	37.95/37.74	16.82/16.78	8.81/22.91	0.66/13.48
min I/O	0.00004/0.0002	0.000046/0.000076	0.000031/0.036	0.000092/0.020
Mean I/O	0.19/0.20	0.074/0.074	0.09/1.36	0.092/0.069
Standard Deviation I/O	0.974/1.003	0.257/0.256	0.177/3.193	0.251/1.945
Variance I/O	0.944/1.006	0.06/0.06	0.031/10.196	0.0006/3.78

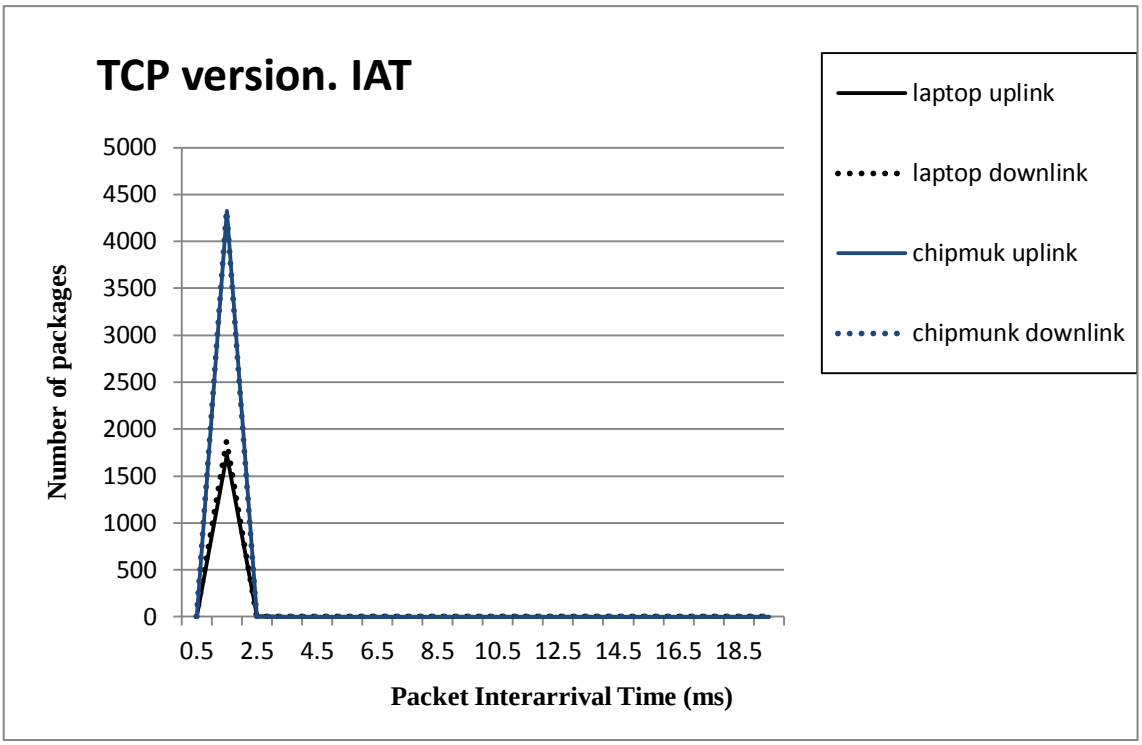


Figure 28: Test 1. TCP version. IAT

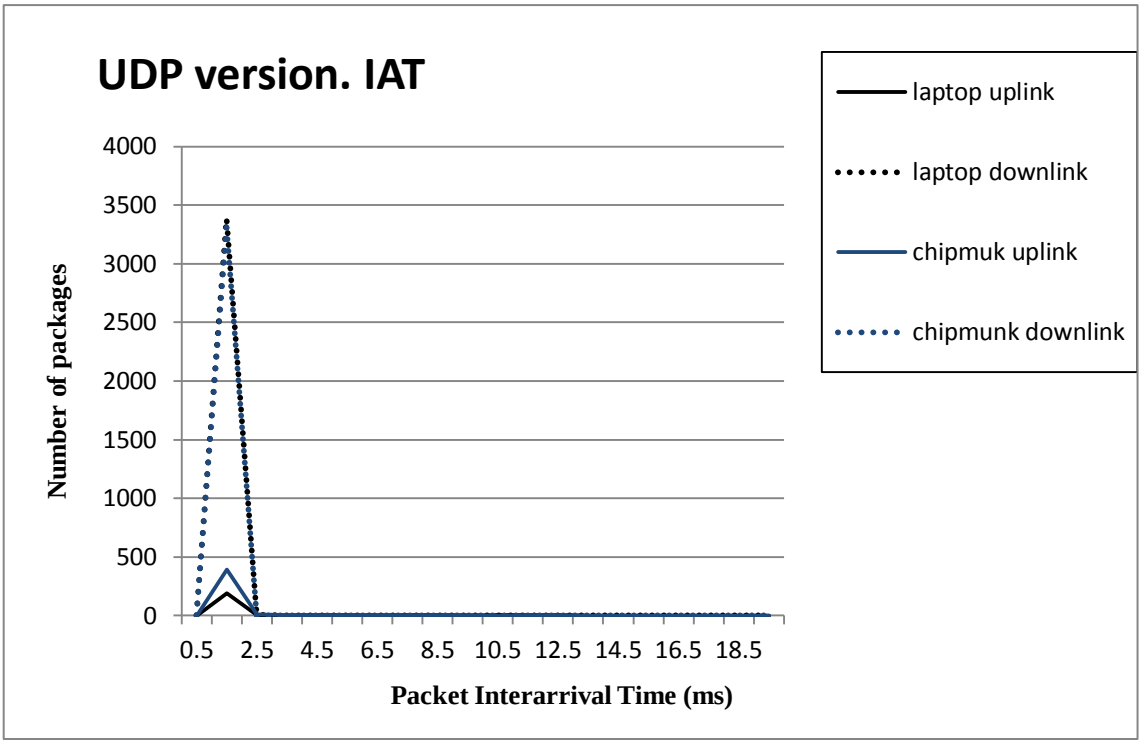


Figure 27: Test 1. UDP version. IAT

The analysis of the IAT in both clients seems that the differences in the clients' hardware do not affect the game network performance; because in both versions and clients the IAT is in the range of 0.5 and 2.5 ms. Therefore, the significant factors which affect the game performance are in first place the networking protocol – TCP/UDP – increasing exponentially the number and size of packets in the case of TCP; and secondly the access technology – Ethernet/Wi-FI– duplicating the loss packet rate in the wireless technology. Also the access technology may be the reason of the delays in the TCP protocol cause the increasing in the size of packets; in order to solve this hypothesis is executed the fourth experiments due to both clients – laptop and chipmunk –belong to different LANs.

The second experiment is designed in order to test the access technology in the previous two versions –TCP and UDP – through the increase of the number of players and packet rate –speed of the ball–; moreover it analyses the different game modes – Graphic/Sounds effects–.

4.3.2. Second Experiment

After the in-depth analysis of the first experiment, this experiment focuses on the client part in order to evaluate the gaming features and effects of access technology.

The second test is designed with the objective of evaluating the graphics and sounds effects implemented in the game throughout the three games detailed in the table 20, in which the last game in addition to evaluate the graphics and sound effects, the packet rate is increased. Thus, the ball speed also increases. Furthermore, the clients are the same than the last experiment in order to gain a better understanding about the differences between access technologies.

In contrast, the third test is designed to evaluate other kind of game features, which is the number of players. The number of players is possibly the most critical factor that contributes to the increase of the network traffic in the server; hence the users' QoS may be reduced if this fact is not suitably managed.

Table 19: Test 2/3. Resume

TEST 2: Gaming features / 3: Player number	
Players	2 / 8
Duration	15 / 5 min
Versions	TCP & UDP
Network environment	University of Portsmouth
Captures	All devices
Server	Project1 (dedicated)
Network Technology	Ethernet
Player 1	laptop
Network Technology	Ethernet
Player 2	chipmunk
Network Technology	Ethernet

The following table details the information of the traces captured such as number of games in the server or kind of games which are played. Moreover, as this experiment contains two tests the number of tables is twice; consequently it is only evaluated the significant parameters which can affect the game features identified.

Table 20: Test 2. Network Trace Information

Network Trace Information (2)	TCP	UDP
Server Host	148.197.34.67	148.197.34.67
Start Time	08/05/2012 16:50	09/05/2012 15:28
Stop Time	08/05/2012 17:10	09/05/2012 15:48
Total Time of Trace	00:20:28 (1228.533s)	00:20:11 (1211.743s)
Nº Games activated	3	3
Nº users in the system	2	2
Ball speed Game 1	100ms -> 10PPS	100ms -> 10PPS
Ball speed Game 2	100ms -> 10PPS	Simplex
Ball speed Game 3	50ms -> 20PPS	50ms -> 20PPS
Client - laptop -	148.197.67.100	148.197.67.100
Start Time	08/05/2012 16:36	09/05/2012 15:13
Stop Time	08/05/2012 17:56	09/05/2012 15:33
Total Time of Trace	00:19:59 (1199.595s)	00:20:11 (1211.590s)
Game Mode 1	Simplex	Simplex
Game Mode 2	Complete Water	Complete Water
Game Mode 3	Complete Water	Complete Water
Client - chipmunk -	148.197.27.14	148.197.27.14
Start Time	08/05/2012 16:37	09/05/2012 15:14
Stop Time	08/05/2012 17:56	09/05/2012 15:33
Total Time of Trace	00:19:02 (1142.379s)	00:19:31 (1171.041s)
Game Mode 1	Complete Water	Complete Water
Game Mode 2	Simplex	Simplex
Game Mode 3	Complete Football	Complete Football

Table 21: Test 3. Network Trace Information

Network Trace Information (3)	TCP	UDP
Server Host	148.197.34.67	148.197.34.67
Start Time	11/05/2012 14:44	11/05/2012 15:04
Stop Time	11/05/2012 14:56	11/05/2012 15:13
Total Time of Trace	00:12:08 (728.213s)	00:09:15 (555.834s)
Nº Games activated	4	4
Nº users in the system	7	6
Ball speed	Standard (10 PPS)	Standard (10 PPS)
Client - laptop -	148.197.67.100	148.197.67.100
Start Time	11/05/2012 14:29	11/05/2012 14:49
Stop Time	11/05/2012 14:38	11/05/2012 14:56
Total Time of Trace	00:08:18 (498.693s)	00:06:34 (394.632s)

Client - chipmunk -	148.197.27.14	148.197.27.14
Start Time	11/05/2012 14:29	11/05/2012 14:49
Stop Time	11/05/2012 14:38	11/05/2012 14:56
Total Time of Trace	00:08:25 (505.015s)	00:06:32 (392.649s)
Client - jaguar -	148.197.27.60	148.197.27.60
Client - kudu -	148.197.27.12	148.197.27.12
Client - gerbil -	148.197.27.94	148.197.27.94
Client - giraffe -	148.197.27.166	148.197.27.16
Client - koala -	148.197.27.225	-

Subsequently to the identification of players and traces, the network trace statistics of these tests can be examined in the 'appendix I'. The main reasons for not exposing those tables below is that, as it was commented in the first test, the statistics were performed with the full time of capture; thus all the game and control information are mixed in the same evaluation. However, the statistics of the appendix I contain indicatives values for a first approach for the evaluation of data captured by the devices. For instance, the statistics of the TCP version shows the same features and issues detected in the last test in an amplified way due to the packet and player increase. These issues are mainly caused by the high variations in the packets' sizes and differences RTTs.

The next figures show the packet rate of the second experiment in order to facilitate the understanding of the previous data and to select the accuracy measures to evaluate the clients' QoS.

The first figures presented in the next page correspond with the traffic analysis of the second test; firstly the server traffic and secondly the client traffic – laptop and chipmunk –. Through this figures it is possible to observe how the traffic increases in the third game due to the ball speed increase. Nevertheless, there is not any substantial difference between the first two games in the clients. Therefore, the graphics and sound effects do not impact on the packets rate. In contrast, the TCP protocol reinforces its low performance in the online games through the laptop traffic. Whereas the chipmunk traffic increases with the increase of the ball speed, the laptop traffic continues with the same values than the previous games.

The third test is presented after the second test only with the analysis of the packet rate in the TCP and UDP servers. The start and finish of the four games can be showed through the figures 35 and 36. However, the packet rate is stabilised when the first game which traffic was captured in both clients is finished. The reason is that there were not any players playing in the others games; therefore, there is not traffic caused by player's movements. Through this test is confirmed the theory about the player's movements as the cause of the peaks in the packet rate charts. The figures of the third test also shows the superiority of the UDP protocol over TCP protocol when the number of games is more than one. As the UDP version send around 80 PPS when there is four games

activated; whereas the TCP version has a packet rate of 150 PPS. Therefore, it is preferable the use of UDP protocol when the number of players is elevated by reason of the consumption of less number of packets; and thus less amount of bandwidth. – Appendix I –

Moreover, it appears that the more clear differences between captures are the networking protocol used; hence to distinguish the rest of features from the effects caused by these protocols is an arduous task. For this reason, it is helpful to have in mind the following considerations about the differences between TCP and UDP.

The protocol TCP requires the packet acknowledge from the clients before that the server sends the next packet; therefore the slowest clients receive a lower number of packets. Also, the transmission of acknowledge involves that the clients need a high output flow. Whereas, the server in the UDP version sends packets independently of clients; thus the output flow of the clients is limited to their movements. As consequence, the number of packets sent by the clients is reduced considerably.

The access technology may affect significantly the performance of these protocols increasing the delay of the packets and certainly causing the loss packet rate. Therefore, these parameters are analysed in the game three of the second test due to the increase in the packet rate, these issues may also increase.

Test 2. Server performance: Packets rate.

Black Line = all the server traffic
Green Line = chipmunk traffic (148.197.27.14)
Blue Impulses = laptop traffic (148.197.67.100)

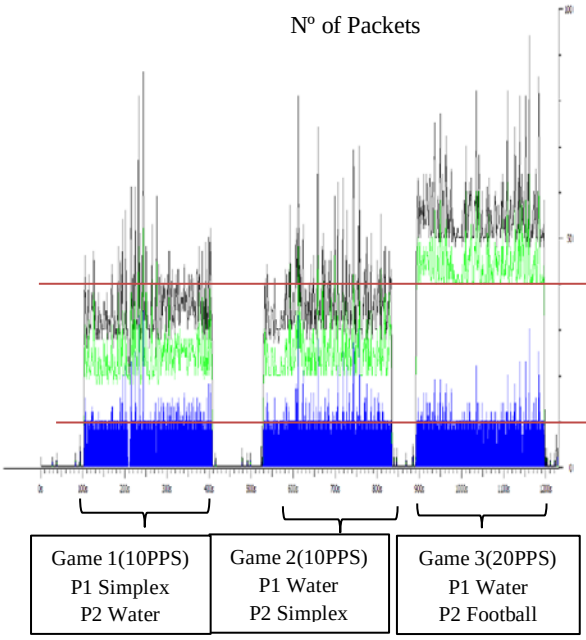


Figure 29: Test 2. Server TCP. Packet rate [PPS]

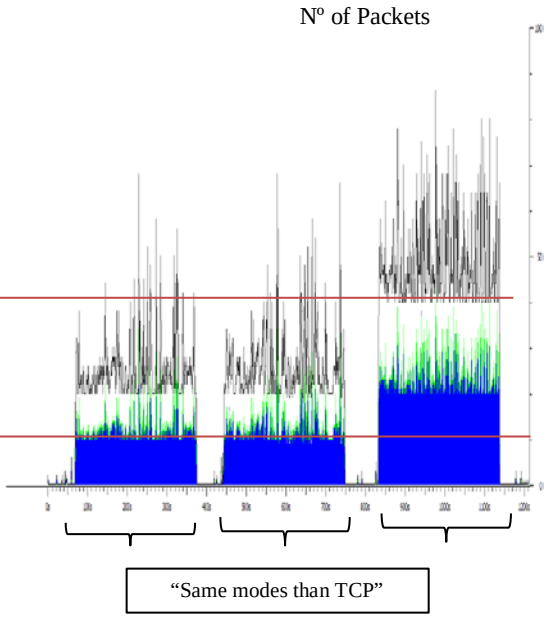


Figure 30: Test 2. Server UDP. Packet rate [PPS]

Test 2. Clients performance: Packets rate.

Black Line = all the traffic
Green Line = input packets
Blue Impulses = output packets

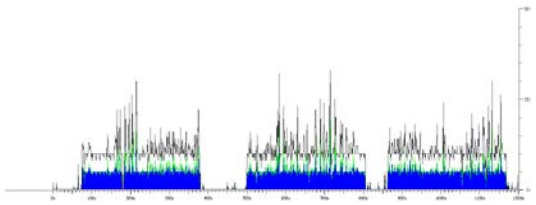


Figure 34: Test 2. Laptop TCP. Packet rate [PPS]

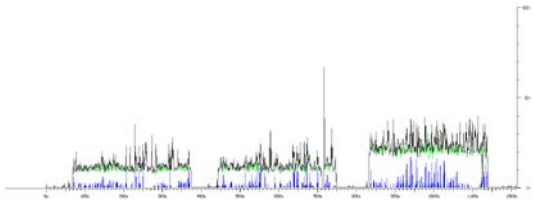


Figure 31: Test 2. Laptop UDP. Packet rate [PPS]

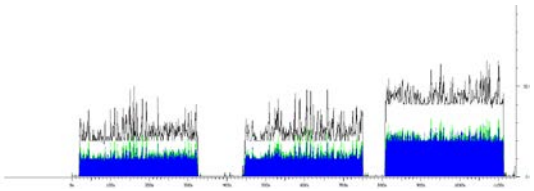


Figure 33: Test 2. Chipmunk TCP. Packet rate [PPS]

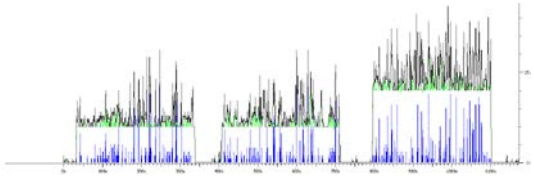


Figure 32: Test 2. Chipmunk UDP. Packet rate [PPS]

Test 3. Server performance: Packets rate.

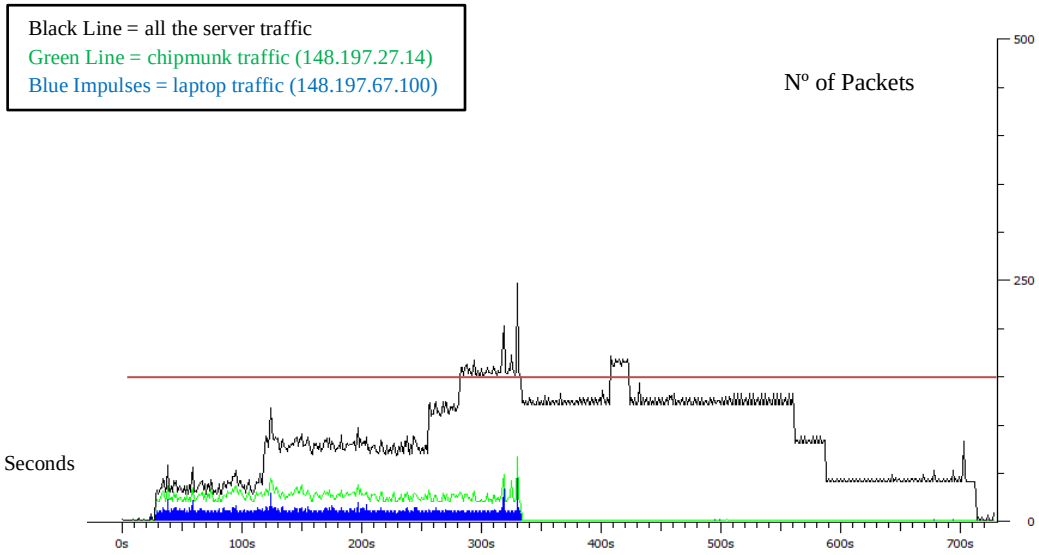


Figure 35: Test 3. Server TCP. Packet rate [PPS]

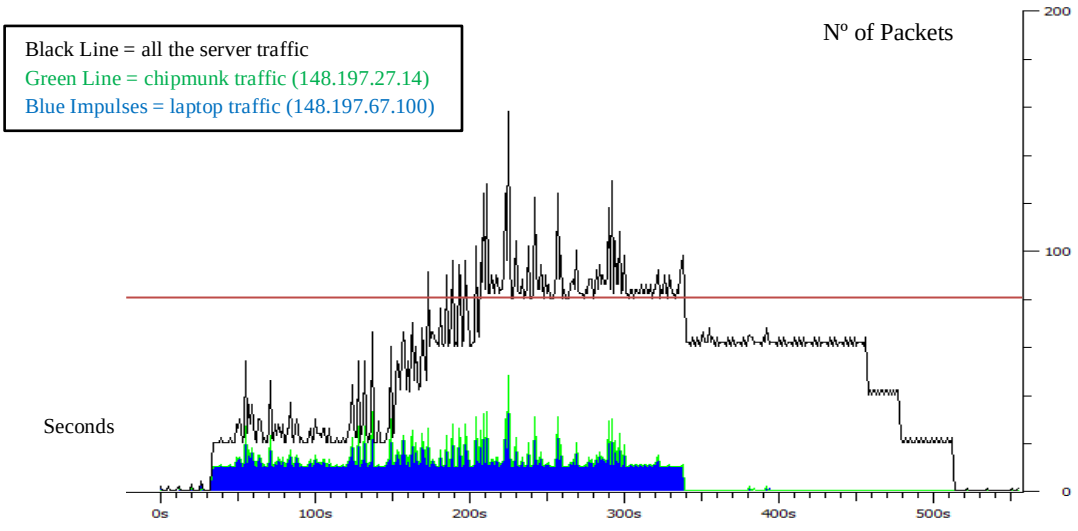


Figure 36: Test 3. Server UDP. Packet rate [PPS]

In light of the packet rate results of the second and third tests, the following analyses are planned in order to evaluate:

- The graphics and sound effects of the games one and two. (Test 2)
- The constant packet rate in the game three of the TCP laptop client. (Test 2)
- The packet loss rate in the game three of the UDP server. (Test 2)
- The QoS of the clients through the packet loss rate. (Test 2 and 3)

The graphics and sounds effects are evaluated through the TCP laptop traffic because of the extreme situation that this client is tolerating through the delay of packets and substantial size of packets which is increased due to delays. Therefore, the small variations which the graphics/sounds effects may cause in the networking parameters can be amplified in this type of situations.

The graphics and sounds resources have the slowest process in the initiation and load of the file. Therefore, the major effect of these resources on the gaming traffic should appear at the game start reducing the rate of the processor, and then increasing at the same time the delay in the transmission. In order to evaluate this effect the following figures show the output RTT times of the game one and two in the laptop client, which is also the client with less power of processing.

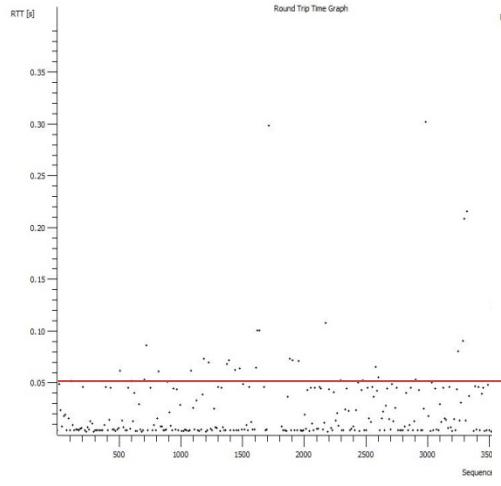


Figure 37: Test 2. Output Laptop RTT. G 1 (simplex)

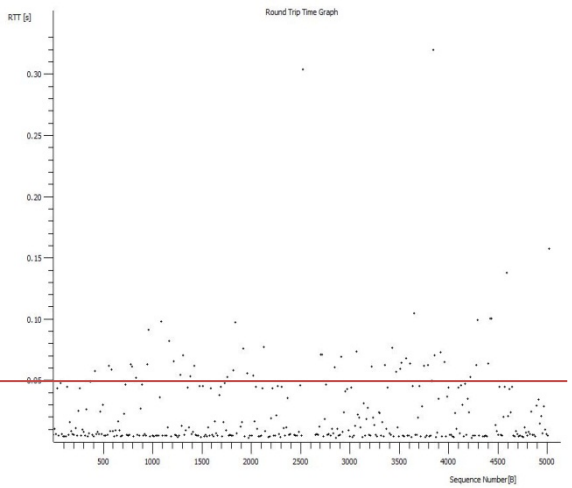


Figure 38: Test 2. Output Laptop RTT. G 2 (water)

This figures show a slight increase in the delay when the laptop client uses images and sounds in the game. However, this increase is not relevant to the network performance because of its little variation. Consequently, it is concluded that these effects do not affect the networking parameters in this game

Subsequently to the graphic and sound evaluation, the delay and throughput of the server in the game three is analysed for the purpose of analysing the packet rate and

explaining the situation in the laptop client. Because, the number of packets should increase as in the chipmunk traffic due to the increase of the packet rate; however this increase is not performed. Furthermore, whether the packet rate does not increase, the size of the packets may be increased with information delayed. Therefore, the packet size of the game three also is evaluated in addition to the RTT and Throughput in the server output stream.

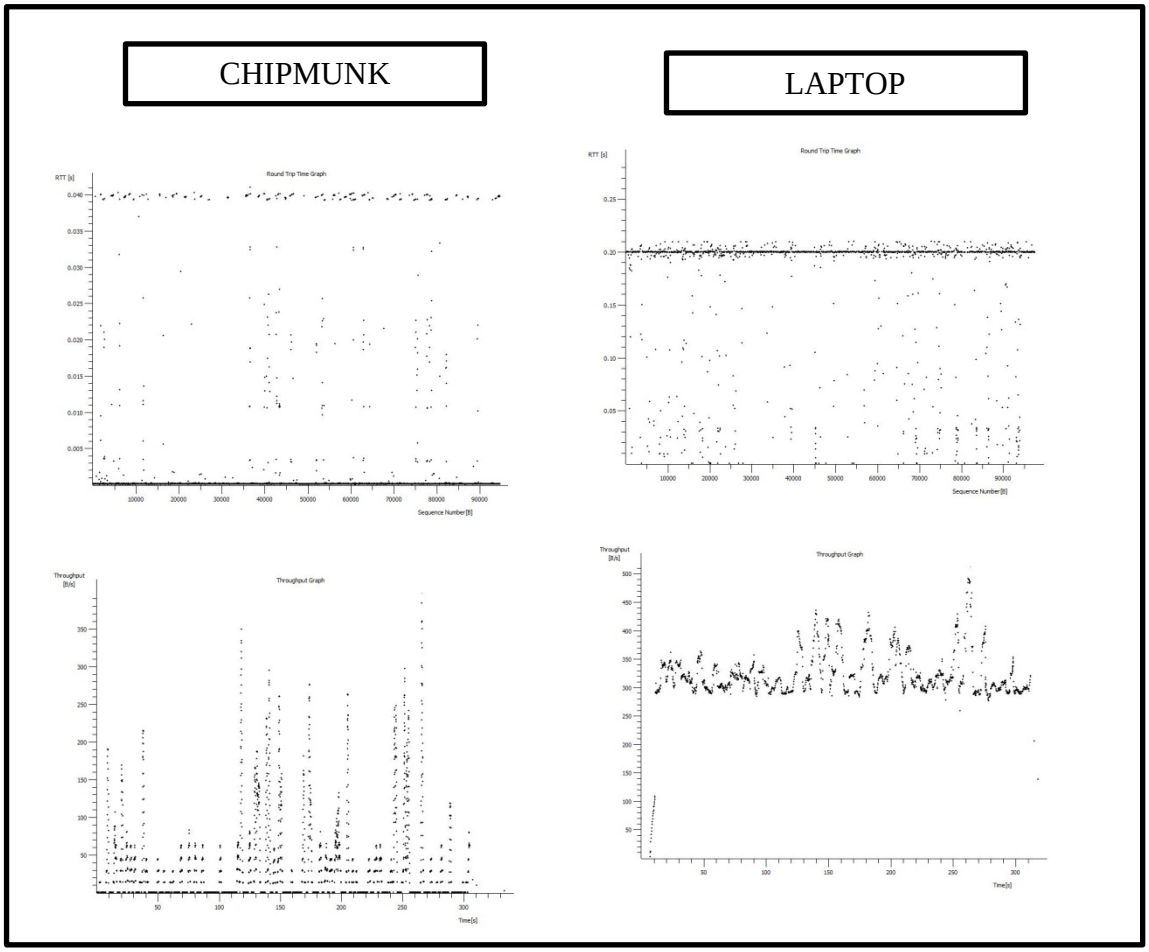


Figure 39: Test 2. Output traffic server. Game 3

The delay in the laptop client is increased four times (0.2s) respect to the first test; therefore the packet size would be increase the same amount. In contrast, the chipmunk client has the same delay than the first test.

Table 22: Test 2. Statistics of the Packet Size. Game 3 TCP

Game 3	TCP		
Size in Bytes	Server	Laptop	Chipmunk
MAX	552	552	186
min	54	47	47
Mean	76.29	84.94	73
Standard Deviation	17.47	35.85	7.35
Variance	305.48	1285.48	54.08

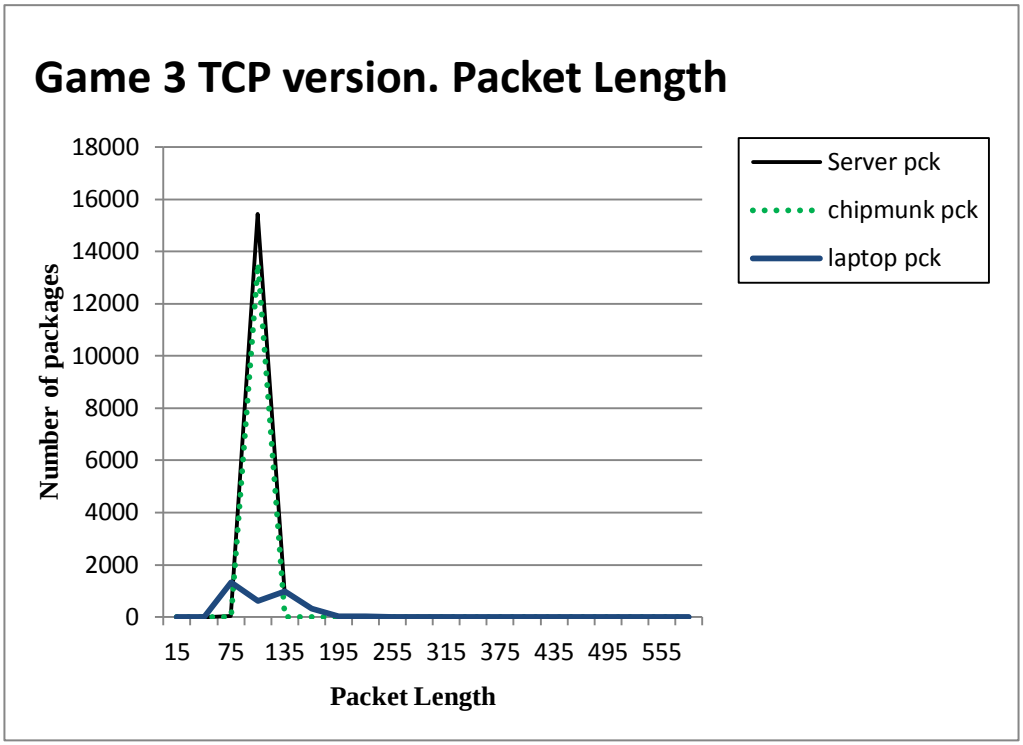


Figure 40: Test 2. Packet Lengths. Game 3 TCP.

These results confirm the increase of the packet size average respect to the first test. However, the max packet size is lower than the max packet size of the first test. Therefore, this measure depends also on the network status which is random and the connection status at the moment of the transmission. Furthermore, the dramatic difference of packets caused by the enormous delay of the laptop is clearly distinguishable in all the tests executed.

The UDP traffic was not commented until now, because all the results are the expected. The server traffic is equally distributed between both clients and whether the ball is update every 100ms the packet rate is 20 PPS (10 each client); instead whether the ball is update every 50ms the packet rate rise to 40 PPS with peaks of 50 or 60 PPS because of the player's movements. However, the UDP game has a disadvantage, the packet loss; therefore the following tables compare the packet loss rate between the game three of the UDP and TCP versions in order to compare both rates and to evaluate other lack of the TCP version, the retransmission. Furthermore, the access technologies are also evaluated due to the significant impact that has wireless and Ethernet over packet loss rate and this affect may be amplified due to the rise of packet rate.

Table 23: Test 2. Evaluation of packet loss rate

List of Packets Game 3 – Test 2	**TCP**		**UDP**	
Devices	Laptop	Chipmunk	Laptop	Chipmunk
Packets sent -Server-	1,680	6,710	6,825	6,739
Packets received -Clients -	1,671	6,710	6,685	6,739
Difference	9	0	140	0
Packets sent -Clients-	1,674	6,837	570	651
Packets received -Server -	1,653	6,837	565	651
Difference	21	0	5	0
TOTAL	30	0	145	0
Percent	1.79%	0.00%	2.12%	0.00%
Packets retransmitted sent- Client-	7	0	-	-
Packets retransmitted sent- Server-	4	0	-	-

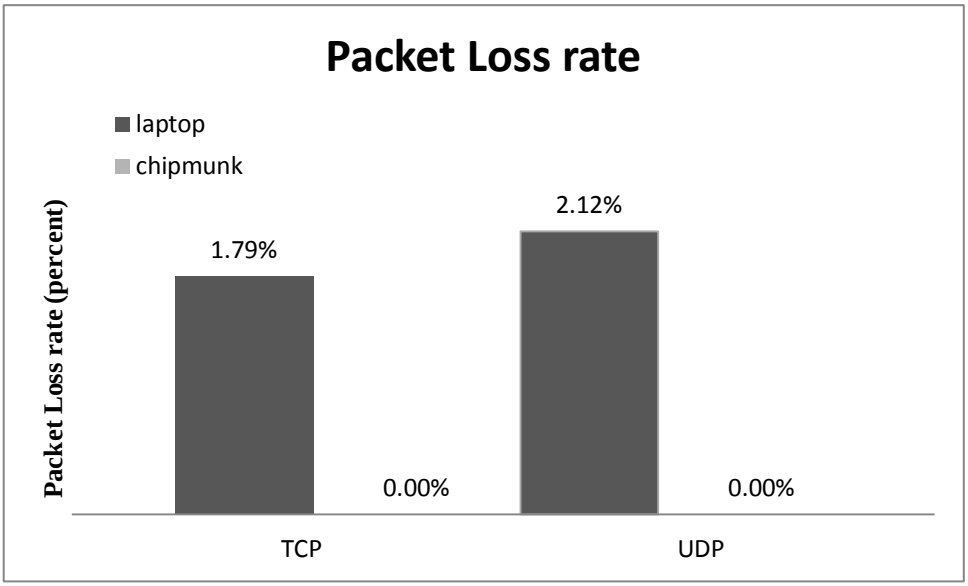


Figure 41: Test 2. Packet loss rate.

The previous results are not coherent with the results of the packet loss rate in the first test. As, the packet loss rate is lower in both version with a packet rate higher. These results may be due to the status of the university network at that moment. However, these results show a clear advantage of Ethernet technology over Wireless technology.

The packet rate and the graphic/sounds effects have been evaluated through the packet loss rate and RTT without very variations respect to the first experiment. However, the number of players may be different from the graphic and sounds effects. Therefore, the loss packet rate is analysed in order to evaluate the performance of the increase in the number of players.

Table 24: Test 3. Evaluation of packet loss rate

List of Packets. Test 3	TCP		UDP	
Devices	Laptop	Chipmunk	Laptop	Chipmunk
Packets sent -Server-	1,663	3,879	3,494	3,639
Packets received -Clients -	1,662	3,879	3,487	3,639
Difference	1	0	7	0
Packets sent -Clients-	1,608	3,988	552	407
Packets received -Server -	1,607	3,988	552	407
Difference	1	0	0	0
TOTAL	2	0	7	0
Percent	0.12%	0.00%	0.20%	0.00%
Packets retransmitted sent- Client-	1	0	-	-
Packets retransmitted sent- Server-	2	0	-	-

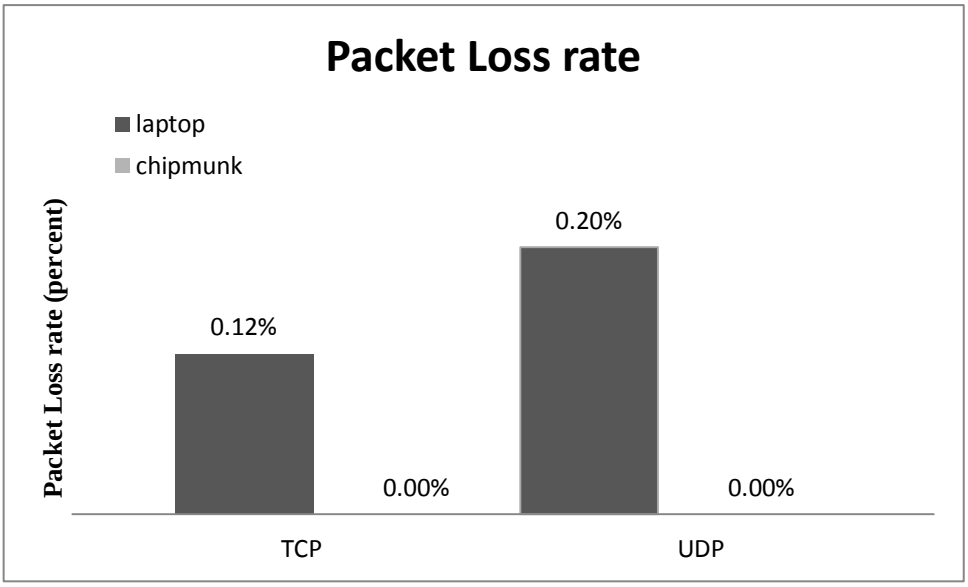


Figure 42: Test 3. Evaluation of packet loss rate

The previous results (Test 2- Game 3 and Test 3) are relatively better than the results obtained in the first experiment, however both are similar one each other. As it was mentioned it may be due to the changeable of network conditions. Therefore, the game is tested in other networks through the fourth experiment in order to compare the LAN conditions against the global Internet conditions.

Regarding the bandwidth consumption, it is an apart which has been ignored throughout the last experiment because of the lower bandwidth consumed by the devices - the higher bandwidth consumption is around 20Kbps-. However, it is highlighted the fact that the TCP version need an extra bandwidth consumption in the clients uplink in order to reply the send of packets – ACK–; condition which the UDP version does not need due to its unreliable service. The following table represent the number of bytes, packets and bandwidth consumed by the server in the game three of the second test which the packet rate of ball update is 20 PPS and the third test which there are four games run simultaneously, for the purpose of demonstrating the last data.

Table 25: Summary Servers. Test 2_ G3; Test 3

Summary Servers	**Game 3**		**Test 3**	
version	TCP	UDP	TCP	UDP
Total Packets sent	8,390	13,564	30,646	26,083
Total Bytes sent	727,930	767,517	2,466,023	1,475,772
Time [s]	300	300	728	555
Bandwidth [Kbps]	18.96	19.99	26.46	20.77
Total Packets received	8,490	1,216	30,708	1,047
Total Bytes received	559,966	72,960	2,027,419	63,014
Time [s]	300	300	728	555
Bandwidth [Kbps]	14.58	1.90	21.76	0.89

This experiment demonstrates once more the superiority of the UDP protocol over TCP protocol in terms of packet rate and number of players. The bandwidth of the UDP version is significantly lower than the TCP bandwidth. Therefore the implementation of arcade games similarly to the Pong game and that allows multiple users should be deployed using the UDP protocol. Furthermore, it was also demonstrated the little influence over network traffic of graphics and sounds effects.

4.3.3. Third Experiment

The third experiment attempt to give other point of view to the tests executed through a different networking architecture – The P2P architecture–. However, this architecture has a limitation; this version is only available in LAN; as the devices have to be connected and the LAN devices can communicate faster than the devices in other networks. The major difference from the others versions is that the client who initiates the game is the client who takes the control of the game sending the position and updating the score to the other client. Therefore, if this client generates this data, this client have available this data before the other client. However, this also means that the number of data transmitted is reduced by half. Moreover, as both clients are in the same network the packet delivery should be faster.

Table 26: Test 4. Resume

TEST 4: P2P architecture	
Players	2
Duration	10 min
Versions	P2P - UDP
Network environment	University of Portsmouth
Captures	All devices
Server	Project1 (dedicated)
Network Technology	Ethernet
Player 1	chipmunk
Network Technology	Ethernet
Player 2	deer
Network Technology	Ethernet

The version one and two (TCP & UDP) are tested in a different way to the previous experiments. The server program is run in one client machine in order to provide equal basis; thus the client who has the server run in his machine initiate his client program in localhost – 127.0.0.1-, and the other player connect his program client to the IP address of the other client.

Table 27: Test 5. Resume

TEST 5: Server in the client machine	
Players	2
Duration	10 min
Versions	P2P - UDP
Network environment	University of Portsmouth
Captures	All devices
Server	chipmunk
Network Technology	Ethernet
Player 1	chipmunk
Network Technology	Ethernet (Localhost)
Player 2	deer
Network Technology	Ethernet

The next tables summarized the information of the traces captured in the fourth and fifth tests

Table 28. Test 4. Network Trace Information

Network Trace Information (4)	P2P - UDP
Server Host	148.197.34.67
Start Time	09/05/2012 16:14
Stop Time	09/05/2012 16:28
Total Time of Trace	00:13:33 (813.293s)
Nº Games activated	2
Nº users in the system	2
Ball speed Game 1	Standard (10PPS)
Ball Speed Game 2	50ms -> 20PPS
Client - chipmunk -	148.197.27.14
Start Time	09/05/2012 16:00
Stop Time	09/05/2012 16:14
Total Time of Trace	00:13:29 (809.053s)
Mode in the Game 1 and 2	Complete Water
Client - deer -	148.197.27.70
Start Time	09/05/2012 16:00
Stop Time	09/05/2012 16:14
Total Time of Trace	00:13:24 (804.993s)
Mode in the Game 1 and 2	Simplex

Table 29: Test 5. Network Trace Information

Network Trace Information (5)	TCP	UDP
Server - chipmunk -	148.197.27.14	148.197.27.14
Start Time	11/05/2012 11:41	11/05/2012 12.12
Stop Time	11/05/2012 11:53	11/05/2012 12:25
Total Time of Trace	00:12:23 (743.395s)	00:12:05 (725.047s)
Nº Games activated	2	2
Nº users in the system	2	2
Ball speed Game 1	Standard (10PPS)	Standard (10PPS)
Ball Speed Game 2	50ms -> 20PPS	50ms -> 20PPS
Client - chipmunk -	127.0.0.1	127.0.0.1
Start Time	11/05/2012 11:41	11/05/2012 12.12
Stop Time	11/05/2012 11:53	11/05/2012 12:25
Total Time of Trace	00:12:46 (766.490s)	00:12:20 (740.717s)
Game Mode 1 and 2	Complete Water	Simplex
Client - deer -	148.197.27.70	148.197.27.70
Start Time	11/05/2012 11:41	11/05/2012 12.13
Stop Time	11/05/2012 11:53	11/05/2012 12:25
Total Time of Trace	00:12:02 (732.776s)	00:11:42 (702.804s)
Game Mode 1 and 2	Complete Water	Simplex

The most novel aspect of this experiment respect to the previous experiments is the substitution of the laptop client with the deer computer in the laboratory A2.3. This change was necessary due to the impossibility to communicate the laptop client with the computer of the lab A2.3 (Appendix J). Furthermore, these tests are executed similarity to the second test, capturing the traffic of two games with different packet rate 10 and 20 PPS.

The trace network statistics of the fourth test are showed below; whereas the fifth test results can be found in the Appendix I.

These results are generic; however it is possible to observe a considerable reduction in the number of packets in both tests. This is because one of the clients does not send packages to the network. In case of the P2P version, because one of the players is the responsible for generating the packets; hence it is not necessary his own retransmission of information generated by him. And, in the case of the first and second version, because one of the clients sends information in localhost; hence this information does not throughout the network. Consequently, the information sends in localhost is not taken into consideration when the fifth test analysis is executed due to the previous reason.

Furthermore, as the clients are the responsible for transmitting the game information, the performance of these clients should be high. Thus, it would be interesting execute an analysis about the IAT to measure the performance of the clients. However, through the specifications of the hardware in this document can be watched the remarkable power of the A2.3 computers. Therefore, the measure of the IAT would not reveal any important information. Consequently, this portion will be focus on the packets loss rate and number of packets transmitted for the purpose of analysing the application performance.

Table 30: Test 4. Trace Network Statistics.

Trace Network Statistics (4)	**P2P - UDP**
Server Host	148.197.34.67
Total Packets	55
I / O	19 / 36
Mean Packet Size	67.164 bytes
I / O [bytes]	65.474 / 68.056
Mean Bandwidth	-
I / O [Kbps]	-
Mean Packets rate	0.068
I / O [PPS]	0.023 / 0.045
Client - chipmunk -	148.197.27.14
Total Packets	11,397
I / O	1,169 / 10,228
Mean Packet Size	56.758 bytes
I / O [bytes]	60.134 / 56.372
Mean Bandwidth	6.25 Kbps
I / O [Kbps]	0.68 / 5.57
Mean Packets rate	14.087 PPS
I / O [PPS]	1.452 / 12.642
Client - deer -	148.197.27.70
Total Packets	11,396
I / O	10,236 / 1,160
Mean Packet Size	59.596 bytes
I / O [bytes]	60.018 / 55.868
Mean Bandwidth	6.59 Kbps
I / O [Kbps]	6.02 / 0.63
Mean Packets rate	14.157PPS
I / O [PPS]	12.842 / 1.441

Test 4. P2P version.

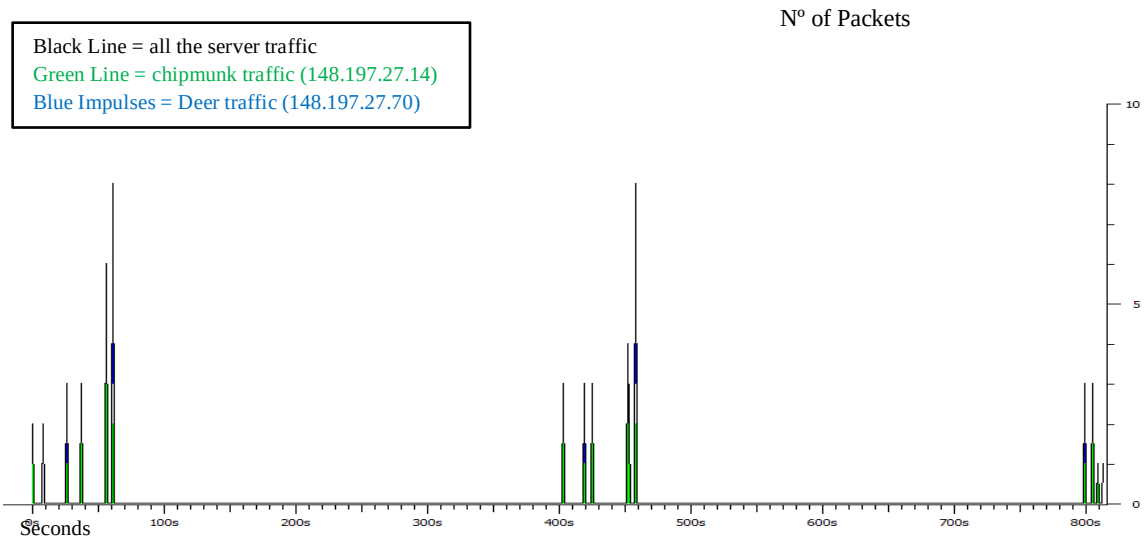


Figure 43: Test 4. Server P2P. Packet rate [PPS]

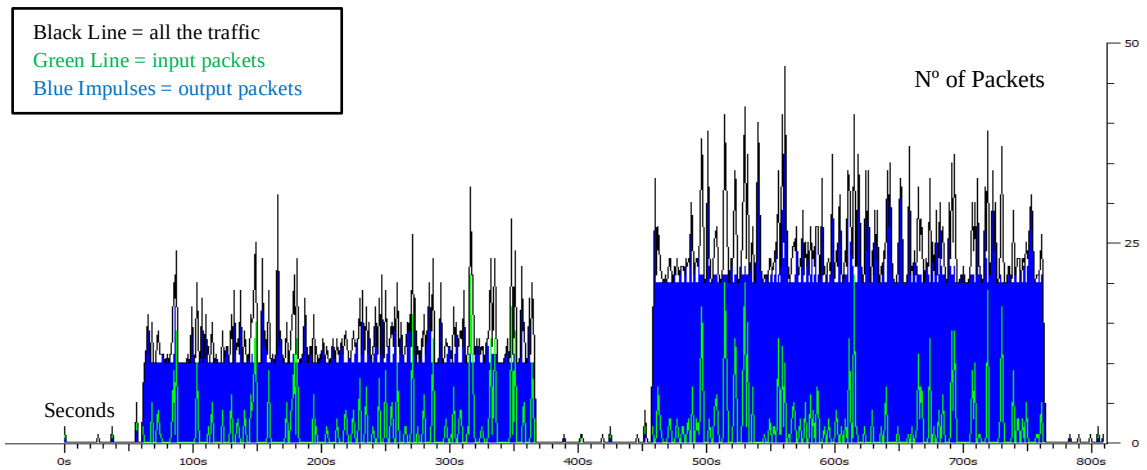


Figure 44: Test 4. Chipmunk P2P. Packet rate [PPS]

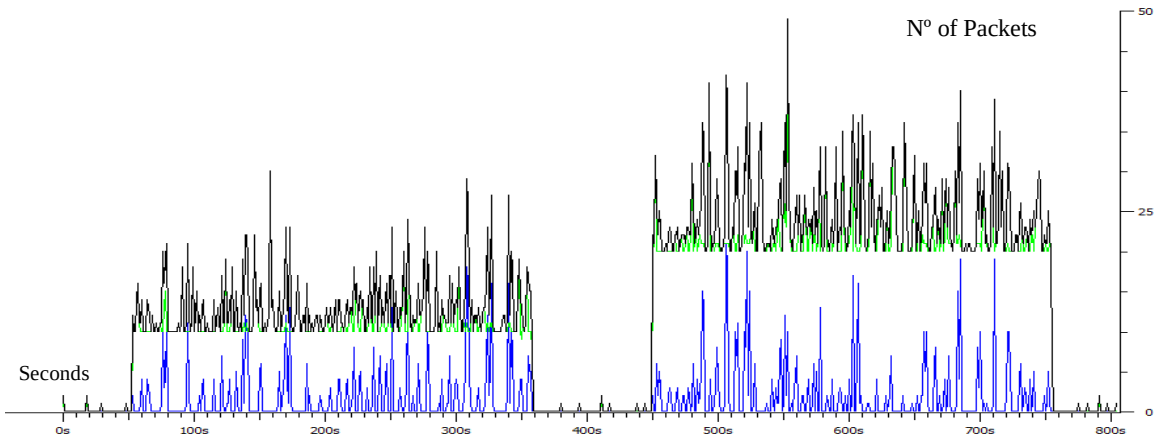


Figure 45: Test 4. Deer P2P. Packet rate [PPS]

The last figures show the reduction of the server performance; moreover below the next figures show the packet rate of the fifth test, in which the UDP version shows first the game with the packet rate of 20 PPS and subsequently the game with the packet rate of 10PPS.

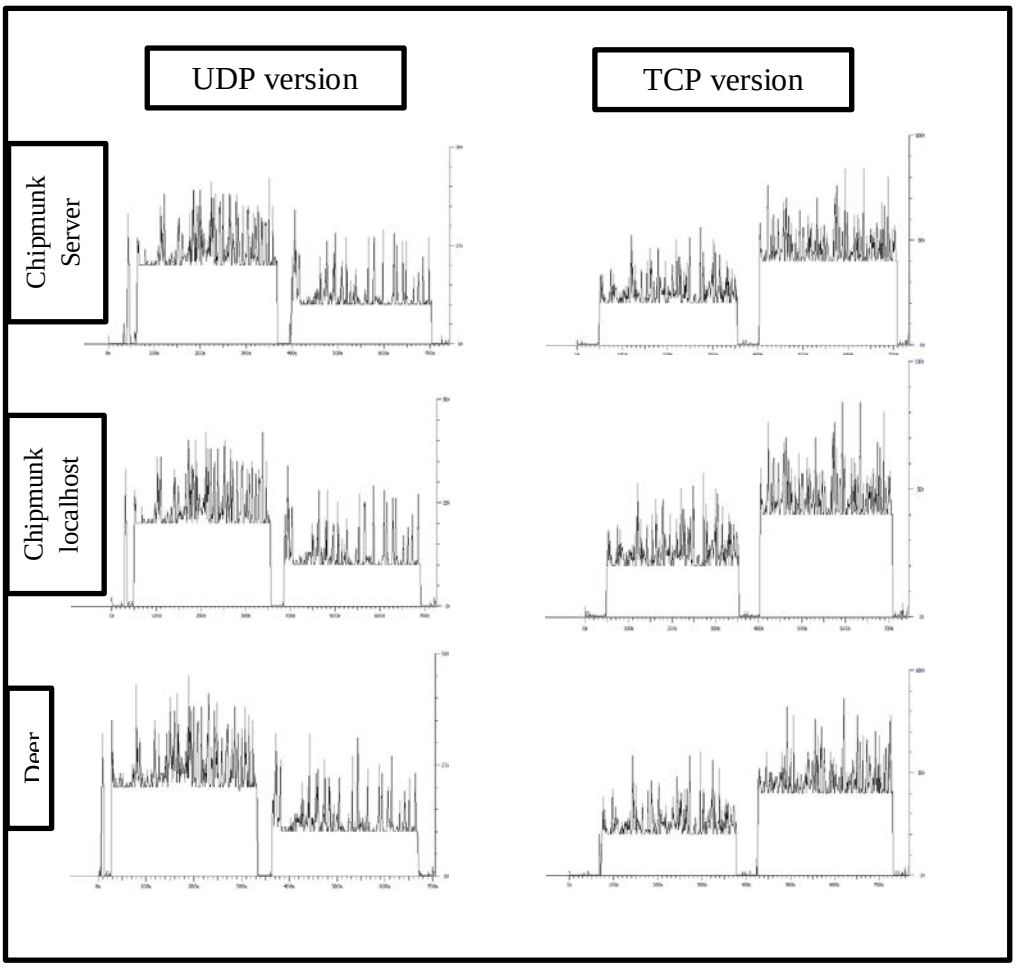


Figure 46: Test 5. Packet rate [PPS]

Through the figure 45 it is possible to observe a higher performance of the P2P version respect to the TCP and UDP versions. It is due to its implementation. This version does not send packets to one client; whereas the others versions despite some information is sent in localhost, it is still packets sent. Therefore the version two obtain a better performance through the packet number reduction.

After the preliminary analysis of the packet flow in both test, it is represented the loss packet rate and the bandwidth consumption of each version which are divided into first game – 10PPS – and second game – 20PPS –

Table 31: Test 4 and 5. Packet loss rate. G1

List of Packets. Game 1	P2P	TCP	UDP
Devices	Deer	Deer	Deer
Packets sent -Chipmunk-	3,429	3,904	3,345
Packets received -Clients -	3,429	3,904	3,345
Difference	0	0	0
Packets sent -Clients-	516	3875	395
Packets received -Chipmunk -	516	3875	395
Difference	0	0	0
TOTAL	0	0	0
Percent	0.00%	0.00%	0.00%
Packets retransmitted sent- Client-	-	0	-
Packets retransmitted sent- Server-	-	0	-

Table 32: Test 4 and 5. Packet loss rate. G2

List of Packets. Game 2	P2P	TCP	UDP
Devices	Deer	Deer	Deer
Packets sent -Chipmunk-	6,789	7,036	6,716
Packets received -Clients -	6,789	7,036	6,716
Difference	0	0	0
Packets sent -Clients-	634	7073	515
Packets received -Chipmunk -	634	7073	515
Difference	0	0	0
TOTAL	0	0	0
Percent	0.00%	0.00%	0.00%
Packets retransmitted sent- Client-	-	0	-
Packets retransmitted sent- Server-	-	0	-

The following figure can summarized the results of both tables due to the accuracy of Ethernet technology. Therefore, the table 33 may show the differences of the captures analyzed in terms of bandwidth.

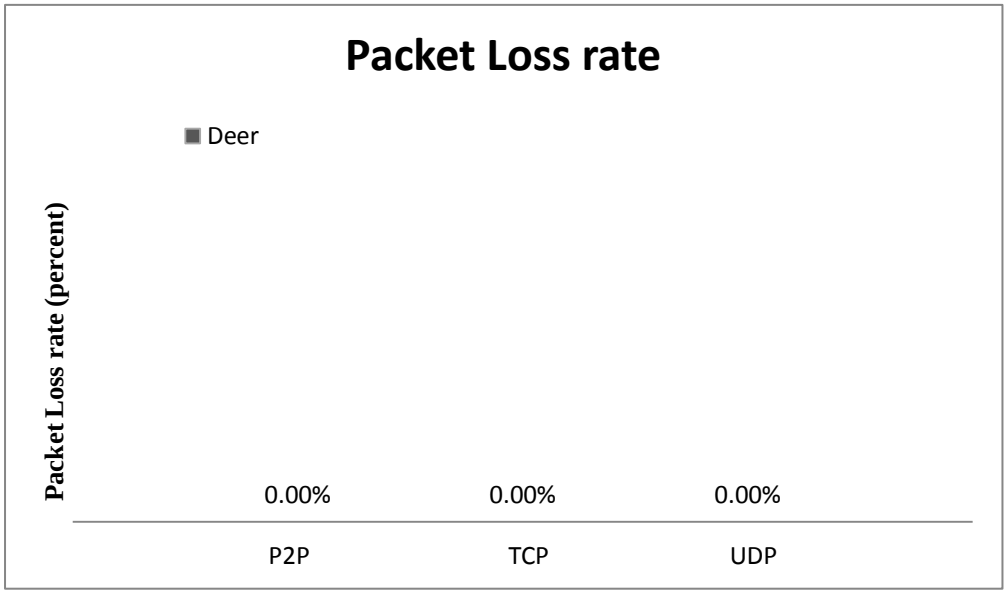


Figure 47: Test 4 and 5. Packet loss rate. G1 & G2

Table 33: Test 4 and 5. Summary servers G1 & G2

Summary Server	Game 1			Game 2		
version	P2P	TCP	UDP	P2P	TCP	UDP
Total Packets sent	3,429	3,904	3,345	6,789	7,036	6,716
Total Bytes sent	193,321	308,603	189,271	382,644	562,734	379,830
Time [s]	300	300	300	300	300	300
Bandwidth [Kbps]	5.03	8.04	4.93	9.96	14.65	9.89
Total Packets received	516	3,875	395	634	7,073	515
Total Bytes received	30,960	262,245	23,700	38,040	475,092	30,900
Time [s]	300	300	300	300	300	300
Bandwidth [Kbps]	0.81	6.83	0.62	0.99	12.37	0.80

Through the results of this experiment, it is possible to remarks the similarity of the versions P2P and UDP in these conditions. However, the UDP version allows the user to communicate and to play through different networks as it is execute in the previous tests. As a consequence of this fact, it involves an increase in the bandwidth. Therefore, the games in LAN are a suitable option to play without spends money in a server using the P2P architecture.

4.3.4. Fourth Experiment

The last experiment examines the performance of the Pong game in an environment different from the University of Portsmouth network, the game is loaded in a server allocated in Madrid (Spain) which is accessible from every network.

The only versions tested are the first and second, because the third version may generate connection problems between the LAN networks; moreover the performance may be very poor due to the strong dependency on clients' machine.

Firstly, the game is executed through the laptop client situated in the University of Portsmouth Network and other laptop situated in the same network than the server. The games evaluated have duration of two minutes and the number of games tested in each test is two, one with the packet rate of 10PPS and other with 20PPS.

After the positive results of the UDP version and the unfortunates results of the version TCP the game is analysed one time more in a different network situate at similar distance than the University of Portsmouth Network and the server network (Madrid - Spain) situated at house of Tenerife (Spain)

The reason of the reduction in the game time to two minutes is the adaptation of the game play to real situations. In other words, it is more probably that the players chose a game with duration of two minutes to a game with duration of five minutes, due to the game characteristics.

The summary and the network trace information of this test are represented below:

Table 34: Test 6. Resume

TEST 6: Game through Internet	
Players	2
Duration	4 min
Versions	TCP & UDP
Network environment	Internet
Captures	All devices
Server	Madrid_Server
Public IP	213.37.10.199
Private IP	192.168.1.7
Network Technology	Ethernet
Player 1	laptop
Private IP	148.197.67.100
Network Technology	Wi-Fi

Player 2	Madrid_Laptop
Private/Public IP	192.168.1.5
Network Technology	Wi-Fi (LAN)
Player 3	Tenerife_Computer
Public IP	87.223.112.241
Private IP	192.168.1.128
Network Technology	Ethernet

Table 35: Test 6. Network Trace Information

Network Trace Information (6)	**TCP**	**UDP**
Madrid_Server	192.168.1.7	192.168.1.7
Start Time	13/05/2012 16:39	13/05/2012 17:45
Stop Time	13/05/2012 16:46	13/05/2012 17:51
Total Time of Trace	00:06:39 (399.178s)	00:06:13 (473.270s)
Nº Games activated	2	2
Nº users in the system	2	2
Ball speed Game 1	Standard (10PPS)	Standard (10PPS)
Ball Speed Game 2	50ms -> 20PPS	50ms -> 20PPS
Client - laptop -	148.197.67.100	148.197.67.100
Start Time	13/05/2012 16:39	13/05/2012 17:45
Stop Time	13/05/2012 16:46	13/05/2012 17:51
Total Time of Trace	00:07:13 (433.721s)	00:05:54 (354.948s)
Client -Madrid_laptop -	192.168.1.5	192.168.1.5
Start Time	13/05/2012 16:39	13/05/2012 17:46
Stop Time	13/05/2012 16:46	13/05/2012 17:51
Total Time of Trace	00:07:47 (467.302s)	00:05:09 (309.764s)
Network Trace Information (6.1)	**TCP**	**Server**
Client -Tenerife_Computer -	192.168.1.128	192.168.1.128
Start Time	13/05/2012 17:19	13/05/2012 17:17
Stop Time	13/05/2012 17:28	13/05/2012 17:28
Total Time of Trace	00:09:42 (582.005s)	00:11:23 (683.270s)

The server and the player two are in the same network, as a result both keeps the three first digits of their IP address equal – 192.168.1.X– .However, this fact also means that the communication between them is faster than with the rest of game devices. This was in part because both use the ARP – Address Resolution Protocol –; instead of the DNS (Domain Name System) method to know the direction of the devices.

Firstly, it was executed the capture of TCP and UDP traffic only between the Madrid_laptop and the laptop client. However, the TCP results are inconclusive due to the difference in the traffic between clients may be due to the difference of networks – LAN/WAN or broadband technology – different types of wireless and distances to the

router –; therefore it was executed other TCP test between the laptop client and computer situated in other different home network situated in Tenerife which uses Ethernet technology. As a consequence, the TCP version is completely clear at the end of the test. The difference in the delays is caused by the difference of networks. Whether both players are in the same network, the TCP version provides a high quality; however whether the players are in different networks, the quality is extremely low. Whereas, the UDP version often provides high quality using an adequate amount of resources – number of packets and bandwidth-; however, the fact which may cause issues to this versions is the broadband technology and the packet loss rate.

The TCP version demonstrated in the entire experiments executed the low quality which offers; nevertheless it is interesting the continuous studies of this version due to the results are amplified in extreme conditions. Hence, it is easier to differentiate the effects due to delays in the different networks or access technologies.

The next figures show the packet rate of the versions TCP and UDP, in which in the case of the TCP server can be observed the difference in the number of packet between both clients. However this difference is lower than the previous test with the clients: chipmunk and laptop. The network trace statistics of this test can be found in the appendix I.

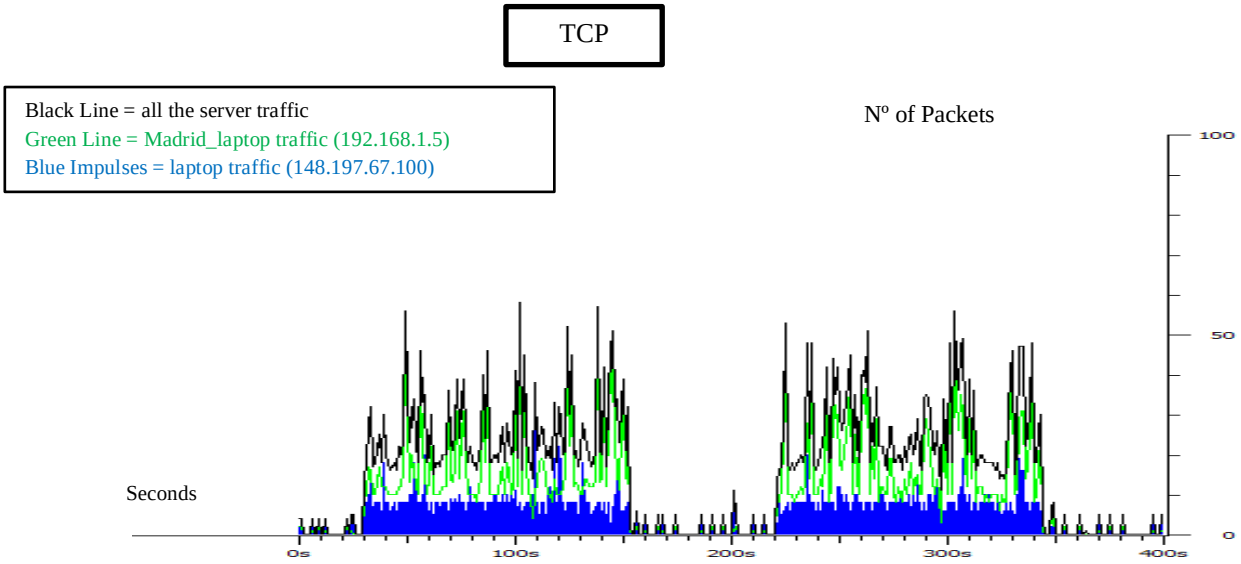


Figure 48: Test 6. Madrid Server TCP. Packet rate [PPS]

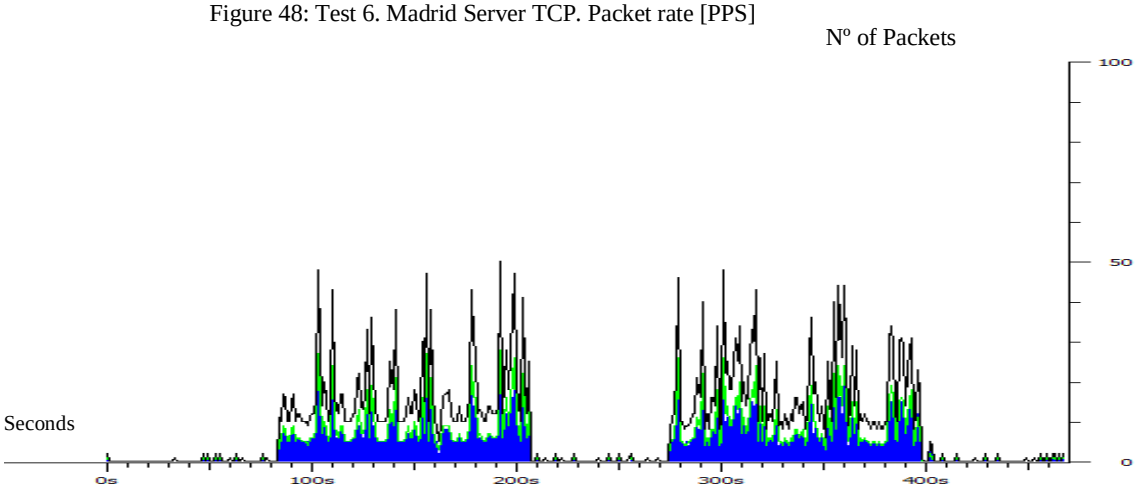


Figure 49: Test 6. Laptop TCP. Packet rate [PPS]

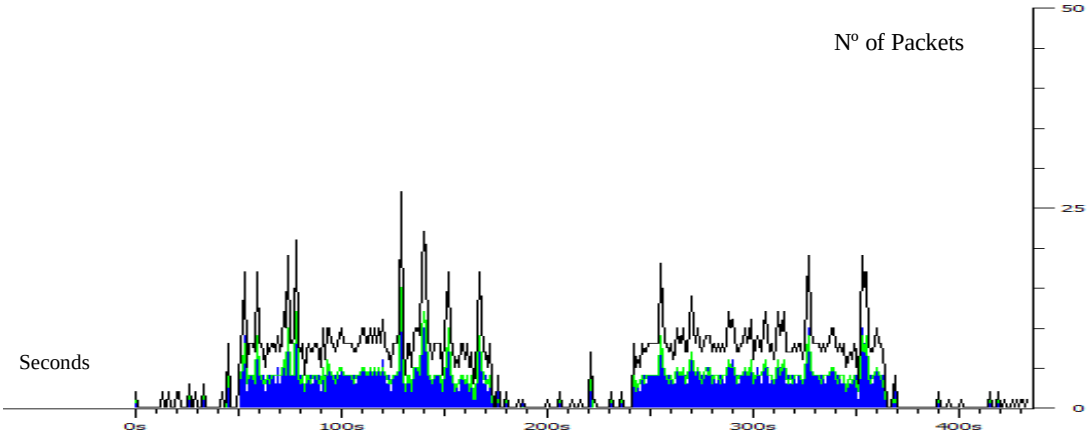


Figure 50: Test 6. Madrid_laptopTCP . Packet rate [PPS]

UDP

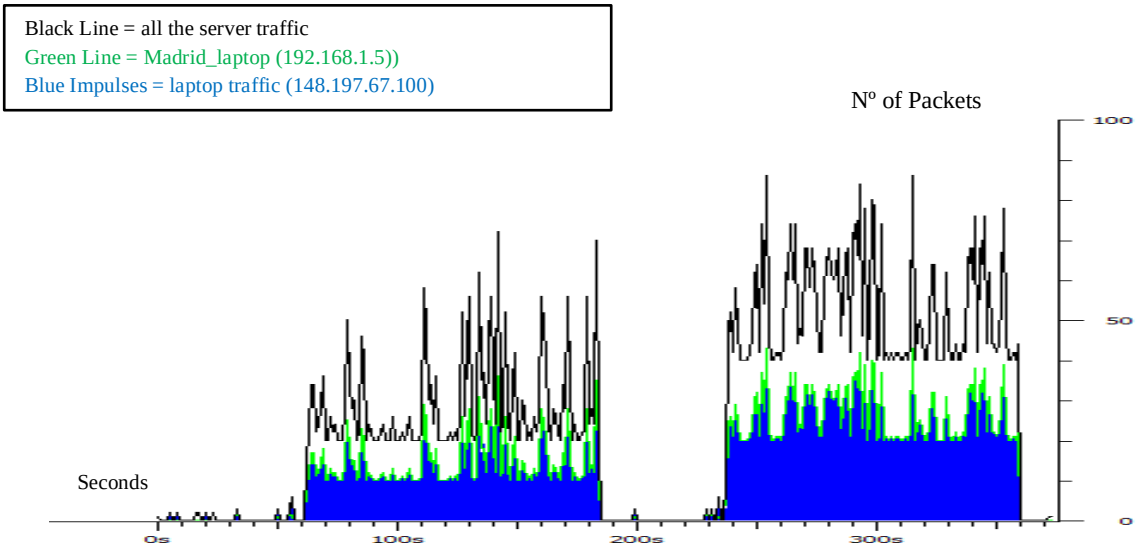


Figure 51: Test 6. Madrid Server UDP. Packet rate [PPS]

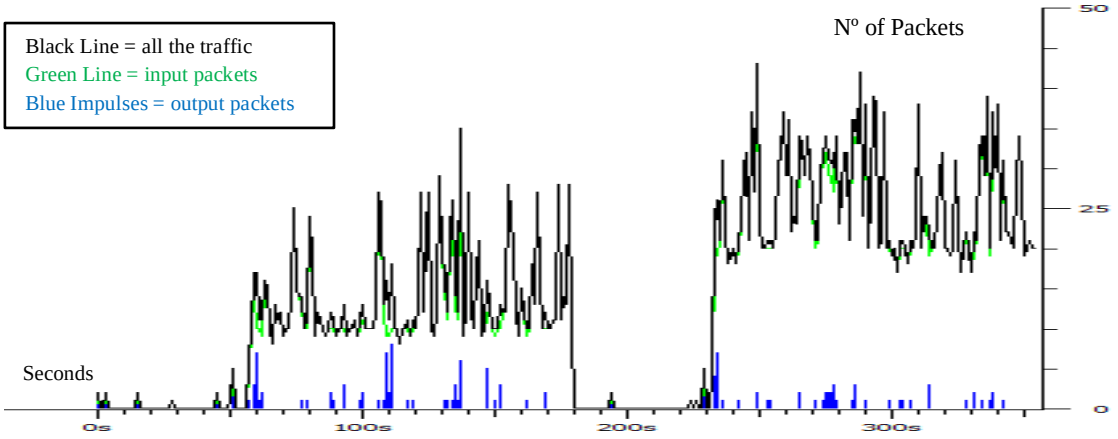


Figure 53: Test 6. Laptop UDP. Packet rate [PPS]

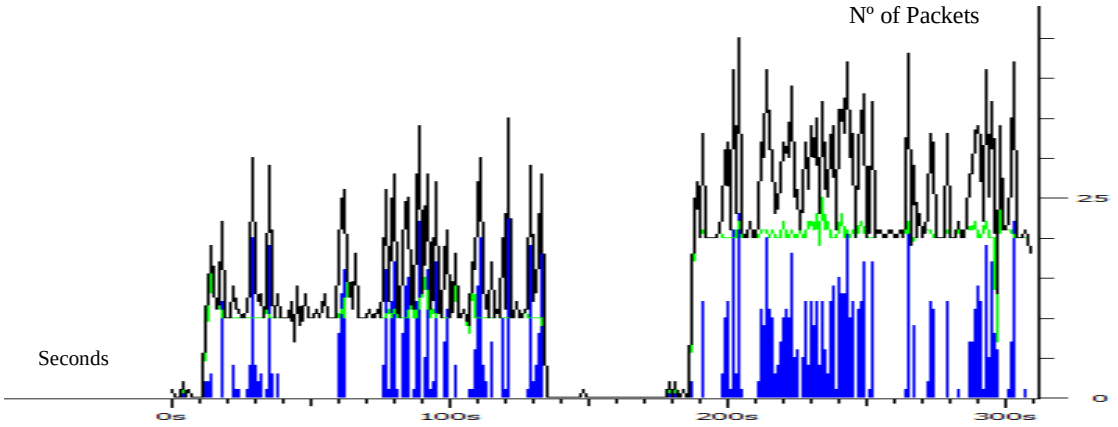


Figure 52: Test 6. Madrid_Laptop UDP. Packet rate [PPS]

Curiously, the difference in the number of packets between clients is removed when both clients are in different networks.

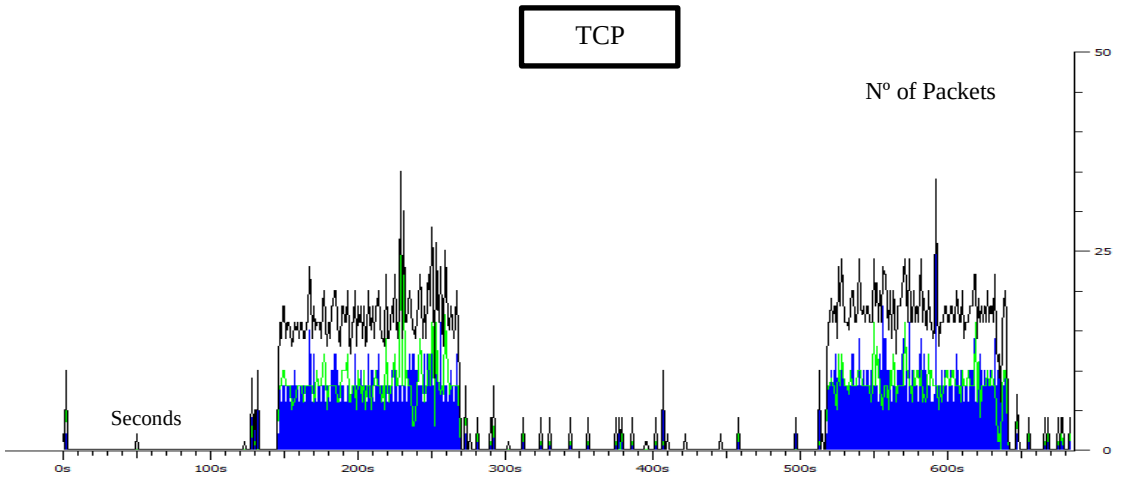


Figure 54: Test 6.1. Madrid_Server TCP. Packet rate [PPS]

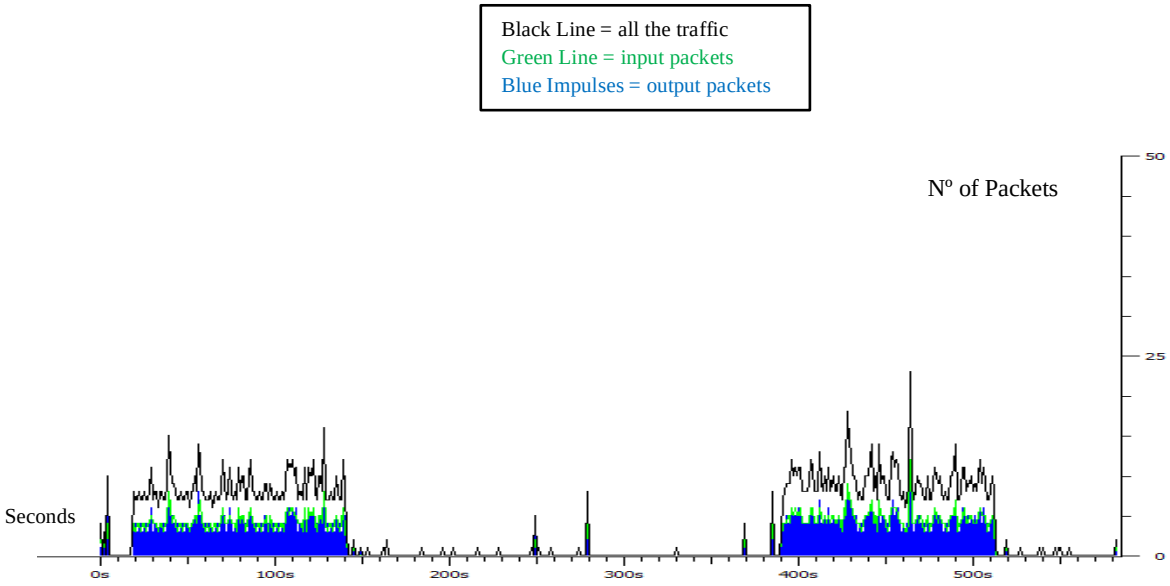


Figure 55: Test 6.1. Tenerife_Computer TCP. Packet rate [PPS]

Below delay is measured and analysed in the different devices of the TCP versions, and finally the UDP version is evaluated in terms of packet loss rate.

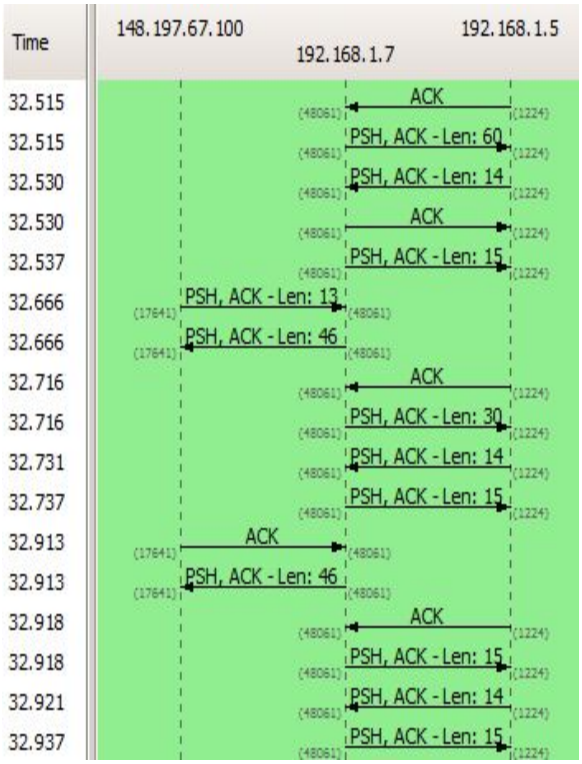


Figure 56: Test 6.TCP flow

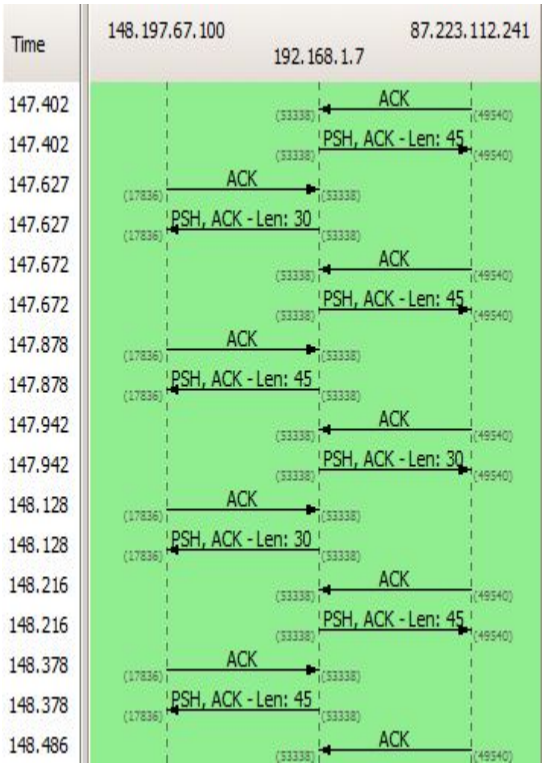


Figure 57: Test 6.1. TCP flow

The previous figures show how the RRT time is lower in the Madrid devices than in the devices of the other networks. However, through the delay measure is possible to observe that the difference is not as enormous as seems in the TCP flow. The delays of laptop and the Tenerife computer clients, which use different access technologies, is around 0.28 s; whereas in the Madrid_laptop client the delay is 0.20. Both delays are higher than the delay found in the first test. This may be caused for the network status, an element that the games developers cannot control.

Furthermore, the delay is not a problem in the case of UDP due to its higher rate. Therefore, the online games with real-time requirements should be implemented with the UDP protocol; except that some data are very important to exchange of speed for security. However, UDP has a problem in the loss packet rate. This problem is showed in the table 36 and the figure 61.

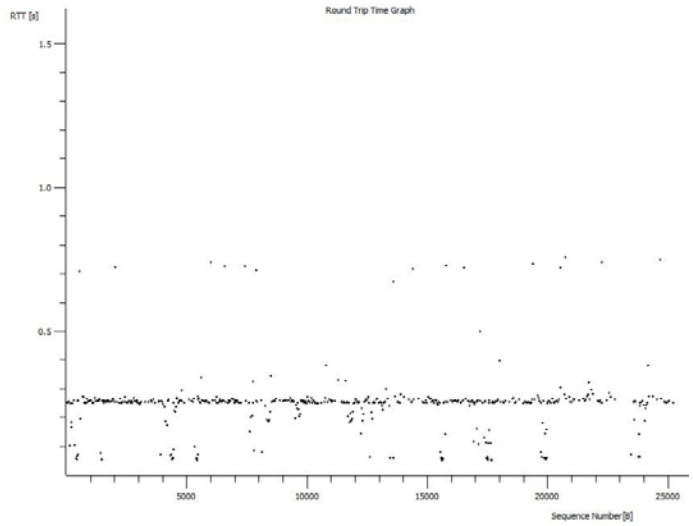


Figure 58: Test 6. TCP Laptop. RTT

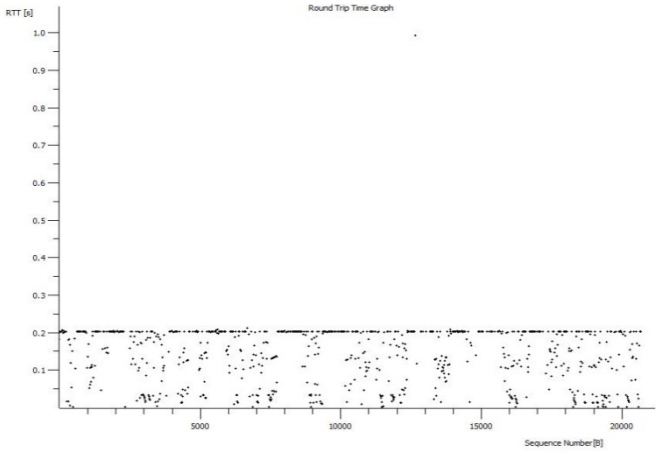


Figure 59: Test 6. TCP Madrid_Laptop. RTT

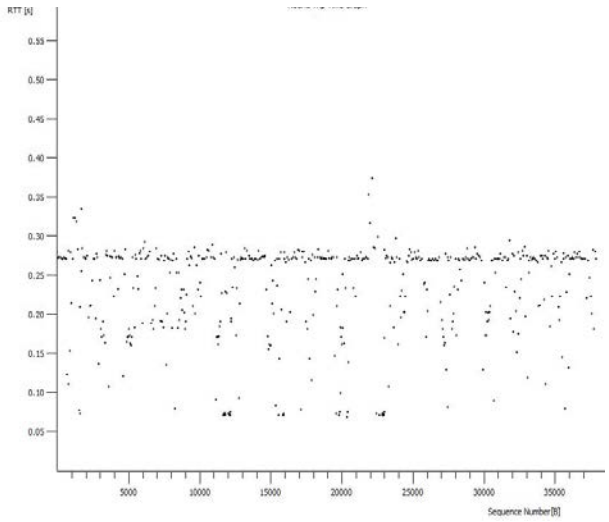


Figure 60: Test 6.1. Tenerife Computer. RTT

Table 36: Test 6. Packet loss rate. Global

List of Packets. Test 6	TCP		UDP		TCP(6.1)
Devices	Laptop	Madrid _L	Laptop	Madrid _L	Tenerife_C
Packets sent -Server-	1,093	2,245	4,832	3,829	1,105
Packets received -Clients -	1,066	2,245	4,746	3,826	1,105
Difference	27	0	86	3	0
Packets sent -Clients-	1,068	2,040	123	1129	1,112
Packets received -Server -	1,068	2,040	123	1128	1,109
Difference	0	0	0	1	3
TOTAL	27	0	86	4	3
Percent	2.47%	0.00%	1.78%	0.10%	0.27%
Packets retransmitted sent- Client-	3	2	-	-	1
Packets retransmitted sent- Server-	5	4	-	-	3

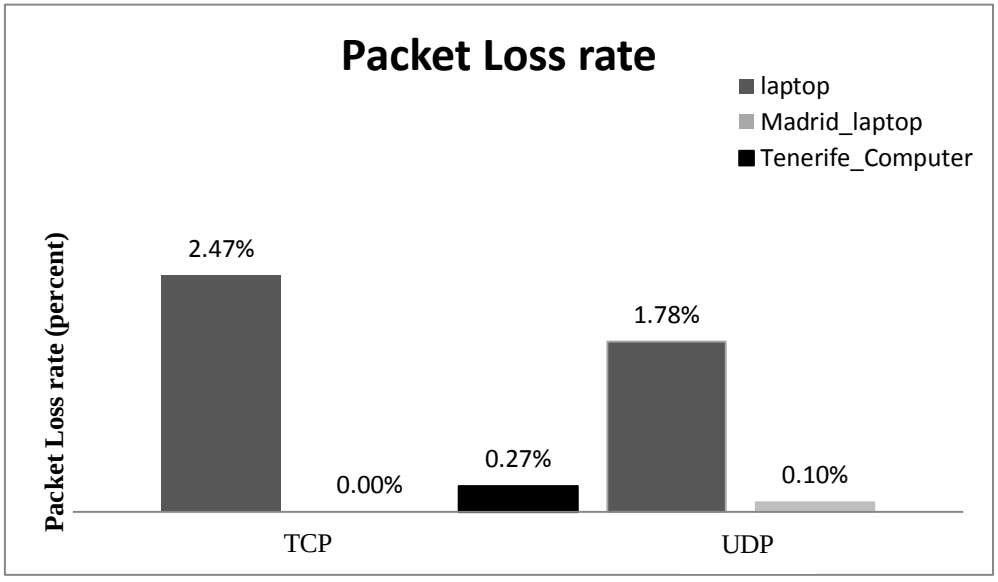


Figure 61:Test 6. Packet loss rate. Global

The packet loss rate seems similar to the packet loss rate of the previous experiments. The marked increase in the difference between packets sent by the server and received by the laptop client may be due to a loss of connection. The rest of values reinforce the hypothesis that wireless networks and UDP protocol have a higher number of packet loss than Ethernet technology and TCP protocol. However this loss is not relevant and significant for time-real applications.

The following table show the bandwidth and number of packets in each game –first game 10PPS and second game 20PPS - which in this case is relevant and significant for time-real applications. The results of the TCP reinforces the bad results obtained in the entire experiments due to the delays caused by the packet acknowledgment – ACK–; which limit the bandwidth in both games becoming worse the game quality. On the contrary, the UDP protocol can adjust the bandwidth depending on the game speed.

Through this test the technical analysis is finished. The most significant results are summarized in the following section together with a list of recommendations for developing games online. – 5. CONCLUSSIONS –

Table 37: Test 6. Summary Servers. G1 & G2

Summary Server	**Game 1**			**Game 2**		
version	TCP	UDP	TCP (6.1)	TCP	UDP	TCP (6.1)
Total Packets sent	1,652	2,983	1,062	1,686	5,678	1,094
Total Bytes sent	136,192	169,312	98,859	175,910	321,851	139,412
Time [s]	120	120	120	120	120	120
Bandwidth [Kbps]	8.87	11.02	6.44	11.45	20.95	9.08
Total Packets received	1,522	521	1,036	1,586	730	1,083
Total Bytes received	98,097	31,544	67,617	103,097	44,008	70,904
Time [s]	120	120	120	120	120	120
Bandwidth [Kbps]	6.39	2.05	4.40	6.71	2.87	4.62

4.4. Quality of Service Analysis

The previous analysis was focused on the technical parameters of the game; however it is important to evaluate the gaming effects on the final users. Therefore, this analysis is focused on the players' opinions respect to the game. The preferences of the players and their opinions are evaluated through the survey of the appendix H. However, firstly the generic considerations which were obtained during the process are showed below.

The captures and server output of the third test can show a high number of chat packets; therefore it can be argued that it is indispensable in all online-multiplayer game a mechanism that allow to communicate and interact the players between them. This method can bring players as close as possible to the experience of the game and to share experiences in the game. This fact may be amplified in LAN due to the players' proximity. Consequently, the chat is a simple mechanism that in addition to allow the synchronization of the games, also allows sharing experiences and make new friends.

From a technical point of view, the UDP version seems to create inconsistence in the players' score through the method of score update implemented. It is due to the loss of packets and unreliable service which offer the UDP version. Nevertheless, this protocol is the fastest; therefore this lack is necessary to be solved in the next version of this game through a different method of score update.

The results of the survey executed show a clear preference of the players for the graphics modes– football and water –; however this is not the element more valuable for the players who coincide in the game logic as indispensable element for playing games. It is achieved in this game through the randomness to the ball.

Despite the choice of the graphics mode as the favourites mode, the players does not believe necessary to implement much more graphics/sounds affect than it is implemented now. However, a very common recommendation made by some of the players is the improvement of the celebration when the game is over and announces the winner.

Regarding the game versions, it can be divided the players into two groups the players who were playing the game using their laptops and the players who were playing with the A2.3 computers. The first players stressed the speed of the UDP version over the TCP version. However, the second group of players did not appreciate difference between game versions.

Finally, the question about the game style results an apparent preference of all the players for the style of the portal; however the most noticeable element was the chat which the majority of players highlight this element as a suitable system for waiting until game start. As a consequence of this analysis, it seem that the interaction between players and an interesting game logic game are essential requirements for becoming successful the game.

This page intentionally left blank

5. CONCLUSIONS

Online multiplayer games represent a considerable portion of traffic coursed on the Internet; in addition this industry has become one of the biggest industries competing even with music and movie industries. However, the study about the traffic of these applications had been largely forgotten until the last years; when bandwidth of games became one of the common concerns of this sector due to the cost involved in the budget of online games deployment. Therefore, this project attempts to assess the scope of techniques used within games to limit network traffic through research of technology and game features which may affect gaming traffic. As result of this project, the previous experiments were summarized and procured a list of recommendations for the accuracy game development which can be observed in the last part of this section. However, before the exposition of the final results it is summarized the final conclusions of each analysis and experiment performed in the previous section – 4 Game Traffic Analysis – about the three versions developed for the Pong game which is an arcade game.

The technical analysis is divided into six rigorous tests grouped into four experiments that evaluate the four areas identified in the research process; whereas the QoS analysis is executed through the characteristic observed during the test and a survey completed by the players – Appendix H –

The "Multi Pong" game is a simple online-multiplayer game that only sends the position of the ball periodically, players' movements and control information. Hence, these packets are not very weighty and do not consume much bandwidth; however the majority of the online games used the same system of small periodic information. Therefore the comparative and evaluation of this game was focused on terms of delay, number of packets, and packet loss rate in order to solve and expand results of this study to others similar games.

The first experiment evaluates the protocols TCP and UDP. This analysis is deeply executed due to the influence of communication protocol over the gaming traffic. The election of one or other protocol conditions the network traffic and the game features implemented.

The TCP protocol is a very reliable protocol; however it is also very slow and does not work properly when the devices are slow or the answer is slow. In the case of the Pong game, a limitation in the number of packets is created caused by the RTT time of the clients' packets acknowledgment. Therefore, the number of packets sent to each client is unequal; consequently clients who received a smaller number of packets receive a packet size rate higher due to the accumulation of information, which is not useful for the game, in the following packets.

On the other hand, UDP packet loss rate is subtly higher than TCP packet loss rate; however this rate depends mainly on the access technology used, in which Ethernet seems have advantage over wireless. Moreover, the difference between technologies also may be the cause of delays generated by different types of networks. This question is demonstrated in the last experiment in which the game is tested in a different server allocated in Spain and the traffic of three devices is captured. One of these devices is in the same LAN than the server – Madrid–; whereas the others two devices are situated in different WAN networks – UK and Tenerife –. This test results a clear superiority of the device in the LAN network when the TCP version is run; however users' QoS and game performance increase with the UDP version independently of the network condition.

These facts reveal the superiority of UDP over TCP in this specific game due to the unimportance of the information delayed in time-real applications and higher quality in different environments.

During the rest of the experiments it is possible to observe the main differences between protocols which condition the rest of the features. Therefore, the evaluation of the gaming features was executed taking as reference the TCP version due to its extreme and unfavourable conditions. Following this approach, the graphics and sound effects were analysed in the second experiment; and thus was discovered the slight variation generated by the graphics and sounds effects.

In contrast to the graphics and sounds effects which do not affect the gaming traffic, the ball speed and the increase in number of players, it was demonstrated that these factors increase the gaming traffic. However, this increase is not a problem in this game because the maximum bandwidth consumed is around 20Kbps, in contrast with the 800 Kbps that consumes the majority of the most popular games – WoW or Counter Strike – . On the other hand, the TCP version, as it was commented previously, reduces considerably the amount of packets of the device that does not belong to the server LAN; hence the quality of game is also reduced.

The third experiment attempts to assess a different architecture from the common client-server architecture through the third version of the game implemented with P2P networking architecture. This version reduces considerably the server load; as the client who initiates the game is the one in charge of the packets transmission. However, the use of this architecture may involve serious difficulties of connectivity and security in the game. This version may be vulnerable to cheating by users, and the game LAN may generate problems of connectivity due to the necessity of communication between clients. In order to compare accurately this version with the last version implemented, the server program is run in one of the clients' machines; hence one of the clients is connected in localhost. The traffic is reduced respect to the others versions in the P2P version; furthermore using this system the server may be avoided. However, the issues of security and accessibility to the game are very concerned.

Finally, in terms of quality of service UDP protocol, LAN networks and Ethernet technology seems to provide the higher quality in terms of speed, delays and packets

loss rate. Moreover, these elements are the favourites of the players according with the survey's results.

Recommendations

As summary of the previous conclusions a list of recommendations for the accuracy development of online games is elaborated. However, it is not possible to give a generic list of recommendation for the development of all online games, because each game has concrete necessities. This project is focus on the development of arcade online games; as this genre is the game with higher network requirements due to real time simulation between players.

The recommendations, like the test planning, can be divided into four categories.

Broadband Technology:
- Ethernet/Wireless

Techniques used in the communication
- Protocols UDP/TCP
- Network Architecture Client - Server /P2P

Game features
- Number of players
- Ball speed
- Graphic/ Sound effects

Others
- Type of networks. LAN/WAN

The element which more has conditioned the quality of the game is the election of the communication protocol –TCP and UDP–. This choice together with the type of network may cause a high or low quality of service.

Therefore, the first questions to consider during the game design process are: what type of game will be developed – arcade/ by turns–; how much valuable is the information transmitted; and where the game would be run –LAN/WAN–.

Based on the experiences obtained throughout the development of this project the recommendation is the use of the UDP protocol due to its simplicity and independency of the client connection. Furthermore, this protocol is faster than the TCP protocol. On the other hand, whether the data transmitted are essential, the TCP protocol assures the delivery of that data. Hence, this project only recommends the use of this protocol in extreme case, in which the data transmitted are indispensable.

Regarding technology parameters, the access technology more efficient is Ethernet; however wireless technology also provides an efficient performance. Mobile

technologies could not be studied due to time limitation; nevertheless agreeing with previous publications, this technology provides a QoS worse than the wireless technology.

Focus on the networking architecture, the typical game architecture is the client – server where it is necessary the acquisition of a server: However, whether the game is deployed to play in LAN, the use of P2P architecture may reduce the server cost and bandwidth due to the lower load in the server and reduction of the packets number transmitted. In contrast, this architecture may generate issues of connectivity with other networks and issues of security as this game mode is vulnerable to cheat by the users.

Finally, the game features are important for the players' satisfaction. Graphics and sounds effects does not affect the network traffic; in contrast the ball speed –Packet rate– and the number of players increase considerably the bandwidth which may be limited by the TCP protocol depending on the delay of the clients' ACK. However the essential factor to become the game success, according to the surveys made by the players, is the game logic. In other words the game has to be funny.

This page intentionally left blank

6. FUTURE WORK

This section attempts to support future students and provide ideas for their final projects through the continuation of this project in a large amount of possible ways. Because, as William Faulkner says: "The end of wisdom is to dream high enough to lose the dream in the seeking of it". Consequently, there is still room for improvement; it is just necessary stop, looking out the window, and think.

The studied topic belongs to an immense area from which was only researched a small part. It is always possible to continue improving the game and testing more and different gaming features. Moreover, the game was developed as a Java application and sockets; however it could also be developed using java applets and evaluate the network through the 80 port which use HTTP to communicate. The development through java applets is likely more simple for the reason that communication is performed on a web server; consequently the use of a data base to store and load information results an attractive alternative to the temporary arrays used in this project.

Along the same line, the study of Vlad Nae, Radu Prodan & Thomas Farihger (2011) provides a radical change in the development of online games for computers using a new network architecture based on Cloud Computing Internet as a service (IaaS). This new business model provides the possibility to save money in the acquisition of recourses like servers or bandwidth.

Furthermore, other possibility is wide the range of devices through the inclusion of mobile devices; such as BlackBerry -Java ME- or Google's Android which use a customized version of Java not based on Java ME; or even game consoles– Xbox, PS3 –

Finally, this section closes thanking you for reading this project and I hope that you have enjoyed reading it as much as I enjoy writing it. Moreover, remember what Vince Lombardy said: "The dictionary is the only place success comes before work. Hard work is the price we must all pay for success. I think we can accomplish anything if we are willing to pay the price."

This page intentionally left blank

The End

Finally, I would like to conclude this project with two of my favourite's quotes:

The first one is from the French writer Honoré de Balzac (1799 – 1850) who claims:
"Elegance is the science of doing nothing the same at the rest, seeming to do everything the same way as them."

And, the second one is from Oscar Wilde (1854 – 1900) who argued:
"The experience is the name everyone gives to their mistakes." (Oscar Wilde, 1982, Lady Windermere's Fan)

Bibliography

Almashor, M., & Khalil, I. (2010). *Reducing Network Load in Large-Scale, Peer-to-Peer Virtual Environments with 3D Voronoi Diagrams* [Electronic version]. Journal of High Perfonrmance Computing (HiPC), pp. 1-10. doi: 10.1109/HIPC.2010.5713197

Armitage, G.,Claypool, M., & Branch, P. (2006). *Networking and Online Games. Understanding and engineering multiplayer internet games.* Chichester, England: John Wiley & Sons, Ltd.

Brackeen D. (2003). *Developing Games in Java*. United States of America. New Riders Publishing.

Ceballos, Fco.J. (2003). *Java 2: Curso de programación*. Spain. RA-MA.

Cisco and/or its affiliates. (2011). *Cisco Visual Networking Index: Forecast and Methodology, 2010–2015*. Retrieved February 14, 2012, from Cisco website: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

Davison , A. (2005). *Killer Game Programming in Java*. United States of America. O'Reilly

Deitel, P., & Deitel, H. (2012). *Java$^{tm:}$: How to Program* (9th Ed.).England: Pearson.

Färber, J. (2002). *Network Game Traffic Modelling* [Electronic version]. Journal of NetGames '02 Proceedings of the 1st workshop on Network and system support for games, 53 – 57. doi: 10.1145/566500.566508

Federal Communications Commission.(2010). *FCC 10-148: Seventh Broadband Deployment notice of Inquiry* [Electronic version]. Washington, D.C.

Hall, R., & Novak, J. (2008). *Online Game Development*. Clifton Park, N.Y: Delmar Cengage Learning.

Hannah, A. (2011). *Packet sniffing basics* [Electronic version]. Journal of Linux Journal, 2011(210), 1 – 4. doi: 10.1145/1517494.1517512

Harbour, J. S. (2008). *Beginning Java Game Programming* (2nd.) Retrieved from http://site.ebrary.com/lib/portsmouth/docDetail.action?docID=10228198

LaPointe, D., & Winslow, J. (2001). *Analysing and Simulating Network Game Traffic*. Unpublished undergraduate dissertation, University of Worcester. Worcester. Retrieved

December 21, 2011 from the Worcester Polytechnic Institute Computer Science website: http://web.cs.wpi.edu/~claypool/mqp/net-game/

Lakkakorpi, J., Heiner, A., & Ruutu, J. (2002). *Measuring and characterization of Internet gaming traffic.* Retrieved February 10, 2012, from Citeseerx website: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.99.3848

Lecky-Thompson, Guy W. (2008). *Fundamentals of Network Game Development* Retrieved from http://site.ebrary.com/lib/portsmouth/docDetail.action?docID=10333475

Lin, L., Cao,D., & Yin, X. (2010). *A Study on China's online games development trend* [Electronic version].Journal of Mechanic Automation and Control Engineering (MACE), 2010 International Conference on, 1725 – 1728.
doi: 10.1109/MACE.2010.5536050

Lyon, G. (2010). *Top 10 Data/Packet Sniffing and Analyzer Tools for Hackers* [Electronic version]. Retrieved April 10, 2012 from the internet Geeks website: http://www.internetgeeks.org/tech/hacking/top-10-data-packet-sniffing-analyzer-tools-hackers/

Martin Garcia, L. (2010). *TCPDUMP & LIBPCAP Documentation.* Retrieved April 10, 2012 from the tcpdump website: www.tcpdump.org

McCabe, J.D. (2003). *Network Analysis, Architecture and Design.* (2nd ed.) United States of America: Morgan Kaufmann.

Montoro, M. (2011). *Cain & Abel v4.9.43 released* [Electronic version]. Retrieved April10, 2012 from Oxid.it website: http://www.oxid.it/ca_um/

Mulholland, A., & Hakala, T. (2004). *Programming Multiplayer Games.* Texas: Wordware Publishing, Inc.

Nae, V., Prodan, R., Iosup, A., & Fahringer,T. (2011) *A New Business Model for Massively Multiplayer Online Games.* Retrieved February 2, 2012 from ACM Digital Library website: http://dl.acm.org/citation.cfm?id=1958785

Niemeyer, P., & Knudsen, J. (2005). *Learning Java* (3rd Ed). United States of America. O'Reilly.

Object Management Group, Inc. (OMG). (2012). *UML® Resource Page.* Retrieved April 15, 2012 from UML webpage: www.uml.org

OnlineMBA (2010). *Online Gaming Statistics.* Retrieved February 15, 2012, from Online MBA website: http://www.onlinemba.com/blog/online-gaming-statistics/

Online Marketing Trends. (2011). *The Top 5 Farmville Economics, Statistics and Trivia.* Retrieved April 4, 2012 from the Online Marketing Trends website: http://www.onlinemarketing-trends.com/2011/02/top-5-farmville-economics-statistics.html

Oracle Corporation and/or its affiliates. (2012). *NetBeans IDE 7.1 Feature.* Retrieved from http://netbeans.org/features/index.html

Plusnet Company. (2011). *All about traffic management.* Available: http://www.plus.net/support/broadband/speed_guide/traffic_management.shtml. Last accessed 20th January 2012.

Qadeer, M.A., Iqbal, A., Zahid, M., & Siddiqui, M.R. (2010). *Network Traffic Analysis and Intrusion Detection using Packet Sniffe* [Electronic version]. Journal of Communicacion Software and Networks, 313 - 317 doi: 10.1109/ICCSN.2010.104

Rabin, S. (2010). *Introduction to Game Development*. 2nd ed. Australia; United Kingdom: Course Technology.

Ratti, S., Hariri, B., & Shirmohammadi, S. (2010). *A survey of First-Person Shooter Gaming Traffic on the Internet* [Electronic version]. Journal of Internet Computing, IEEE, 16 (5), 60 – 69. doi: 10.1109/MIC.2010.57

Russell,G., F.Donaldson, A., & Sheepard, P. (2008). *Trackling online game development problems with anovel network scripting language* [Electronic version]. Journal of NetGames '08 Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games, 85 – 90. doi: 10.1145/1517494.1517512

Saldaña Medina, J.M, et al. (2011). *Bandwidth Efficiency Improvement for Online Games by the use of Tunneling, Compressing and Multiplexing Techniques* .Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS 2011, pp.227-234, The Hague, Netherlands. ISBN: 978-161-782-309-1

Savitch, W. (2012*). Java^{tm}: An Introduction to Problem Solving and Programming* (6th Edition). England. Pearson.

Sanders, C. (2007). *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problem.* Retrieved from http://site.ebrary.com/lib/portsmouth/docDetail.action?docID=10183331

Sheffield,B., Fouts, N., Theodore, S., Kastbauer, D., Undersander, E., Cooke, M., Llopis, N., & Lehman, Z. (2012). *Front line awards 2011*. Journal of Game developer magazine, 19 (1), 7-11.

Sharpe, R (2011). *Wireshark User's Guide* [Electronic version]. Ed Warnickle. Retrieved February 24, 2012from Wireshark website: http://www.wireshark.org/docs/wsug_html_chunked/

Stallings, W. (2011). *Data and Computer Communications* (9th ed.).New Jersey: Pearson Education, Inc.

Wang, W., Taekyoung Kwon, T., Choi, Y., Chen, M., & Zhan, Y. (2012). *Characterizing the Gaming Traffic of World of Warcraft: From Game Scenarios to Network Access Technologies* [Electronic version]. Journal of Network IEE, 26 (1), 27-34, doi: 10.1109/MNET.2012.6135853

Zhang, X., Wang, W., Shi, W., Lian,S., & Feng, D. (2012). *Understanding and Analyzing Network Traffic* [Electronic version]. Journal of Network IEE, 26 (1), 4 -5. doi: 10.1109/MNET.2012.6135849

Tutorials and software used throughout the project development.

Aaron. (n.d). *Paddle Battle.* Retrieved from http://www.aaroncox.net/tutorials/arcade/PaddleBattle.html

Casas, F. (2010, April 8). *Pong hecho en Java.* [Web log message] Retrieved from http://www.facebook.com/photo.php?fbid=459755613232&l=acc9294ce5

EON. (2008, September 30). *Juego Ping-Pong.* [Web log message] Retrieved from http://casidiablo.net/implementacion-juego-ping-pong-java/

Jordan82. (2009, May 11). *Tutorial: Creating a simple Pong game* [Web log message] Retrieved from http://www.tutorialized.com/tutorial/Java-Creating-a-simple-Pong-game/52943

Sedgewick, R., & Wayne, K. (2004).*INTRODUCTION TO COMPUTER SCIENCE*. Retrieved from http://introcs.cs.princeton.edu/java/36inheritance/Pong.java.html

Savage,N. (n.d). C programs to analyze the IAT and Packet size. Retrieved from: http://savagen.myweb.port.ac.u

Appendices

Appendix A – APDF form

ONLINE TRAFFIC EVALUATION

Investigate progression of techniques used within games to limit network traffic of the game. Look into how broadband may aid online gaming. You are required to develop an online interactive gaming environment (simple mutli-players game) to experiment and evaluate the different affects that such applications could make on the network traffic (e.g. the use of sound, graphics ...)

To achieve the objectives of the project, the project will include:

1. Research the progression of techniques used within games to limit network traffic

2. Identify game features that may have effect on network traffic

3. Build simple mutli-players game

4. Use the developed game to experiment the different effects of the identified game features

5. Analyse the results of the experiments and evaluate the different effects

6. Recommendations based on the results and evaluation.

Network programming knowledge is required

Appendix B – Statistical and forecast for 2010 – 2015 from Cisco

Retrieved from: (Cisco and/or its affiliates, 2011)

Global IP Traffic, 2010 - 1015

IP Traffic, 2010–2015							
	2010	2011	2012	2013	2014	2015	CAGR 2010–2015
By Type (PB per Month)							
Fixed Internet	14,955	20,650	27,434	35,879	46,290	59,354	**32%**
Managed IP	4,989	6,839	9,014	11,352	13,189	14,848	**24%**
Mobile data	237	546	1,163	2,198	3,806	6,254	**92%**
By Segment (PB per Month)							
Consumer	16,221	23,130	31,592	42,063	54,270	70,045	**34%**
Business	3,930	4,894	6,011	7,357	8,997	10,410	**22%**
By Geography (PB per Month)							
North America	6,998	9,947	12,978	16,116	18,848	22,274	**26%**
Western Europe	4,776	6,496	8,819	11,774	15,187	18,858	**32%**
Asia Pacific	5,368	7,317	9,847	13,341	18,060	24,150	**35%**
Japan	1,414	1,923	2,540	3,283	4,019	4,762	**27%**
Latin America	665	993	1,465	2,158	3,238	4,681	**48%**
Central and Eastern Europe	708	1,004	1,413	1,955	2,700	3,713	**39%**
Middle East and Africa	253	366	550	802	1,235	2,019	**52%**
Total (PB per Month)							
Total IP traffic	20,151	28,023	37,603	49,420	63,267	80,456	**32%**

Source: Cisco VNI, 2011

Exabytes per month as of Year End 2015

	Consumer	Business	Total
Internet	53.3	6.1	59.4
Managed IP	11.8	3.0	14.8
Mobile data	4.9	1.3	6.3
Total	70.0	10.4	80.5

Source: Cisco VNI, 2011

Global Consumer Internet Traffic, 2010–2015

Consumer Internet Traffic, 2010–2015							
	2010	2011	2012	2013	2014	2015	CAGR 2010–2015
By Network (PB per Month)							
Fixed	12,355	17,467	23,618	31,318	40,842	53,282	34%
Mobile	174	399	858	1,654	2,930	4,931	95%
By Subsegment (PB per Month)							
File sharing	4,968	6,017	7,277	8,867	11,040	13,797	23%
Internet video	4,672	8,079	12,146	17,583	24,357	33,620	48%
Web, email, and data	2,393	3,113	4,146	5,325	6,769	8,592	29%
Video calling	308	442	659	905	1,251	1,736	41%
Online gaming	49	68	95	133	187	290	43%
Voice over IP (VoIP)	138	147	153	157	160	168	4%
Other	0	1	1	3	8	11	132%
By Geography (PB per Month)							
North America	3,301	5,000	6,579	8,306	10,012	12,537	31%
Western Europe	3,147	4,360	6,075	8,224	10,841	13,896	35%
Asia Pacific	4,403	6,006	8,142	11,129	15,249	20,758	36%
Japan	638	932	1,317	1,807	2,344	2,968	36%
Latin America	482	735	1,106	1,667	2,577	3,850	52%
Central and Eastern Europe	454	667	971	1,381	1,963	2,805	44%
Middle East and Africa	103	166	286	459	784	1,399	68%
Total (PB per Month)							
Consumer Internet traffic	12,528	17,866	24,476	32,973	43,771	58,214	36%

Source: Cisco VNI, 2011

Definitions:

> The **exabyte**: (derived from the SI prefix exa-) is a unit of information or computer storage equal to one quintillion bytes (short scale). The unit symbol for the exabyte is EB. The unit prefix exa indicates the sixth power of 1000:

1 EB = 1000000000000000000B = 1018 bytes = 1000000000gigabytes = 1000000terabytes

(Retrieved February 24, 2012, from Wikipedia website: http://en.wikipedia.org/wiki/Exabyte)

Appendix C – Classification Generic (by Game Style)

The games may be classified according the purpose of the game and the number of players. Below appear the main four genres:

- MUD (Multi User Dungeon Games).
  It is the most basic example of real multiplayer network game which only consists of a web interface based on text; where the players write their movements following a set of rules. Furthermore, this genre of games was the first kind of multiplayer games (Lecky-Thompson, 2008, p. 28). It is possible to find an example of this genre active from 1974; its name is Dungeons & Dragons. (Retrieved February 16, 2012, from Wikipedia website: http://en.wikipedia.org/wiki/Dungeons_%26_Dragons)

- RPG (Role Playing Game).
  This kind of game can be considered as an evolution of the MUD games, because in both the players are involved in a fantasy world. However, the main difference respect to MUD games is the graphic interface which increases the realism of the game. Some of the most famous RPGs are the Final Fantasy series and Wasteland (Rabin, 2010, p. 37)

- FPS (First Person Shooters).
  This game is an action game in which the user is situated "behind the eyes" of the game figure. The goal of this game is to shoot the rest of characters with a wide variety of weapons. Some instances of this kind of games are Doom or Half-life. (Rabin, 2010, p. 37)

- RTS (Real Time Strategy games).
  The majority of these games have the aim of creating a town with two types of characters, the collectors and the attackers, which usually are the army, and then invading the towns of the other players. The winner in these games is the player which resist the attackers of the enemies and conquest all the towns. Some RTS games can be Command and conquer and Blizzard's Warcraft. (Rabin, 2010, p. 37)

- MMO (see on the background)

Appendix D – Acquisition of the specifications

The following list of Linux commands was used in order to know the specifications of the A2.3 computers – CentOS –.

In the case of Windows devices, it was enough with the access to the control system. However in case of Linux devices, the acquisition of the specifications was executed remotely through the connection SSH and the used of the following commands:

The permissions of super user are available in the case of the server; therefore only in the server is possible the use of commands with the extension of super user – sudo-

Commands	Description
Only in the server	
$ sudo dmidecode –t bios	shows all the information about the bios
$ sudo lshw	Shows an exhaustive and detailed report about the technical specifications of the system. It is also possible to use: $cat /proc/cpuinfo | grep -i “model name" and $cat /proc/cpuinfo | grep -i mhz
Commands for both systems: server and allocated computer	
$ uname	Print system information
	Add: -a, --all; -m, -- machine; -p, --processor; -i, --hardware-platform ; -o, -- operating system;
$ lspci|grep -i vga	Shows information about the graphic card
$ aplay -l	Shows the model of the sound card
$ hwinfo	Shows a report about the current HW of the system
$ glxinfo|grep direct	Shows if the system has graphic acceleration - direct rendering -
$ curlicanhazip.com	Shows the public IP address of the system
$ iwconfig	Shows information about Wi-Fi card
$ java -version	Shows the version of java
$ xrandr | grep '*'	Shows the screen features
cat /proc/meminfo	Shows the memory RAM
$ ethtool eth0 | less	speed of Ethernet

Information retrieved from the next websites:

http://hatteras.wordpress.com/2011/03/31/conocer-el-hardware-con-la-terminal/
http://linux.about.com/library/cmd/blcmdl1_uname.htm
http://www.cyberciti.biz/faq/how-do-i-find-out-screen-resolution-of-my-linux-desktop/
http://www.unix.com/aix/21321-how-know-server-specifications.html

Appendix E – Planning: Project initiation and planning tables (Gantt charts)

The following tables and figures show the initial planning of the project. The activities necessaries to meet the deadlines are represented and organised by tasks, phases and milestones. Finally, it is showed the Gantt charts which summarizes the previous measures chosen for the project planning.

Milestone Plan

Milestone	Description	Milestone date
End of Literature Review	The literature review, necessary for the implementation of the Preliminary review and to start the development phase, is finished	04/03/2012
Preliminary Report	The first approach for the realization of the project is finished. It include the project initiation and the following plans	05/03/2012
1st Version of SW	The first prototype of the game is concluded	19/03/2012
1st Evaluation	The first analysis of the network and the establishment of measures for improvements are completed.	26/03/2012
Final prototypes	The time dedicated for improvements is expired	30/04/2012
Final Evaluation	The evaluation and comparison of all the prototypes are completed	07/05/2012
Finalized Project	Final Report and Logbook are submitted	14/05/2012 (30/05 deadline)
PROJECT DAY	Exposition of the project	1/06/2012

Table Appendix E.0: Planning: Milestone Plan

Activities Plan

Phase	Activities	Description
(1) LITERATURE REVIEW	(1) Analysis of Bibliography	Review of the information available for the development of the project
"(1) "	(2) Background and Previous Game Design Elaboration	Evaluation of: Techniques, technologies and, progression of the previous publications. Analysis of possible tools for developing games
(2) DEVELOPMENT	(3) Prototype Construction	Implementation of the first version of the game
"(2)"	(4) Adding of improvements to the version 1	Adding extras to test the behaviour of the network with those extras
(3) EVALUATION	(5) Evaluation of the Network necessities	Packet sniffer is chosen and the networks, where the game will be tested, are evaluated
"(3) "	(6) Continuous study of the Network and game features	Continue studying the Network parameters and the possible improvements of the game
"(3)"	(7) Evaluation of the first prototype	Capturing the packets and evaluation of the network traffic
"(3) "	(8) Study of the Results	Measure and comparison between results in order to acquire patterns. Moreover, elaboration of recommendations
(4) FINAL PHASE	(9) Complete the Supplies	The Logbook, Final Report, the Poster and the Executive Summary are produced.
"(4)"	(10) Exposition of the project	Prepare the project on Thursday and expose the project on Friday

Table Appendix E.1: Planning: Activities Plan

Tasks and Effort Plan

P	A	T	Effort (days)
1	1	(1) Bibliographic seek	5
1	1	(2) Bibliographic review	5
1	2	(3) Analysis of progression of the publications	5
1	2	(4) Analysis of the technologies which use broadband	5
1	2	(5) Analysis of the techniques for limiting traffic	5
1	2	(6) Evaluation of the identified features of the game	5
1	2	(7) Decide Game Purpose and preliminary tools	5
2	3	(8) Learning of the Language chosen	5
2	3	(9) Design and Implementation	5
2	3	(10) Testing the version 1	2
2	4	(11) Adding more players	5
2	4	(12) Adding sound	10
2	4	(13) Adding graphic	5
2	4	(14) Adding other features	5
3	5	(15) Selection of the Packet Sniffer	2
3	5	(16) Analyse the network features	3
3	6	(17) Study the obtained results with others reports	25
3	6	(18) Analysis of possible improvements	25
3	7	(19) Capture the first traffic of the game	3
3	7	(20) Measures selection	5
3	8	(21) Evaluate the rest of prototypes	2
3	8	(22) Comparison of results	3
4	11	(23) Complete the final report and the logbook	5
4	12	(24) Preparation of the Exposition	4

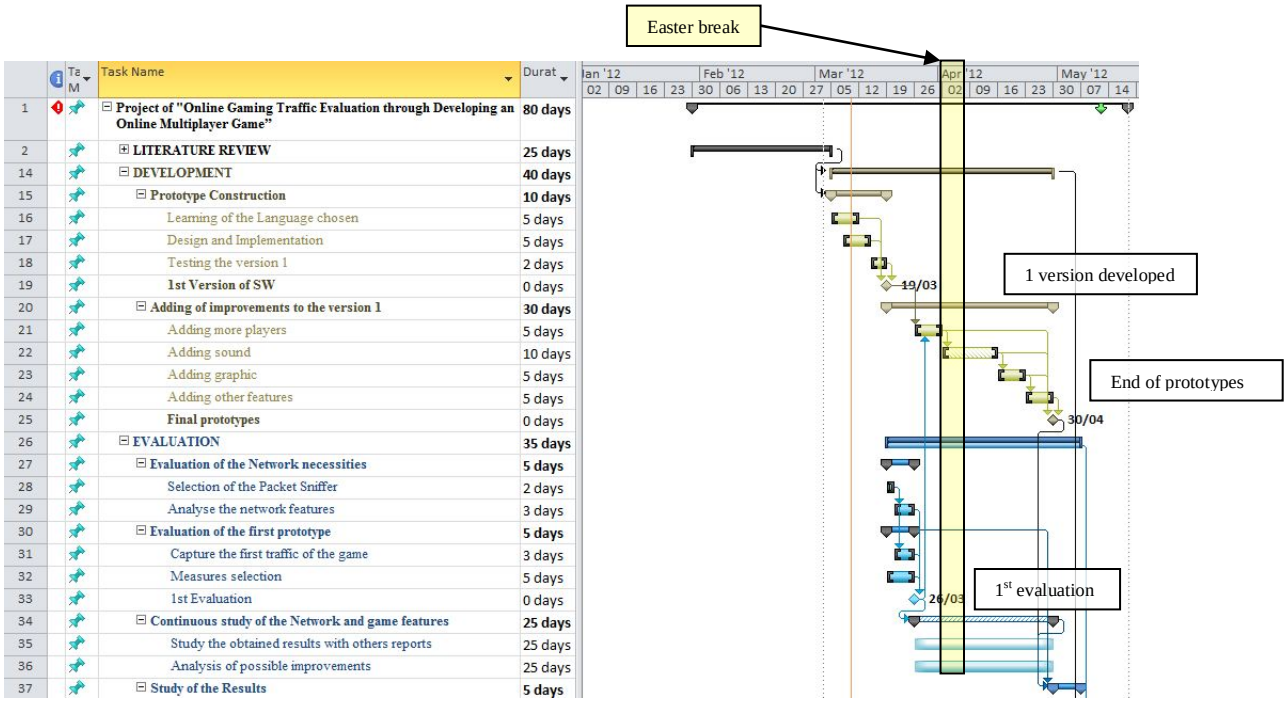
Table Appendix E.2: Planning: Task and Effort Plan
(P=Phase; A= Activities; T=Tasks)

Below is represented the Gantt chart, which include the previous task and activities.

	Ta M	Task Name	Durat	Start	Finish	Predecessors
1		**Project of "Online Gaming Traffic Evaluation through Developing an Online Multiplayer Game"**	**80 days**	**Mon 30/01/12**	**Fri 18/05/12**	
2		**LITERATURE REVIEW**	**25 days**	**Mon 30/01/12**	**Mon 05/03/12**	
14		**DEVELOPMENT**	**40 days**	**Mon 05/03/12**	**Mon 30/04/12**	**2**
15		**Prototype Construction**	**10 days**	**Mon 05/03/12**	**Mon 19/03/12**	**2**
16		Learning of the Language chosen	5 days	Mon 05/03/12	Mon 12/03/12	
17		Design and Implementation	5 days	Thu 08/03/12	Thu 15/03/12	
18		Testing the version 1	2 days	Thu 15/03/12	Mon 19/03/12	
19		**1st Version of SW**	0 days	Mon 19/03/12	Mon 19/03/12	16;17;18
20		**Adding of improvements to the version 1**	**30 days**	**Mon 19/03/12**	Mon 30/04/12	
21		Adding more players	5 days	Mon 26/03/12	Mon 02/04/12	33;19
22		Adding sound	10 days	Mon 02/04/12	Mon 16/04/12	21
23		Adding graphic	5 days	Mon 16/04/12	Mon 23/04/12	22
24		Adding other features	5 days	Mon 23/04/12	Mon 30/04/12	23
25		**Final prototypes**	0 days	Mon 30/04/12	Mon 30/04/12	21;22;23;24
26		**EVALUATION**	**35 days**	**Mon 19/03/12**	**Mon 07/05/12**	
27		**Evaluation of the Network necessities**	**5 days**	**Mon 19/03/12**	**Mon 26/03/12**	
28		Selection of the Packet Sniffer	2 days	Mon 19/03/12	Wed 21/03/12	
29		Analyse the network features	3 days	Wed 21/03/12	Mon 26/03/12	28
30		**Evaluation of the first prototype**	**5 days**	**Mon 19/03/12**	**Mon 26/03/12**	
31		Capture the first traffic of the game	3 days	Wed 21/03/12	Mon 26/03/12	28
32		Measures selection	5 days	Mon 19/03/12	Mon 26/03/12	
33		1st Evaluation	0 days	Mon 26/03/12	Mon 26/03/12	29;32;31
34		**Continuous study of the Network and game features**	**25 days**	**Mon 26/03/12**	**Mon 30/04/12**	**33**
35		Study the obtained results with others reports	25 days	Mon 26/03/12	Mon 30/04/12	
36		Analysis of possible improvements	25 days	Mon 26/03/12	Mon 30/04/12	
37		**Study of the Results**	**5 days**	**Mon 30/04/12**	**Mon 07/05/12**	**25;30;34**
38		Evaluate the rest of prototypes	2 days	Mon 30/04/12	Wed 02/05/12	
39		Comparison of results	3 days	Wed 02/05/12	Mon 07/05/12	38
40		**Final Evaluation**	0 days	Mon 07/05/12	Mon 07/05/12	39
41		**FINAL PHASE**	**9 days**	**Mon 07/05/12**	**Fri 18/05/12**	**26;14**
42		**Complete the last Supplies**	**5 days**	**Mon 07/05/12**	**Mon 14/05/12**	
43		Complete the final report and the logbook	5 days	Mon 07/05/12	Mon 14/05/12	40
44		**Finalized Project**	0 days	Mon 14/05/12	Mon 14/05/12	43
45		**Exposition of my project**	**4 days**	**Mon 14/05/12**	**Fri 18/05/12**	**42**
46		Submission of the Report and Logbook	0 days	Wed 16/05/12	Wed 16/05/12	
47		Preparation of the Exposition	4 days	Mon 14/05/12	Fri 18/05/12	
48		**PROJECT DAY**	0 days	Fri 18/05/12	Fri 18/05/12	47

Figure Appendix E.0: Planning: List of work

First phase: Preliminary Report; development of the first version of the game and first evaluation.



	Te M	Task Name	Durat
1		**Project of "Online Gaming Traffic Evaluation through Developing an Online Multiplayer Game"**	**80 days**
2		**LITERATURE REVIEW**	**25 days**
14		**DEVELOPMENT**	**40 days**
15		**Prototype Construction**	**10 days**
16		Learning of the Language chosen	5 days
17		Design and Implementation	5 days
18		Testing the version 1	2 days
19		**1st Version of SW**	0 days
20		**Adding of improvements to the version 1**	**30 days**
21		Adding more players	5 days
22		Adding sound	10 days
23		Adding graphic	5 days
24		Adding other features	5 days
25		**Final prototypes**	0 days
26		**EVALUATION**	**35 days**
27		**Evaluation of the Network necessities**	**5 days**
28		Selection of the Packet Sniffer	2 days
29		Analyse the network features	3 days
30		**Evaluation of the first prototype**	**5 days**
31		Capture the first traffic of the game	3 days
32		Measures selection	5 days
33		1st Evaluation	0 days
34		**Continuous study of the Network and game features**	**25 days**
35		Study the obtained results with others reports	25 days
36		Analysis of possible improvements	25 days
37		**Study of the Results**	**5 days**

Figure Appendix E.1: Planning: Gantt chart. Phase1

Final phase: Finalization of the project and Project day

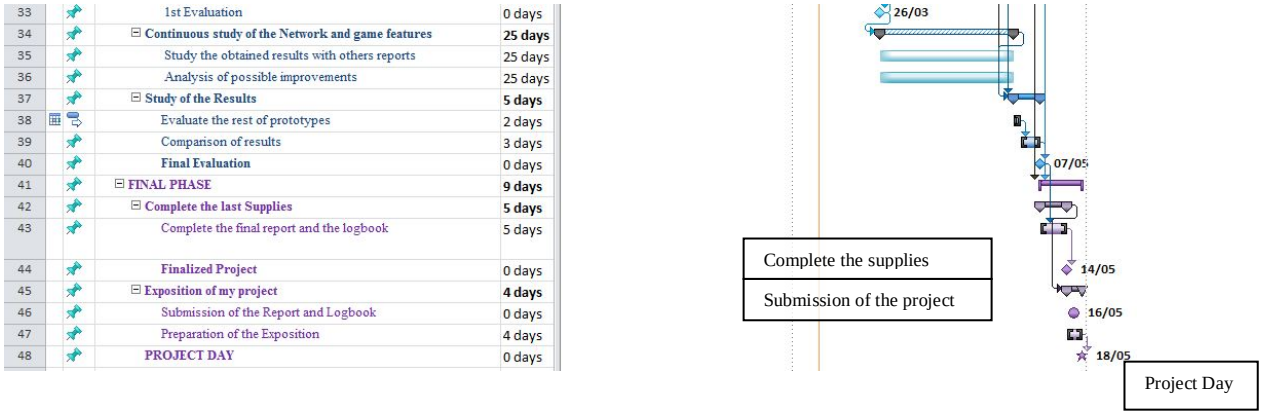


Figure Appendix E.2: Planning: Gantt chart. Final phase

Project timeline

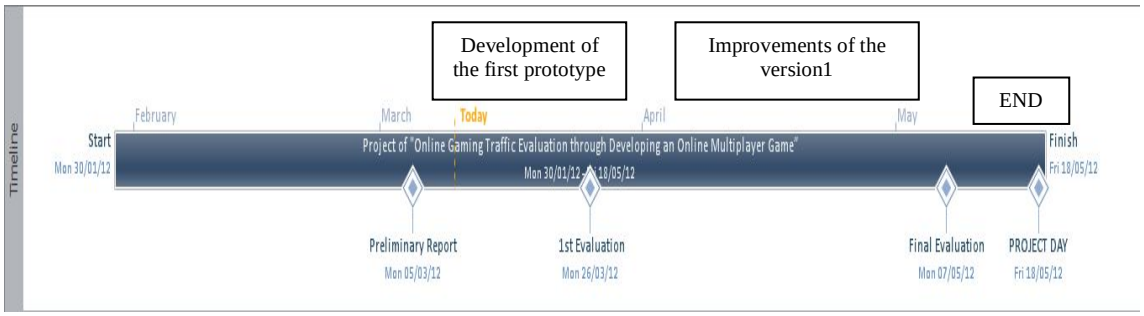


Figure Appendix E.3: Planning: Project time

Appendix F – Code: First approach.

By convention is adopted the following rules in order to fulfil an order in the programming process and avoid errors in the compiling process:

- The method names begin with a lowercase first letter and subsequent words in the name begin with a capital letter. (Deitel, P & Deitel,H, 2012, p. 108)

- The class names begin with a capital letter. (Savitch, 2012, p.265)

- The main class which always must store a method called main has the same name than the name of the packet. (Savitch, 2012, p.265)

- It will be attempted to create variables with names that define their function.

The files are divided into two NetBeans projects belonged to the client and to the server.

Furthermore the files are distributed by different types. The name of the program is in the first line, later it is followed by the packages and finally by the class and Java files.

The files of the game package are the same in both applications; however the rest of files are completely different. Although sometimes it is possible to find similarities between the files that content the packets which are transmitted.

The packets will change with the different versions; however the portal will be the same.

Due to the immense number of lines and versions below is only detailed the name of the files. The code of these files can be observed and studied through the CD provided with this document.

CLIENT

Name of the project: ClientLab

Sources:

Packet: Communication
- Information.java
- Information_game.java

Packet: Game
- Ball.java
- Pallet.java
- GameOn.java
- GameOnline.java
- GameOffline.java

Packet: Resources:
- bGal.wav
- bhit_sport.wav
- bk_sport.wav
- bl_sport.jpeg
- blue_ball.png
- blue_boot0.png
- blue_boot1.png
- dball.png
- del0.png
- del1.png
- logo_blue.png
- logo_porteria.png
- logo_porterias.png
- w_hit.wav
- w_ocean.wav
- w_score.wav
- WaterBK.jpeg

Packet: ClientLab
- ClientPong.java
- Main.java
- ContainerWindow.java
- CreateGame.java

SERVER

Name of the project: ServerLab

Sources:

Packet: server.communication
- Information.java
- Information_game.java

Packet: Game
- Ball.java
- Pallet.java

Packet: ServerLab
- Main.java
- MainServer.java
- Pong_Server.java

Appendix G – UML Diagrams

The UML class diagram indicates the name of the methods and attributes of the main classes used in the game developed.

Class diagram:

Information class client_v1:

Information
- Connection connection - PacketChat chat_pck - UpdateGames update_pck - AckServer ack_pck - NewGame game_pck - JoinGame join_pck - int port_game - int number_Games - int max_games = 50 - GameCreated [] list_games - String buffer_out - String buffer_in - Formatter output - Scanner input
+ Information() + createList(Scanner input) + SerializationPck (Type_pck type_pck) + DeserializationPck (String message)

JoinGame
- int ID
+ JoinGame() + getID() + setID()

NewGame
- string name Game - int numberPlayers - int duration - int gameID - fqBall
+ newGame() + GET/SET

PacketChat
- string name - string message
+ PacketChat() + GET / SET

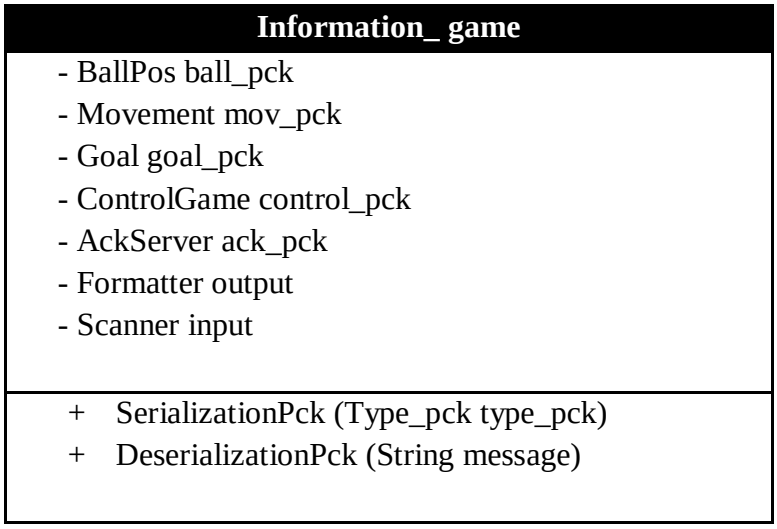
AckServer
- Type_pck - String info
+ AckServer() + GET / SET

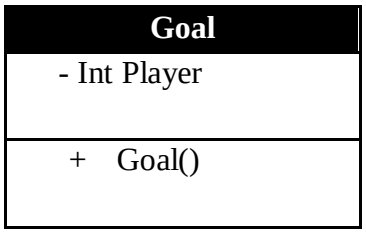
Game package:

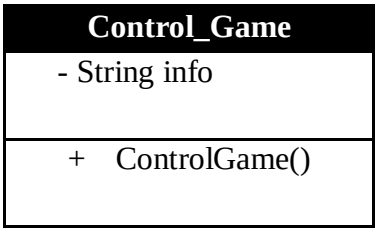
PacketChat
- string name - string message
+ PacketChat() + GET / SET

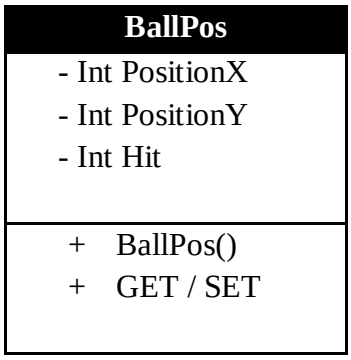
Pallet
- int PositionX = 25 - int PositionY = 215 - int dimX = 25 - int dimY = 100
+ Pallet() + GET / SET

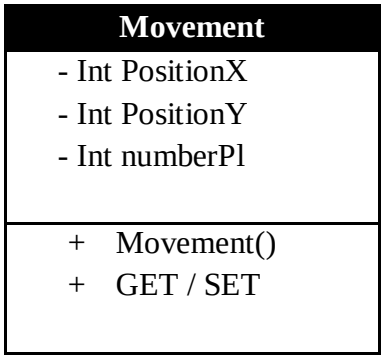
Information Game:











Appendix H – Survey to evaluate the preferences of the users.

University of Portsmouth

"Online Gaming Traffic Evaluation through Developing an Online Multiplayer Game"

MULTI_PONG GAME SURVEY

Dear player,
In first place, I would like to thank you by the time which you have spent playing to Multi Pong game; moreover I would like to ask your collaboration in order to understand the users' preference,

Information	
PC name:	
IP address location Network connection	

Example:

Chipmunk; my_laptop_David.
148.197.27.14
Laboratory A2.3
Ethernet.

To know the IP address write in the terminal ipconfig (windows) or ifconfig (Linux) or write in Google "my IP".

What was your first impression of the game portal?

Was the style of the game pleasing to you? (Blue colour, game modes, chat, text font)

How many times have you played the game?

Number of times in Offline game =
Number of times in online game =

What is your favourite mode of game – Simple; Football; Water - ? Why?

Does the game seem to you slow?

What version were the slowest?

What is your qualification from one to ten for the next versions of the game?
V1: V2: V3:

What is the element of the game more valuable for you? (The number of players, the graphics, the sounds or the game logic)
Explain why.

Thank you for your collaboration.

Appendix I – Wireshark Generic Information

Table Appendix I.1: Test 2. Trace Network Statistics

Trace Network Statistics (2)	TCP	UDP
Server Host	148.197.34.67	148.197.34.67
Total Packets	38,737	30,384
I / O	19,335 / 19,402	2,873 / 27,511
Mean Packet Size	74.625 bytes	56.959 bytes
I / O [bytes]	67.120 / 83.376	60.039 / 56.637
Mean Bandwidth	18.38 Kbps	11.16 Kbps
I / O [Kbps]	8.10 / 10.31	1.11 / 10.21
Mean Packets rate	31.531 PPS	25.075 PPS
I / O [PPS]	15.738 / 15.827	2.731 / 23.076
Client - laptop -	148.197.67.100	148.197.67.100
Total Packets	10,059	14,967
I / O	5,101 / 4,958	13,704 / 1,263
Mean Packet Size	75.726 bytes	59.605 bytes
I / O [bytes]	94.028 / 56.897	60.013 / 55.183
Mean Bandwidth	4.96 Kbps	5.75 Kbps
I / O [Kbps]	3.14 / 1.84	5.39 / 0.45
Mean Packets rate	8.385	12.353 PPS
I / O [PPS]	4.276 / 4.133	11.496 / 1.042
Client - chipmunk -	148.197.27.14	148.197.27.14
Total Packets	28,695	15,191
I / O	14,276 / 14,419	13,576 / 1,615
Mean Packet Size	73.432 bytes	59.534 bytes
I / O [bytes]	79.652 / 67.273	60.013 / 55.503
Mean Bandwidth	14.41 Kbps	6.03 Kbps
I / O [Kbps]	7.78 / 6.63	5.51 / 0.60
Mean Packets rate	25.119 PPS	12.972 PPS
I / O [PPS]	12.498 / 12.622	11.748 / 1.379

Table Appendix I.2: Test 3. Trace Network Statistics

Trace Network Statistics (3)	TCP	UDP
Server Host	148.197.34.67	148.197.34.67
Total Packets	61,532	27,128
I / O	30,749 / 30,789	1047 / 26,083
Mean Packet Size	73.270 bytes	56.715 bytes
I / O [bytes]	66.023 / 80.512	60.185 / 56.831
Mean Bandwidth	48.37 Kbps	21.63 Kbps
I / O [Kbps]	21.78/ 26.60	0.89 / 22.47
Mean Packets rate	84.497 PPS	48.806 PPS
I / O [PPS]	42.228 / 42.282	1.884 / 50.831
Client - laptop -	148.197.67.100	148.197.67.100
Total Packets	3,295	4,067
I / O	1,683 / 1,612	3,510 / 557
Mean Packet Size	72.066 bytes	59.462 bytes
I / O [bytes]	86.559 / 56.934	60.204 / 54.786
Mean Bandwidth	3.72 Kbps	4.79 Kbps
I / O [Kbps]	2.33 / 1.44	4.21 / 0.60
Mean Packets rate	6.607 PPS	10.306 PPS
I / O [PPS]	3.438 / 3.232	8.942 / 1.411
Client - chipmunk -	148.197.27.14	148.197.27.14
Total Packets	7,893	4,078
I / O	3,900 / 3,993	3,664 / 414
Mean Packet Size	73.773 bytes	59.759 bytes
I / O [bytes]	79.009 / 67.868	60.212 / 55.824
Mean Bandwidth	8.96 Kbps	4.85 Kbps
I / O [Kbps]	4.77 / 4.19	4.46 / 0.47
Mean Packets rate	15.629 PPS	10.386 PPS
I / O [PPS]	7.723 / 7.907	9.473 / 1.070

Table Appendix I.3: Test 5. Trace Network Statistics

Trace Network Statistics (5)	TCP	UDP
Server - chipmunk -	148.197.27.14	148.197.27.14
Total Packets	21,912	11,117
I / O	10,954 / 10,952	918 / 10,199
Mean Packet Size	73.484 bytes	56.684 bytes
I / O [bytes]	67.346 / 79.634	60.013 / 56.581
Mean Bandwidth	16.92 Kbps	6.81 Kbps
I / O [Kbps]	7.75 / 9.17	0.59 / 6.24
Mean Packets rate	29.476 PPS	15.533 PPS
I / O[PPS]	14.735 / 14.733	1.266 / 14.115
Client - chipmunk -	127.0.0.1	127.0.0.1
Total Packets	21,934	11,116
I / O	-	-
Mean Packet Size	73.497 bytes	56.529 bytes
I / O [bytes]	-	-
Mean Bandwidth	16.43 Kbps	6.63 Kbps
I / O [bytes/sec]	-	-
Mean Packets rate	28.616	15.007 PPS
I / O[PPS]	-	-
Client - chipmunk -	148.197.27.70	148.197.27.70
Total Packets	21,902	11,208
I / O	10,950 / 10,952	10,198 / 917
Mean Packet Size	73.773 bytes	59.707 bytes
I / O [bytes]	79.637 / 67.343	60.016 / 55.846
Mean Bandwidth	17.16 Kbps	7.44 Kbps
I / O [bytes/sec]	9.30 / 7.86	6.83 / 0.58
Mean Packets rate	29.889 PPS	15.948 PPS
I / O[PPS]	14.943 / 14.947	14.562 / 1.319

Table Appendix I.4: Test 6 Trace Network Statistics

Trace Network Statistics (6)	TCP	UDP
Madrid_Server	192.168.1.7	192.168.1.7
Total Packets	6,580	9,964
I / O	3,137 / 3,443	1,270 / 8,694
Mean Packet Size	79.556 bytes	57.310 bytes
I / O [bytes]	64.840 / 92.965	60.791 / 53.802
Mean Bandwidth	10.25 Kbps	11.95 Kbps
I / O [Kbps]	3.98 / 6.26	1.62 / 10.51
Mean Packets rate	16.484 PPS	26.694 PPS
I / O [PPS]	7.859 / 8.625	3.402 / 23.679
Client - laptop -	148.197.67.100	148.197.67.100
Total Packets	2,192	4,893
I / O	1,111 / 1081	4,762 / 131
Mean Packet Size	89.365 bytes	59.937 bytes
I / O [bytes]	119.521 / 58.371	60.049 / 55.885
Mean Bandwidth	3.53 Kbps	6.46 Kbps
I / O [Kbps]	2.39 / 1.18	6.29 / 0.17
Mean Packets rate	5.054 PPS	13.785 PPS
I / O [PPS]	2.562 / 2.577	13.418 / 0.383
Client -Madrid_laptop -	192.168.1.5	192.168.1.5
Total Packets	4,348	4,972
I / O	2,284 / 2064	3,838 / 1,134
Mean Packet Size	71.989 bytes	59.099 bytes
I / O [bytes]	81.353 / 61.627	60.059 / 55.850
Mean Bandwidth	5.23 Kbps	7.41 Kbps
I / O [Kbps]	3.11 / 2.13	5.81 / 1.63
Mean Packets rate	9.304 PPS	16.051 PPS
I / O [PPS]	4.888 / 4.417	12.390 / 3.744
Trace Network Statistics (6.1)	**TCP**	**Server**
Client -Tenerife_laptop -	192.168.1.128	192.168.1.5
Total Packets	2,253	4,443
I / O	1,137 / 1,116	2,174 / 2269
Mean Packet Size	82.984 bytes	87.745 bytes
I / O [bytes]	106.040 / 59.496	65.699 / 108.869
Mean Bandwidth	2.51 Kbps	4.46 Kbps
I / O [Kbps]	1.62 / 0.89	1.63 / 2.82
Mean Packets rate	3.871 PPS	6.503 PPS
I / O [PPS]	1.954 / 1.918	3.182 / 3.321

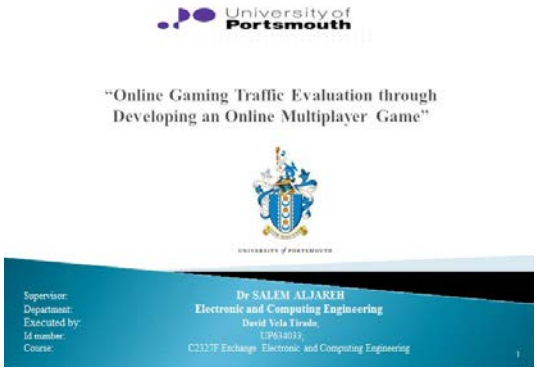
Appendix J – Network Abstraction.

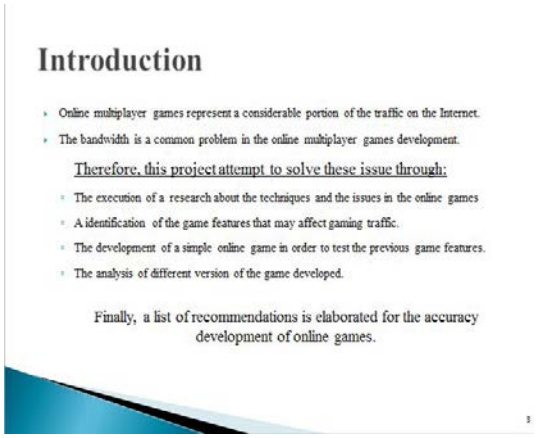
The Wireless network of the university the Portsmouth is separate from the LAN of the A2.3laboratory. Therefore, it is impossible to establish a direct connection between the laptop device and the A2.3 computers.

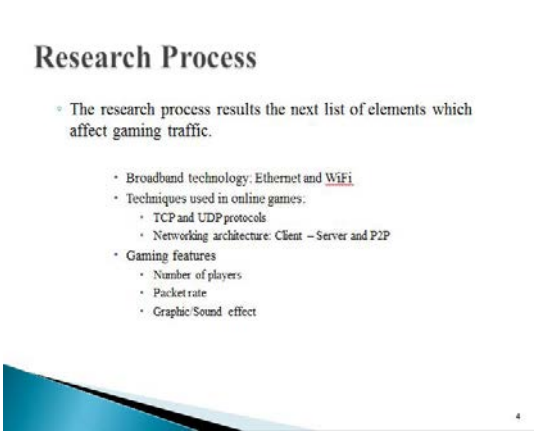
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\David>tracert 148.197.34.67

Tracing route to project1.ee.port.ac.uk [148.197.34.67]
over a maximum of 30 hops:

  1     6 ms     3 ms     3 ms  148.197.143.254
  2   142 ms     3 ms     3 ms  192.168.192.169
  3     3 ms     4 ms     3 ms  project1.ee.port.ac.uk [148.197.34.67]

Trace complete.

C:\Users\David>ping 148.197.34.67

Pinging 148.197.34.67 with 32 bytes of data:
Reply from 148.197.34.67: bytes=32 time=97ms TTL=62
Reply from 148.197.34.67: bytes=32 time=3ms TTL=62
Reply from 148.197.34.67: bytes=32 time=3ms TTL=62
Reply from 148.197.34.67: bytes=32 time=2ms TTL=62

Ping statistics for 148.197.34.67:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 97ms, Average = 26ms

C:\Users\David>ping 148.197.27.14

Pinging 148.197.27.14 with 32 bytes of data:
Reply from 148.197.27.14: Destination host unreachable.
Reply from 148.197.27.14: Destination host unreachable.
Reply from 148.197.27.14: Destination host unreachable.
Reply from 148.197.27.14: Destination host unreachable.

Ping statistics for 148.197.27.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\David>
```

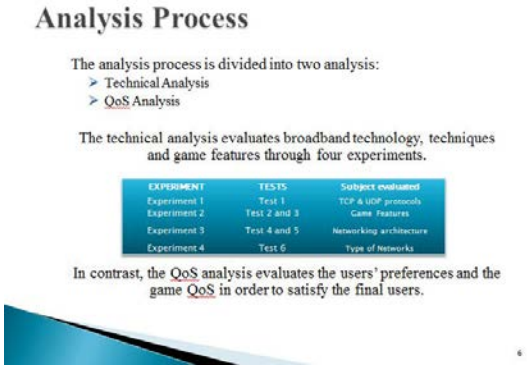
Figure Appendix J. Test 6 Attempt to communicate laptop with chipmunk.

Appendix K – Presentation.









Analysis Process (I)

First Experiment: TCP vs UDP

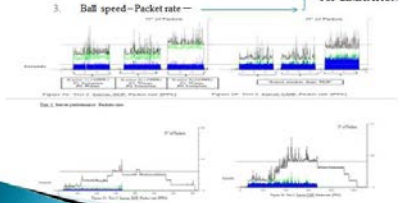
The TCP version presents low QoS due to delays in the packet acknowledgment.

7

Analysis Process (II)

Second Experiment: Game features:

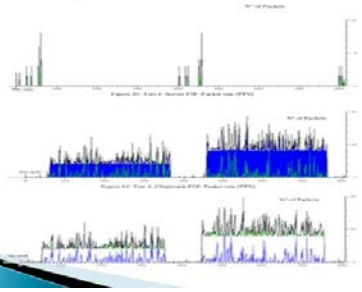
1. Sounds & Graphics effects → NOT EFFECT
2. Number of players → TCP LIMITATION IN LAPTOP CLIENT
3. Ball speed – Packet rate →



8

Analysis Process (III)

Third Experiment: Networking architecture

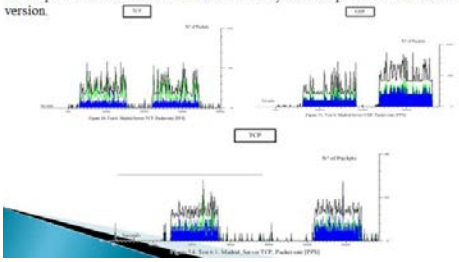


9

Analysis Process (IV)

Fourth Experiment: Type of Network

This experiment reveals the main cause of the delays and bad performance of the TCP version.



10

Conclusions

- Experiment 1 reveals the superiority of the UDP protocol over the TCP protocol.
  - TCP – delays, accumulation of information. ACK limitation.
  - UDP – higher packet loss rate. Faster. Score inconsistences.
- Experiment 2 .
  - Graphics and Sound effects does not affect the gaming traffic
  - Packet rate – Ball speed
  - Number of players.
- Experiment 3.
  - P2P architecture has a lower number of packets with a excellent quality; however has several limitations respect to C/S architecture – Security and connectivity -
- Experiment 4
  - LAN / WAN. The difference between networks is the main cause of the packet acknowledgment delays in the TCP version.

11

Recommendations

The recommendations can be divided into four areas:

- Broadband technology. **Ethernet**/Wireless/Mobile
- Techniques
  - Protocols. **TCP**/ **UDP**.
    UDP is clearly more recommendable for developing arcade games.
  - Architecture. C/S or P2P.
    C/S is the typical; however through the P2P architecture is possible to reduce server cost at the cost of reducing the security and increasing possible issues of connection.
- Game features. The **game speed** and the **number of players** increase the bandwidth; however the protocol selected may limit the gaming traffic reducing the quality of the game. Regarding the **Graphics and sounds effects** are a good feature to improve the game without affect the network traffic.
- **LAN / WAN.** The UDP version has a high performance in both networks; however the TCP version may generate delays in the WAN networks.

12

Thank you

13

Appendix L – CD content

The cd included in the report contains the following information and files:

- Electronic copies of:
  - The report
  - The appendix
  - The cover
  - The presentation
  - The Executive Summary

- Game developed – file: MULTI PONG
  This file contains the three versions of the three games.
  - The version three is the last one and it is divided into the following files:
    - App – executable of the games
    - Game_versions – NetBeans projects + their executable.

- C programs from Nick Savage – file: Wiresark C programs Nick Savage

- Test executed – file: Wireshark captures.
  This file is divided into the sixt test executed, the pictures obtained from the tests and the excel files which contains the charts of each test.