POLITECNICO DI MILANO

Master of Science in Computer Engineering

Dipartimento di Elettronica e Informazione

POLO REGIONALE DI COMO

Networking Systems Aimed to Games Platforms

Supervisor:

Prof. Pier Luca Lanzi

Co-Supervisor:

Enrico Tabet

Master Graduation Thesis by:

Pedro Monteiro KAYATT   733825

Academic Year 2009-2010

POLITECNICO DI MILANO

Laurea Specialistica in Ingegneria Informatica

Dipartimento di Elettronica e Informazione

POLO REGIONALE DI COMO

Sistemi di Reti per gli Piataformi di Videogiochi

Relatore:

Prof. Pier Luca Lanzi

Co-Supervisor:

Enrico Tabet

Tesi di Laurea di:

Pedro Monteiro KAYATT 733825

Anno Academico 2009-2010

Acknowledgements

I dedicate this thesis to my family, which has been supporting me since the beginning. Through long periods of time away from home and long Video-Chat conversations, all the laughs that made me stronger.

Also need to thanks my friends that have never forgotten about me. The old ones that are still going in my house in Brazil thinking that I am going to be back already, and the new ones that have been here in a strange country fighting to live their dreams.

Thanks to my mom, Cristina, that always shows me the right and the wrong and how I must follow my heart and not only my head. To my dad, Cesar Augusto, who is my example of knowledge and my inspiration to become and suppress, with all the right advices showing me that to love is to take care.

To my brothers; Renato who has always been an example of a warrior for me, since the birth fighting against a handicap, and showing the world that he can go further. Thank you to show me that nothing is impossible. And to Ricardo, a big guy with a big heart, always pushing harder than it is, training in the army and doing the college, you are crazy brother.

To my girlfriend; Keila Keiko who has supported and helped me in so many ways, teaching me another way of thinking and giving me hope about great changes. Thanks for the dreams.

To my grandparents, going beyond and learning how to handle with computers and appearing in video conversations, definitely I can see from where my parents take.

To Novecento Games and all my colleagues there, learning and doing our best trying to achieve a state of art over video-games, all the jokes, the snow storms, the coffees and the good will to try to understand my poor Italian, grazie mille ragazzi!

In memory of Carlos Francisco Kayatt.

“Even though I walk through the
valley of the shadow of death,
I will fear no evil, for you
are with me; your rod
and your staff, they
comfort me.”

Psalm 23:4

Sistemi di Reti per gli Piataform di Videogiochi

Pedro Monteiro Kayatt

Luglio 2010

Sommario

Il campo dei videogiochi si è evoluto, diventando ogni anno, un caso più importante di studio. Molti gli miglioramenti tecnologici sono stati fatti al fine di sostenere la comunità gamer e degli affari. Con un fatturato che già hai superato il mercato di Hollywood e il algoritmi di computer prendendo piu spazio su molte conferenze, infine il soggetto si è presentato con serietà.

Aggiungendo l'avvento dell'Internet, l'aumento della larghezza di banda e l'adozione sulla maggior parte delle case in tutto il mondo; il servizio on-line relative al gaming è diventato un must. In questa tesi c'è un'analisi delle sistemi di reti per Videogiochi, come sono organizzati e quali differenze possono essere trovati in ognuna di queste, con il fuoco su come si propongono di sopprimere le barriere che separano il giocatore comuni ai giocatori on-line.

Tuttavia si presenta parti di un progetto che è stato sviluppato presso a Novecento Games, una società di videogame italiano, dove i punti comuni di questi servizi di rete sanno stati assemblati nel modo di costruire una biblioteca comune.

Networking Systems Aimed to Games Platforms

Pedro Monteiro Kayatt

July 2010

Abstract

The videogames field has been evolving and becoming each year a more serious case of study. Many technological improvements have been done in order to support the gamer community and business. With revenue overcome the Hollywood market and the computer algorithms getting space on many conferences, finally the subject has presented it as serious it is.

Adding that to the advent of the Internet, the increased evolution of bandwidth and the great adoption of it on most of the homes across the globe, the online service related to gamming has become a must. In this thesis has an analysis of the actual networking gaming services, as how they are organized and which differences can be found in each one of them focusing on how they propose to suppress the barriers that separate the common player to the on-line player.

Nevertheless is presented parts of a project that was developed at Novecento Games, an Italian videogame company, where the common points of these networking services where assemble in the way to build a common library.

Table of Contents

List of Images

List of Tables

Glossary of Acronyms

Acronyms	Meaning
E3	Electronic Entertainment Expo
US	United States
ITU	International Telecommunication Union
GFWL	Games for Windows Live
HD	High Definition
UK	United Kingdom
LAN	Local Area Network
XBLM	Xbox Live Marketplace
SCE	Sony Computer Entertainment
PS3	PlayStation 3
PSP	PlayStation Portable
ID	Identification
XMB	Xross Media Bar
AP	Access Point
SDK	Software Developer Kit
P2P	Peer-to-Peer
MMORPG	Massively Multiplayer Online RPG
RPG	Role Playing Game
COD	Call of Duty
MW	Modern Warfare

DS	Nintendo DS Handheld
Wii	Nintendo Wii Console
NWC	Nintendo Wi-Fi Connection
WPA	Wireless Application Protocol
WEP	Wired Equivalent Privacy
DLC	Downloadable Content
VE	Virtual Environments
FPS	First Person Shooter
PC	Personal Computer
IP	Internet Protocol
NAT	Network Address Translation
STUN	Session Traversal Utilities for NAT
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
ACK	Acknowledge
CEO	Chief Executive Officer
FSM	Finite State Machine
VPN	Virtual Private Network
ISP	Internet Service Provider

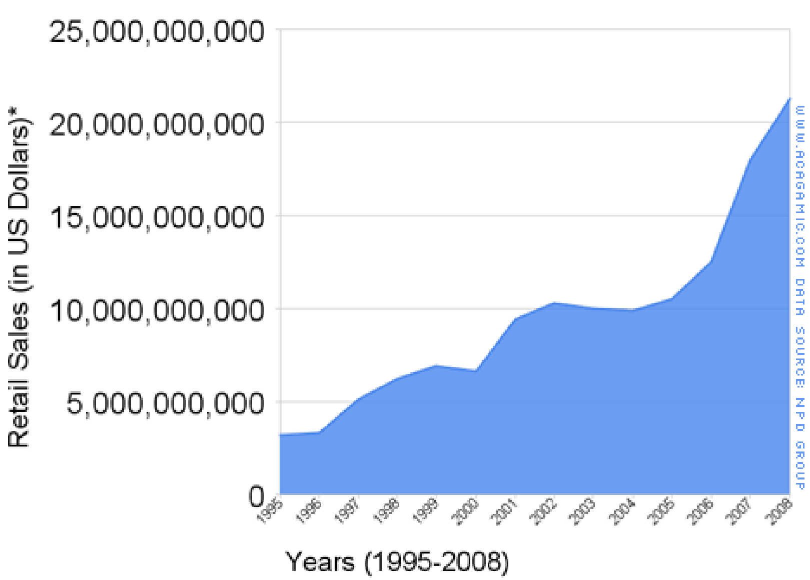
1 Introduction

1.1 Motivation

Not long time ago the internet was not even imagined, a virtual network that can connect everyone making possible to do the exchange of the most widely variations of media. Nevertheless, just after the boom of the Internet [I] rapidly new technologies start to rise in order to fulfill the need to be always connected. The worldwide network then became a certain, every desk computer should be connect to the internet otherwise it would be useless, in the end "what are computers for if they cannot be connected?"

Thus was held another "Connection Revolution": the Wi-Fi. Not far from being connect, people now can (and want) to be wireless connected. Nothing physical holds you, the Internet is now Anytime/Anywhere. Through these evolutions, the Media has been changing dramatically in order to follow the abrupt innovations.

In the mean time the game industry has changed from "water to wine", a small business in the 80's became an industry that holds approximately 42 billion's dollars (source: *IbisWorld*) suppressing the Hollywood market and have set-ups on the houses of most of families in the world.



* Sales of US video games, which includes portable and console hardware, software and accessories.

Figure 1 - Video Game Industry Revenue (US)

Projects that used to be hold in suburbia's garages, now take an average of 3 years [II] and with a budge around 5 to 20 thousand dollars [III]. Therefore this "media" also has been suffering from changes caused by the Networking's Revolutions, the online multiplayer started to be widespread in the year of 1995-1996 through the game called Quake produced by the iD Software (see www.idsoftware.com).

From there and after the multiplayer games start to be a fever, every game had the NEED of have the option "Play Online". Gamers start to spend more and more hours playing all night long. Having this in mind the biggest Game's companies developed complex systems that can support thousand of players as well as different games.

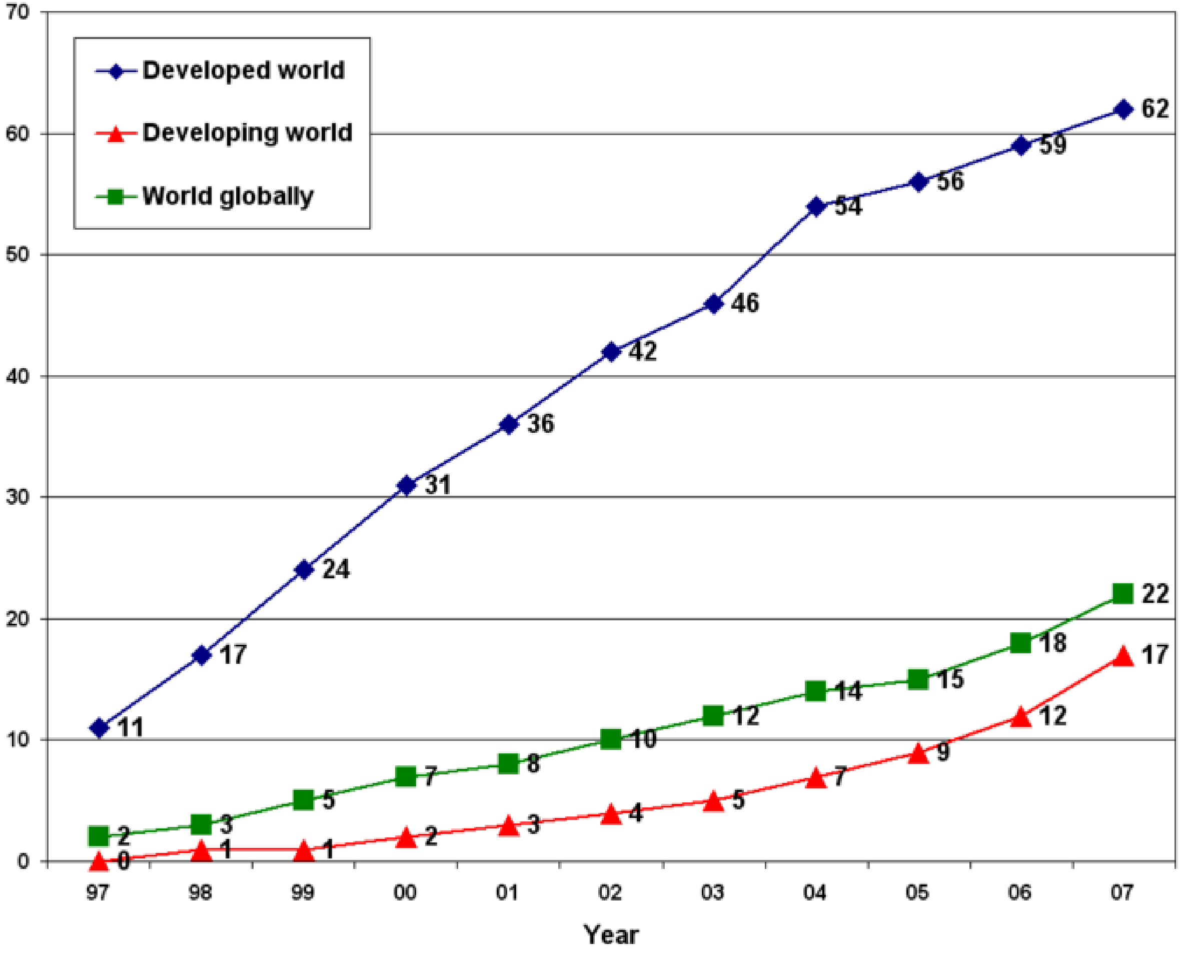


Figure 2 - Internet User per 100 inhabitants (Source: ITU)

1.2 Definition of Problem

With that being said that, it is obvious that the online market is in expansion, and most of all, a big slice of it is composed by inexperienced users that have just start the basics of networking communications at few years.

The focus of our problem will be in provide a simple system that have a very fast learning curve, nevertheless it is robust enough to expand the online experience to it limits. The main idea is to satisfy both the new market as the standard gamers.

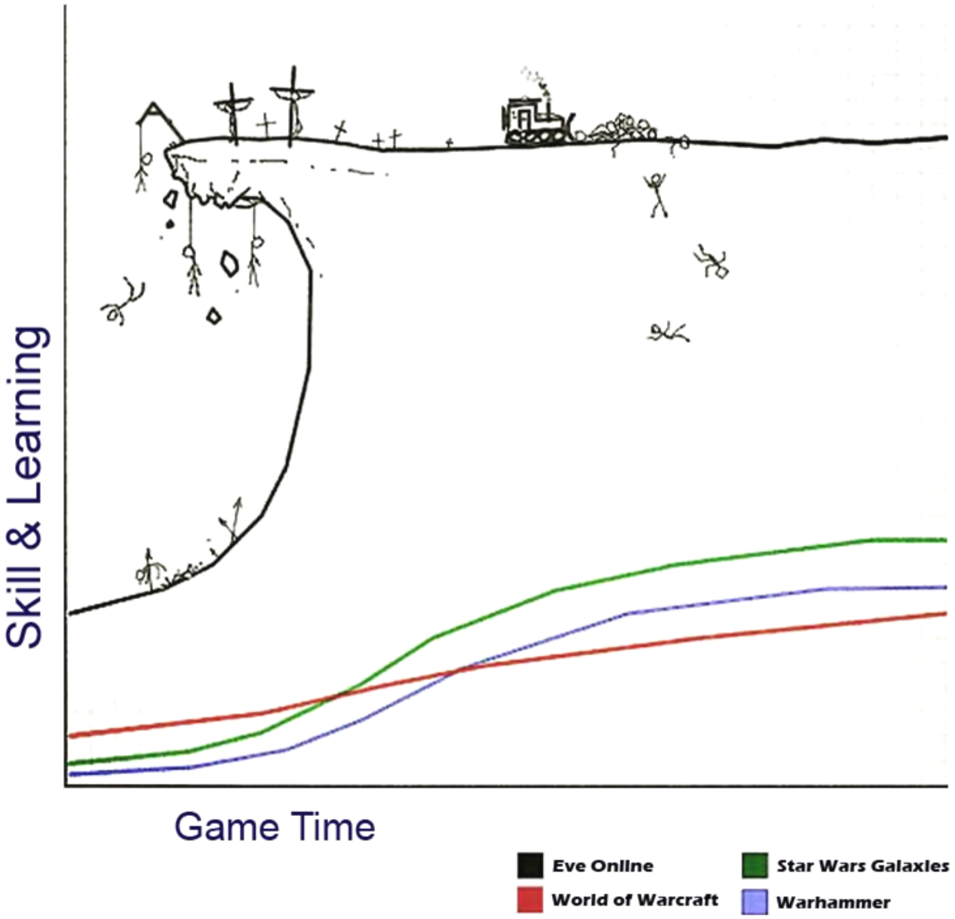


Figure 3 - Example of Learning Curves of Games

Furthermore, there is a big problem involving the development of a multi-platform online application, as a game. As we will be exposed after, each system has its own way to handle with networking packages and connections, as well as protocols (handshaking, encryption, etc…).

Beyond that, even with the continuous evolution of the internet and the increasing of speed and quality of services, a online application is always under the delay problem, what is worst related to games leading the user to a frustrated experience, and afterward, to a denial of the service.

1.3 Solution Strategy

As solution it has been decided that the networking actions must be expressed in the simplest way possible turning into an easy task to both client and developer to access this functions. The first step of the project is then to recognize the most basic actions into the different networking systems of the actual market.

The next step will be to analyzing the algorithms and methods that each one of the systems provides, and to decide the most suitable ones to take then as similar. Using the best practices of these systems an interface must be defined.

It is a must to remember that this system must be tested in a real application, having this in mind a sample game will be developed (or adapted) in order to invoke the interface. The functionality of it must be tested not only in a closed environment, but also in the open internet, as will be further detailed.

1.4 Structure

The structure of this document is defined as followed:

- Chapter 1 is the Introduction;
- Chapter 2 is the State of The Art, which brings relevant information in order to give an update data of available technologies and network systems that have been used.
- Chapter 3 is the Methodology, where steps of development will be taken to exemplify how our problem can be solved;
- Chapter 4 exposes the results achieved. Analyzing data and solutions acquired from tests, as well as this ones have been established;
- Chapter 5 is the conclusion. It presents the conclusion of this project and suggests how it could be further developed.

2 State of the Art

2.1 Xbox Live

Figure 4 - Xbox Live Logo

2.1.1 Introduction

The Xbox Live, also referred as simply Live, is the oldest Gaming Online Service analyzed into this project and it is the only one which at the beginning have charged his users, now there are two versions: Gold (which is charged) and Silver (free), it differences will be catch later on. Live was designed to the first console from Microsoft, the Xbox, and it was one of the killing features of it. The Live logo (Figure 4) is printed on the cover of games that have access to it.

Taking profit of the increasing adoption of broadband connection (in 2001), the Xbox console (Figure 5) was built with a standard Ethernet Port (10/100) and an internal hard-disk drive, already predicting the launch of the Live system. As promised by Microsoft, the service was once showed in the 2002's E3 and in 15th November it was finally available for the users.

Since that the Xbox Live have been suffering a lot of updates, it is constantly under the participation of Beta Testers that are users that accepted this task and Microsoft reward then with gift as t-shirts, free games, memory cards, etc. in the exchange of features feedback and service improvement.

At the time of the launch of its last console, Xbox 360 (Figure 6), the Live was already evolved to another level. At beginning it has basically a friends list, instead of at the date's launch it was possible to watch other gamers playing, competing against each other, this was granted by the 5000th patent of Microsoft.

The number of users subscripted to the Xbox Live is also a criterion to consider, on January 6th, 2009, they had 17 million members and Microsoft has announced on February 5th, 2010, that they have reached 23 million members.

It must be remember that Microsoft provides also the Games for Windows - Live, also kwon as GFWL, that uses a similar system to that one on Xbox, but it does not charge the users and will not be on the scope on this study.

Figure 5 - Microsoft Xbox Console

Figure 6 - Microsoft Xbox 360 Console

2.1.2 Features

Gamercards: it is basically a business card for a gamer: displaying the player's nickname (gamertag); a *MOTTO*, which is a verbal representation of who you are online; a gamer picture (usually about one of his games), a small biography, etc.

Virtual Avatar: an avatar created by the user that should resembles him into the virtual world.

Game achievements: they are a kind of tasks that the player should accomplish when playing determined games.

Gamerscores: for each achievement gathered by the gamer it will add points (Gamerscore points) that are sum up here.

Reputation: a rate which is voted on by other players who decide to either prefer or avoid another player. The reputation defaults to 100% (Five stars) over time after one player has preferred a user.

Friends list: a list of the gamers added by the user. Through it you can exchange messages, start audio or video chats, invite to games, compare the achievements of games, etc.

Recent player list: is a list of the last 50 players that the user has played with.

Complaint filing system: This allows a user to report another user that has broken the Xbox Live Terms of Use.

Feature	Live Silver	Live Gold
Pricing	Free	Subscription fee
Voice chat	Yes	Yes
Party chat	No	Yes
Video chat	No	Yes
Avatars	Yes	Yes
Downloadable content	Yes	Yes
Multiplayer gaming	No	Yes
Parties	No	Yes
Netflix movie streaming	No	Yes
Sky Player	No	Yes
Xbox Live Arcade point results	Yes	Yes
Facebook	No	Yes
Twitter	No	Yes
last.FM	No	Yes
Zune	No	Yes
1 vs 100	No	Yes

Table 1 - Comparison of Xbox Live Silver and Xbox Live Gold

Windows Live Messenger integration: well know instant messenger from Microsoft.

Access to the Xbox Live Marketplace: virtual store that offers games, music and movie content.

Voice Chat and *Video Chat:* user must have headset and Lite Vision Camera (for Video Chat).

Multiplayer Gameplay: Main focus of our work here, the Multiplayer is only provided on the Gold Live, as show in the Table 1.

Enhanced matchmaking: another one of the main subjects of our study, the matchmaking is responsible for find the best player available to play together into an online party. It takes user of the cumulative gamerscore, reputation, location/language profile, and the gamer zone.

Parental controls: very important option that create limitation on the children's exposure to other users (through the menu "Family Settings").

Inside Xbox: is a newsletter about Xbox Live news, events, products, interviews and games that is integrated in the Xbox 360 Dashboard. Content for Inside Xbox is created by the Xbox global marketing team and features streaming video segments "Pick Up and Play", "Major’s Minute" with Larry Hryb and "Game with Fame".

Last.fm: is a well-know website that also provides an application that allows users to stream their favorite music and search for related artists.

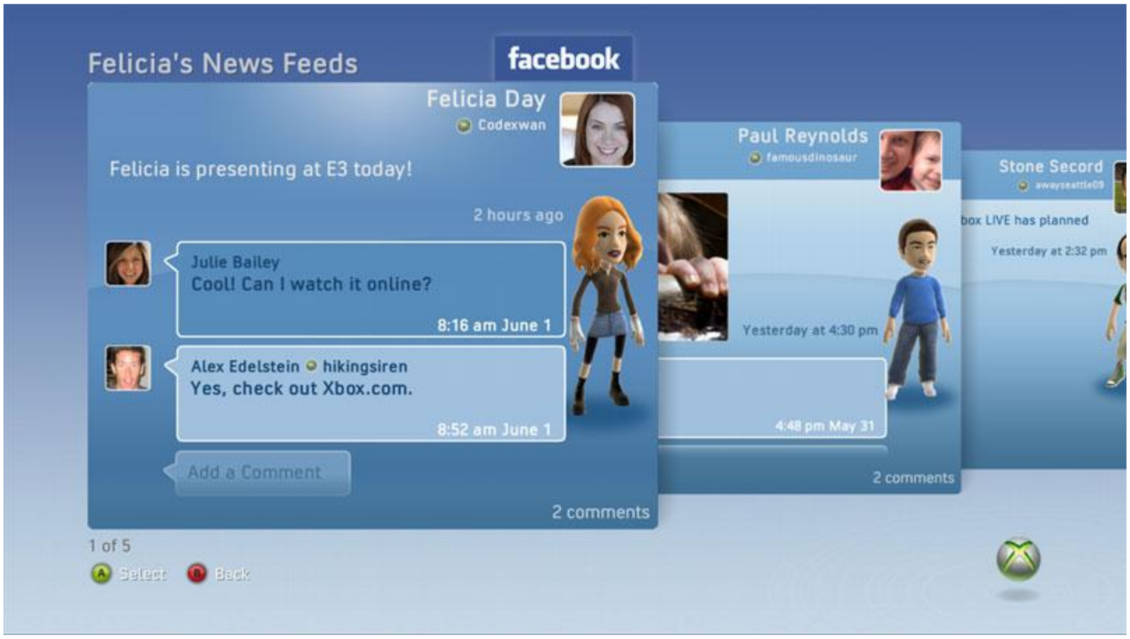
Social networking sites: famous social networks are also available over the Xbox Live as Facebook and Twitter. Users between the ages of 13-17 can access the application under a parent's consent.

Zune: application that allows users to stream movies and TV shows from the Video Marketplace instantly in 1080p HD with 5.1 surround sound.

Movie Parties: is an application that allows movies and TV shows to be viewable with friends over Xbox Live.

Sky Player: is available in the UK and Ireland, offering live and on-demand TV including Sky News, Sky Sports and the Disney Channel.

Netflix: application that allows members with unlimited rentals to stream any movie or TV show from their Netflix Instant Queue. This feature is only available in the US.



Felicia's News Feeds

facebook

Felicia Day
Codexwan

Felicia is presenting at E3 today!

2 hours ago

Julie Bailey
Cool! Can I watch it online?
8:16 am June 1

Alex Edelstein hikingsiren
Yes, check out Xbox.com.
8:52 am June 1

Add a Comment

2 comments

Paul Reynolds
famousdinosaur

Yesterday at 4:30 pm

4:48 pm May 31

2 comments

Stone Secord
awayseattle09

box LIVE has planned

Yesterday at 2:32 pm

1 of 5

Figure 7 - Facebook application running on XBox Live

Game Room: is a virtual space, offering an initial library of 30 original arcade and console classics including Centipede, Asteroids Deluxe, and Super Cobra, displayed in their original cabinet designs. Purchased arcade titles are playable via Xbox Live and Games for Windows Live. Game Room is an Xbox 360 and Windows application with games sold separately for 240 Microsoft Points. Users can also buy a 'credit' to play one session of a game for 40 Microsoft Points.

2.1.3 TrueSkill Ranking

In fact the TrueSkill Ranking was a creation of the Microsoft Research Laboratory (in Cambridge) and is consist in a Bayesian raking algorithm used to find the finest level of a player. This system is very similar to the Elo Rating system, which rates the skill of Chess and Go players.

The introduction of the TrueSkill release paper states: “Skill rating in competitive games and sports serve three main functions. First, they allow players to be matched with other players of similar skill leading to interesting, balanced matches. Second, the ratings can be made available to the players and the interested public and thus simulate an interest and competition. Thirdly, ratings can be used as criteria of qualifications for tournaments.”

It is impossible to measure the performance of a player, but given the facts as quantity of games and the win rate, some estimation can be done. Since the ranking of a player will be updated each time he plays a game, the rank acquires more confidence about the player’s skill.

Into the TrueSkill ranking, the player’s ranking is dated as a Normal distribution (Ɲ), which is characterized by a mean (µ-mu) and a variance (σ - sigma, which is the actual reliability of the mu value). Such Ɲ(x), then, can be observed as being the probability of a player has the “correct” ranking x.

$$p(\mathbf{s}) := \prod_{i=1}^{n} \mathcal{N}(s_i; \mu_i, \sigma_i^2)$$

Equation 1- Performance of a player

The follow graph shows how the iterative process of the TrueSkill works. There are four types of variables: s_i for the *skill* of all players, p_i for the performances of all players, t_i for the performance of all teams and d_j for the team performance differences. The first row of factors encode the prior, the product of the remaining factors characterizes the likelihood for the game outcome Team 1 > Team 2 = Team 3. The arrows indicate the optimal message passing schedule. First, all light arrow messages are updated from the top to bottom. In the following, the schedule over the team performance (difference) nodes are iterated in the order of the numbers. Finally, the posterior over the skill is computed by updating all the dark arrows messages from the bottom to top.

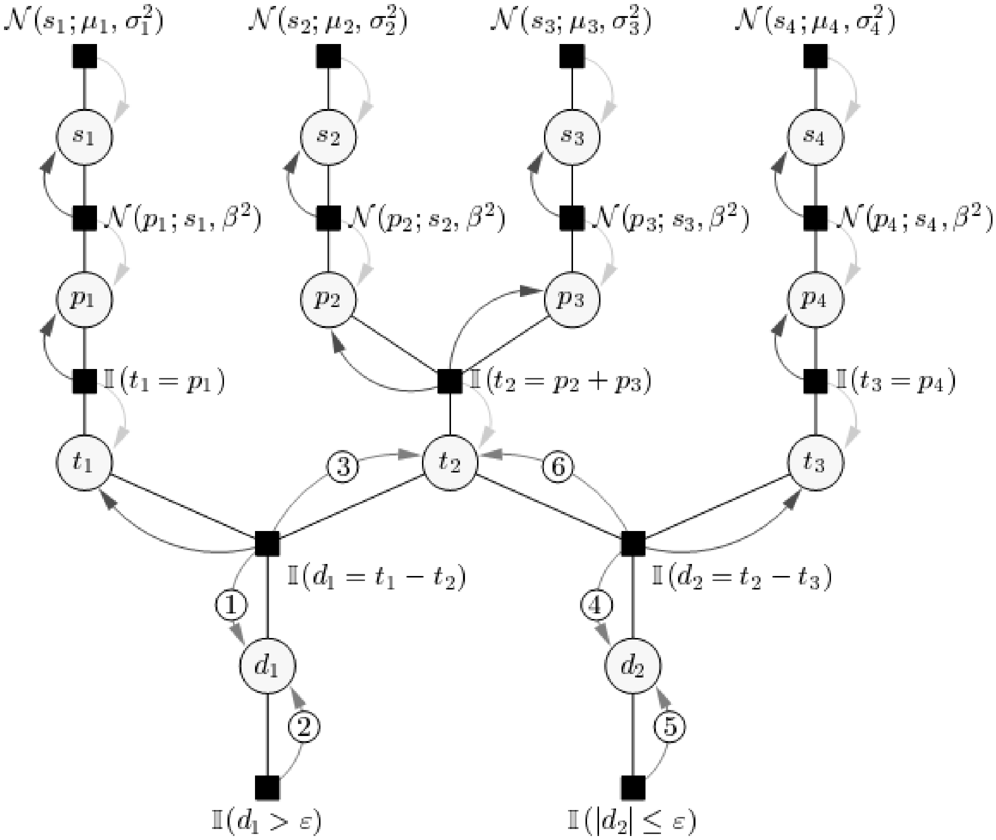

$\mathcal{N}(s_1;\mu_1,\sigma_1^2)$
$\mathcal{N}(s_2;\mu_2,\sigma_2^2)$
$\mathcal{N}(s_3;\mu_3,\sigma_3^2)$
$\mathcal{N}(s_4;\mu_4,\sigma_4^2)$
s_1
s_2
s_3
s_4
$\mathcal{N}(p_1;s_1,\beta^2)$
$\mathcal{N}(p_2;s_2,\beta^2)$
$\mathcal{N}(p_3;s_3,\beta^2)$
$\mathcal{N}(p_4;s_4,\beta^2)$
p_1
p_2
p_3
p_4
$\mathbb{I}(t_1=p_1)$
$\mathbb{I}(t_2=p_2+p_3)$
$\mathbb{I}(t_3=p_4)$
t_1
3
t_2
6
t_3
$\mathbb{I}(d_1=t_1-t_2)$
$\mathbb{I}(d_2=t_2-t_3)$
1
4
d_1
d_2
2
5
$\mathbb{I}(d_1>\varepsilon)$
$\mathbb{I}(|d_2|\leq\varepsilon)$

Figure 8 - An Example of a TrueSkill factor graph

At the Xbox Live, the starting values are μ = 25 and σ = 25/3, what correspond to a probability of a positive skill of 99%. Through a fast analyze of the data the researchers of Microsoft could arrive in three important points which are the follow:

- Games differ in the number of effective skill levels. Games of chance (e.g. single game Backgammon or Uno) have a narrow skill distribution, in the other hand more realistic games (such as: Racing simulators or shooters) have a wide skill distribution.
- The skill display results in a feedback loop back to the players, who often check their status and consequently view that as a reward or punishment. In order to improve the ranking players start to choose more carefully his opponents or even cheat.
- The total skill distribution is shifted to below the prior distribution when new players lose their first few games (what is expected).

2.1.4 TrueSkill Matchmaking

All the complexity of the TrueSkill is useless if it is not used. The purpose of it in our scope is to make possible to find the best solution in matchmaking, for that it is necessary to emphasize what are the main characteristics of an online player and how to achieve their satisfaction.

Players can be divided in two main categories:

- *Play for Fun*: Are players that just want to meet new people or either play with friends for some time. They are mainly interested in have a simple and enjoyable game, not carrying for rewards or competition.
- *Play for keeps*: As referred as the hardcore gamers, they want to see how they play, their scores, get all rewards and bonus of levels. In fact they want to compete to see who is better.

In order to satisfy both of this kind of gamers the Xbox live has two kind of online matches (respectively):

- *Player Match*: It is designed for recreation, it does not affect the player’s ranking, instead it just record the last players and the most common players that player have been gaming with and put them as priority.
- *Ranked Match*: This is the one who take profit of the TrueSkill Ranking, trying to get the skill of that player and put him in front of players at the same level (of skill).

There was a time when the mainly multiplayer experiences were designed to Local Area Networks (LAN), but now the main focus is in the online experience what means that several factors must be watched as the ones follow in the Table 2 and also in the Figure 9.

	LAN	Internet
Number of simultaneous players	8 – 32	1,000 – 1,000,000
Informal agreement reachable	Yes	No
Adversarial (cheating)	No	Yes
Anonymous	No	Yes
Session browser adequate	Yes	No
Latency	Known	Unknown

Table 2 - LAN versus Internet connections

One of the most important of these factors is certainly the number of simultaneous players, in order to avoid this of being a big trouble some recommendations can be followed, trying to reach the best matchmaking possible.

- *Use voting mechanism to get match parameters* (map, type, car set, etc.); this will minimize the results and make the search for the best solution faster.
- *Avoid Host/Join session* for the player interface; the gamer just want to start a game, the minor details of configuration can be deal by the networking system.
- *No player should have more power* in the selection of match parameters or to boot other players.
- *Show* information about *the progress of the matchmaking* process; showing how many players are needed or the current level difference helps the player judge how much they have to wait

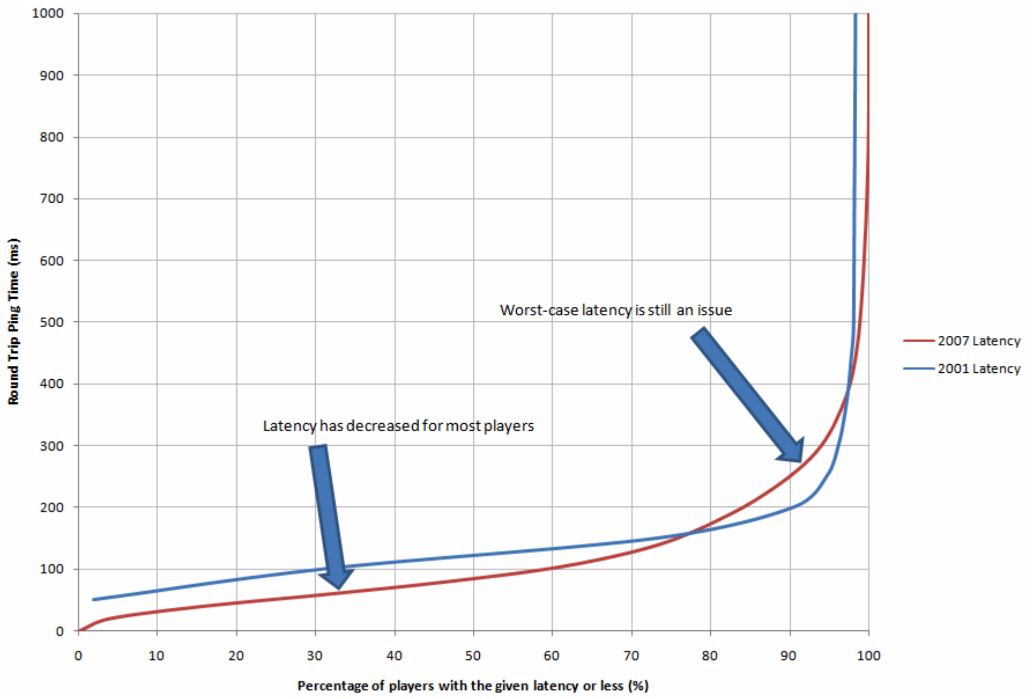


Figure 9 - Round Trip Latency of Console to Xbox LIVE

For ranked games some other characteristics must be emphasized,

- *Focus on the balanced gameplay*; always ensure that the match parameters are such that both of the teams have an equal chance to win.

- Do not give information, to a player, about the opposition or match parameters before the match starts; since there is no penalty when players quit the game before the start of a game, the TrueSkill matchmaking can be compromised.
- Discourage players from leaving the matching after the game starts; informing them that this will cause a drop on their skill level.

Nevertheless there is always the relation between waiting time and the quality of the match, in other words, the match must be well set at the same time that the player do not want to wait too much to start the game. In the TrueSkill matchmaking the waiting time can be calculate by;

$$WaitingTime = \frac{GamesMode * MatchTime * PlayersPerMatch * SkillBins}{PlayersOnline}$$

Where:

- *GameModes*; is the total number of game modes in the game. Game modes define the different ways that the game can be played.
- *MatchTime;* is the average duration of a match (in minutes).
- *PlayersPerMatch;* is the average number of players per match.
- *SkillBins;* is the number of skill bins (the allowable variance in skill).
- *PlayersOnLine;* is the average number of players online at any given time.
- *WaitingTime;* is the waiting time for a match (in minutes).

Therefore, for internet multiplayer games the developer must narrow possibilities to speed up the matchmaking process. Some of the main ideas that can be consider as cleaver ways to decrease the time waiting are the follow.

- *Keep the number of game modes small;* The larger the number of game modes, the more the player base is divided into smaller and smaller sets. Game modes should be used only to define styles of play that require distinctly different skill sets to be successful. On Xbox Live, game modes are used in conjunction with game types to create separated TrueSkill leaderboard for each game type/game mode pair.

- *The preceding equation calculates the average waiting time*. For highly populated skill bins, the waiting time can be smaller even if the number of skill bins is large (that is, the level gap for matchmaking is small). If you choose to add filters for skill, you should have an adaptive level gap threshold that increases over (waiting) time.
- *Design for relatively short matches to minimize waiting time in the matchmaking lobby*. A match of two hours may sound like fun, but it is not only a large time commitment on the player's part, but it also significantly increases the waiting time of players searching for a match.

2.1.5 Xbox Live Market Place

The *Wikipedia* defines: “The Xbox Live Marketplace (XBLM) is a virtual market designed for Microsoft's Xbox 360 console that allows Xbox Live members to download purchased or promotional content. The service offers movie and game trailers, Video Store, game demos, Xbox Live Arcade games, Xbox Live Indie Games (Previously Community Games), Games on Demand (Xbox 360 and Xbox Originals), downloadable content such as map packs, gamer pictures, and Xbox 360 Dashboard themes.”

2.2 PlayStation Network (PSN)

Figure 10 - PlayStation Network Logo

2.2.1 Introduction

On May 15, 2006, was the time of Sony. The big Japanese company that had the major console sold in the last generation (PlayStation 2) started to follow Microsoft and announced it own unified online game service. First called PlayStation Network Platform, it was claimed that differently from Xbox Live it would be a no costs service and focus on the main two consoles: PlayStation 3 (PS3) and PlayStation Portable (PSP).

At the beginning of the service was only possible to register to it through a PlayStation 3 or a PSP system interface, nowadays the subscription is also possible to be done accessing the website of the PlayStation Network (IV). In September 21, 2006, Sony revealed at the Tokyo Game Show that the system was going to make possible the acquisition of titles as download and they would provide first the games with small sets of data.

The registration of an account in the PlayStation Network is not linked to a console serial number; nevertheless in this situation the player cannot buy titles, making in this way the registration of a console necessary to buy games. As the accounts are not directly linked to the serial number of each machine, multiple consoles can be using the same account at a maximum of five, but the user must not be deleted, or the content data will be blocked.

Figure 11 - PlayStation 3 Console

Figure 12 - PlayStation Portable (PSP) Handheld

2.2.2 Features

As well as the Xbox Live, the PSN has a large number of features. Some of them will be cited and specified in the follow:

Profiles: a concept that is not far from the one of a Gamercard (see Xbox Live Features in item 2.1.2). It has the gamer nickname, favorite game, citation, and Trophies (similar to Xbox Live Achievements).

Sign-in ID/ Online-ID: basically it is the email used into the registration process. It is through this ID that the users makes the login and also receive information about news from Sony.

Friend List: as well as in the Xbox service, it provides a list up to a hundred players which must be added through the system interface as friends.

Instant Messaging: it integrates into the XMB (Xross Media Bar) a simple option to send messages to the gamers friends.

Lobbies/Matchmaking: Provides an internal solution to create Worlds, Lobbies and Rooms in order to provide a matchmaking system. This item will be more detailed in further sections.

Multiplayer gameplay: The online gameplay of the PSN is hold with up to seven players local and up to 64 into the same game session. As well the matchmaking this will be subject of further explanations.

Scores/Ranking: Differently from the Xbox TrueSkill Ranking (section 2.1.3) the ranking over the PSN is calculated over the sum of his Trophies and giving a level for a certain amount of points got from the trophies, the tables below detail these points.

Level	Points Required
Level 1	0
Level 2	210
Level 3	600
Level 4	1,200
Level 5	2,400
Level 6	4,000
Level 7	6,000
Level 8	8,000
Level 9	10,000
Level 10	12,000
Level 11	14,000
Level 12	16,000
Level 13	24,000
Level 14	32,000
Level 15	40,000
Level 16	48,000
Level 17	56,000
Level 18	64,000

Table 3- Number of points to achieve a Level

Trophy	Point Value
Bronze	15
Silver	30
Gold	90
Platinum	180

Table 4 - Number of Points that are added for each Trophy

Trophies: as already cited they are rewards giving to the gamer when it completes certain goals inside the games. The Trophies can be: Platinum, Gold, Silver or Bronze as described in the image below.

Avatars: Virtual image of the gamer that is his representation on the virtual world of the console. These avatars can be used in the social-game network PlayStation Home (reference).

Voice/Video Chat: Users that have a camera and a head-set can communicate between themselves through a real-time chat. This option, however, is not available in-game.

Cross Game Chat Room: Text messages that can be exchange on the flow at same time that the gamer is playing.

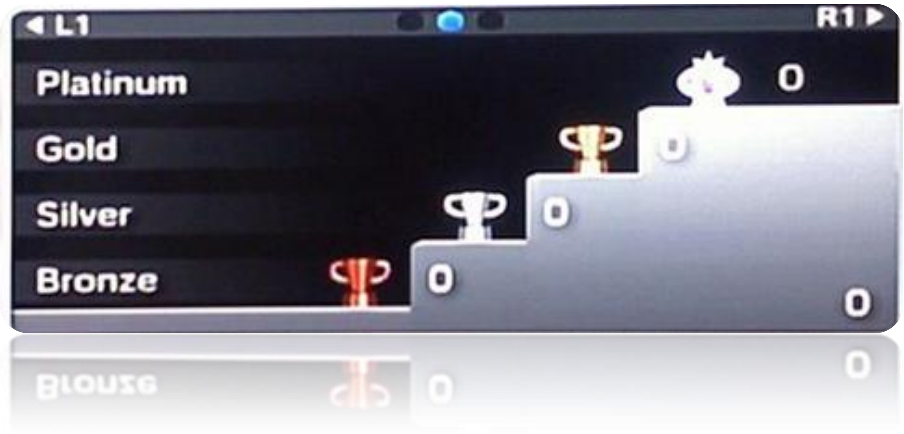


Figure 13 - Trophies of the PSN

Ad-hoc Party: An exclusive option of PSP, it makes possible to a user connect to another user without the use of an Access Point (AP).

Internet Browser: Powered with the Google Search Engine, a simple way to access normal Internet pages is available into this feature.

PlayStation Home: It is a community-based social game networking, similar to the famous Second Life [V], it creates the experience of a virtual world where users have they homes, cloths, items, etc.

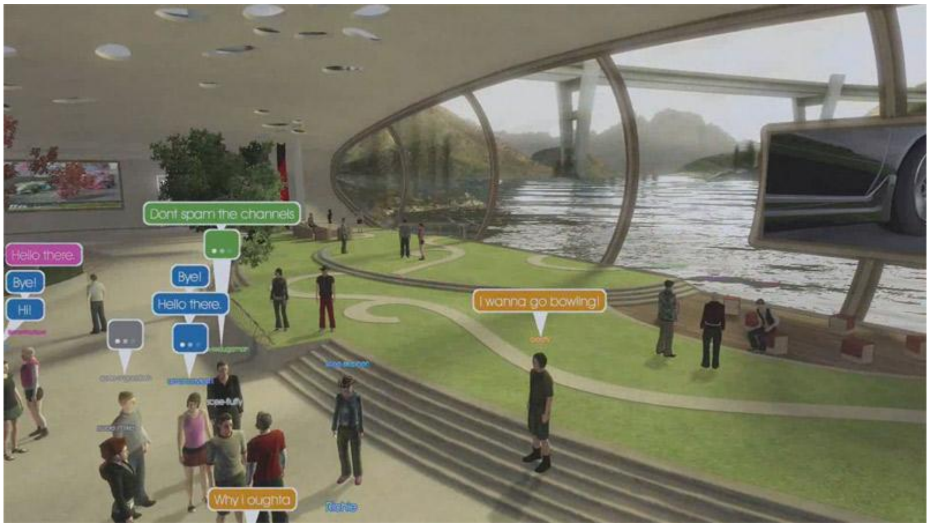


Figure 14 - PlayStation Home Central's Plaza

Parental Control: limiting the access of BD, DVD and the Internet Browser for children's control.

System Update: Both PS3 and PSP can do firmware updates directly from the PSN.

What's New: A RSS like blog which update the user to the very last news about the consoles and games.

Entertainment Third-Party Applications: PlayStation Network has many contracts with different service provides and they change from region to region. The services online until now are: FirstPlay, Qore, VidZone, ABC iView, BBC iPlayer, MLB.tv, Netflix, now TV, RTE Player, TVNZ ondemand.

PlayStation Store: Similar to the Xbox Live Market Place (see section 2.1.5) it is a content download provider. User can download data from videos and games, including some demonstrations and trailers.

2.2.3 PSN System

As well as the Xbox has his TrueSkill Matchmaking the PlayStation Network also has a matching system. But differently from the Xbox Live, the Matching system of PSN is focused on the organization of sessions to group players together, focusing into a more group party games than a player-to-player confrontation. That means to play in groups of more than two players.

One thing that is clear from this moment is that we will not have ranking systems as accurate as the TrueSkill, but instead of it the players must choose against who they will play. The method is based on the most common old-fashioned way; there is a player who hosts a game and others that join it.

Unfortunately there is no information more that can be shared, since they are part of confidential documents of SCE, holding the explanation of this subject at this point. Further details must to be claimed directly to the official developers or to Sony itself.

2.2.4 PlayStation Store

The *Wikipedia* defines: “The PlayStation Store is an online virtual market available to users of Sony's PlayStation 3 (PS3) and PlayStation Portable (PSP) game consoles via the PlayStation Network. The Store offers a range of downloadable content both for purchase and available free of charge. Available content includes full games, add-on content, playable demos, themes and game and movie trailers. The service is accessible through an icon on the XMB on the PS3 and PSP. The PS3 store can also be accessed on the PSP via a Remote Play connection to the PS3. The PSP store is also available via the PC application, Media Go. As of September 24, 2009, there have been over 600 million downloads from the PlayStation Store worldwide."

2.3 Nintendo Wi-Fi Connection

Figure 15 - Nintendo Wi-Fi Connection

2.3.1 Introduction

Even with the out-of-the-box strategy that Nintendo has adopted in Wii, their last console, they also were very concerned about the connective systems to improve the gamer's experience. Before talk about the online service, it must be said that Wii is a console, as well as DS, that was build over innovative aspects and focusing on the market share of non-players instead of regular gamers.

The service had his launch on 14th, November, 2005 in the United States and 21th, November in Europe being released together with Mario Kart DS and Tony Hawk's American Sk8land, two titles for the Nintendo DS. The NWC (Nintendo Wi-Fi Connection) became popular and on 30th May of 2007 it had already 5 million members and had around 200 million accesses.

The game who started to use the NWC on Wii was Pokémon Battle Revolution, and they used exactly the same technology (as in the DS), and the games that have the service available has the logo of the Figure 15 in their cover. Instead of a wired Ethernet connection (as Xbox and PS3, see section 2.1.1 and 2.2.1) both Wii and DS have a wireless 802.11 connection (Wi-Fi) and they connect to public HotSpots or to Access Points (APs). Nevertheless the build-in device of DS does not support the security protocol WPA, but only WEP.

Differently from the other online gaming services, the NWC does not include game digital stores, instead of Nintendo has the Wii Shop Channel or the DS Shop. In this way the digital distribution of each console is managed by different applications.

Figure 16 - Nintendo Wii Console

Figure 17 - Nintendo DS Handheld

2.3.2 Features

Multiplayer match: Up to sixteen players on Nintendo DS and up to thirty-two players on Wii can play online multiplayer for free.

Worldwide Matchmaking: The NWC creates an infra-structure to make easy to players match with different options.

Leaderboards: Each game can have it own leaderboard to show the player which has acquired more points into it.

Tournaments: It is not hard to organize tournaments inside the games to compete with friends and have a match into a knock-out mode.

Friend Codes: This is one of the exclusive features of NWC. Each game (on both Wii and DS) has a unique identification number (be aware that is for each game, not Title), as well as each player has its own unique ID. In this way these two are combined in order to create a Friend Code. This code provides the possibility to add another player as a Friend, if both are in concordance, creating in this way another set of features that follows.

Friend List: A list where the player can see if his friends are online and interact with them.

Cooperative Play: Player that have added each other with the Friends Codes can play together to defeat the challenges of a game.

Text Chats: An instant messenger that provides the exchange of messages between friends.

Voice Chats: If the user has the necessary devices (i.e. headset) he can talk with his friends through the Friend Codes.

Rivals: It is a feature similar to Friends Codes, but instead of make possible other features it mainly memorizes the rival player (after a match) to play again against him.

Pay & Play: Through this option is possible to access DLC (Downloadable content) paying the content with Nintendo Points. It was launched in 2008.

WiiConnect24: is the possibility to remain connected to the Internet even when the console is in stand-by mode.

2.3.3 Matchmaking

The matchmaking of the Nintendo Wi-Fi Connection has two main characteristics, which are: it can be hold between friends (see Friend Code in the section 2.3.2) or it can be hold with random players.

The second option however has a skill level system that tries to match together players with the same experience and abilities in the game; this is done seamlessly via Nintendo's servers through the use of a skill mapping system creating the easiest environment open to any age/skill level users.

Unfortunately further specifications about the behavior of the system are confidential, that is why they cannot be specified here. Only official developers of Nintendo have access to the official manuals and documentation that can provide this information.

3 Methodology

This section is divided in main three parts: Network Structures, Connection Socket Samples and Organization of Network Sessions. The steps and studies here presented were part of a continuous learning process to understand and research the actual network systems and the application of them over the common Network Systems described in the section 2 – State of the Art.

3.1 Network Structures

In order to design a network application first we must understand how different can be the network processing depending on each kind of application that is being developed. The Figure below shows difference between diverse virtual environments (VE) and the relationship of them with different game's genre.

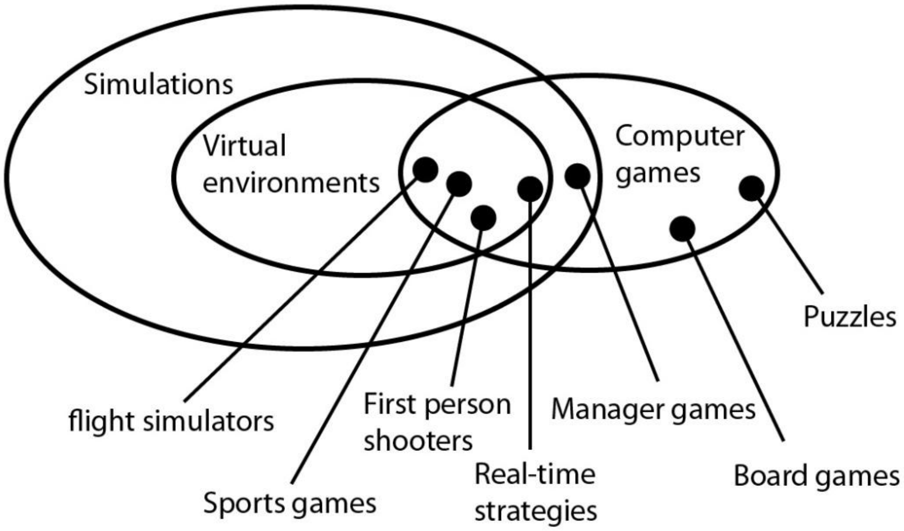


Figure 18 - Relationship of simulations, VE and computer games with game's genre.

The meaning of this clustering process is that as more a game is to the left (into the simulation- virtual environments field) more important is to have a response time very low, while puzzles and board games, which are more to the right portion of the graph, can have some delays (also known as *lag* in networking games).

The lack of a good response time in a game influences a lot the player performance, in the Figure 19 we have an analysis of the curves of performance directly linked to the latency of a network package. In order to improve the latency and reduce the bandwidth requirements there are outnumbered techniques (for more details, see [VI]).

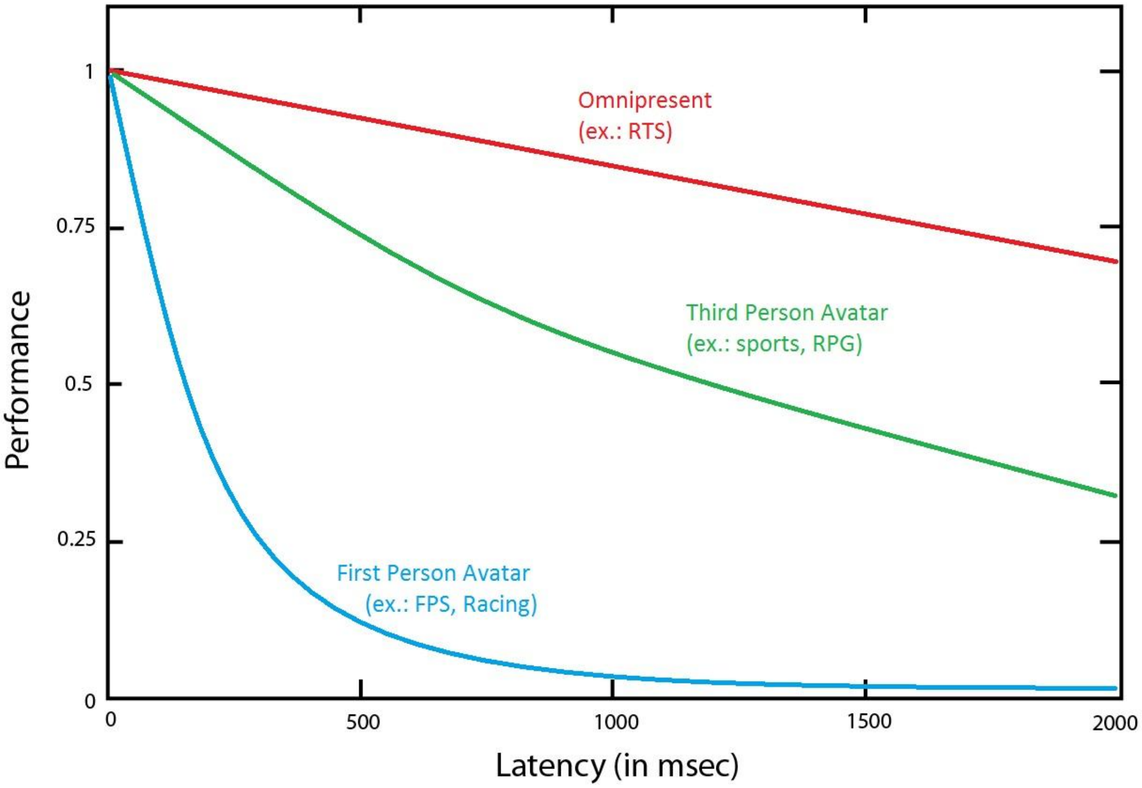


Figure 19 - User Performance under Different Induced Latencies for Several Classes of Games

Nevertheless not only the exchange of information is important, but also where it remains. There are several architectures to handle the messages; as well there are others to handle the storage of the data. The importance of these architectures changes mainly depending on the importance of each player in a game, and the different hardware/connection configurations that can be found.

For instance, the most common architecture is the client-server architecture. In this architecture there is a player, as referred as the Host, which opens a game and wait for other players to join to that game. The other players that join that section will be manipulated and have their data handled by the Host.

In this case, when each player joins the game the host will need to do a *handshake operation.* This process will exchange prior data to the new player, informing him about the current situation of the game and giving needed information for the player to visualize the game as him. At that time the host also will need to refresh the information of all the other clients, because of the new player.

Not only at the joining a game, but at any action that any player takes on the match will need to be sent to the server (host) and he will need to refresh the information to all the others players. Following this way it is clear that the number of data that must be send is always bigger than it could be, but the process is always synchronized and controlled by the server.

Also if that player who is considered the server has a problem and consequently drops the connection the match will surely stops since the others players will not receive any kind of answer from each other.

Furthermore there are other architectures, as presented in the Figure 20, and each one of these has their good and bad points. The best option must be always been taken looking careful for the kind of game (game's genre) that you are developing the network function.

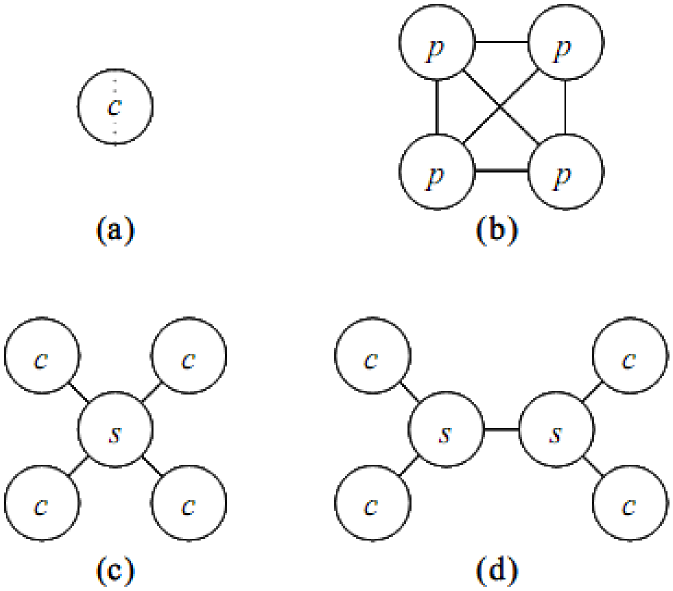


Figure 20 - Degrees of deployment: (a) a split-screen on a single computer, (b) a peer-to-peer architecture, (c) a client/server architecture, and (d) a server-network architecture.

On the other hand there is the peer-to-peer architecture where every player changes information directly to the other players having the payout that the bandwidth must be really large, since each player will send N-1 times the data (being N the number of players). Therefore none of the players is essential to continue a match, putting aside complications in the case of a player's drop connection.

For instance, when the game is of the kind first person shooter (FPS) it is a must that all the information be concise and the response time the lower possible. In that case usually it is used an client-server platform, since the player's experience in game would be too low if the peer-to-peer configuration had been taken, in view of the fact that the data amount in the network would be bigger and for some player the information would arrive at different times, not carrying about packages that could be lost.

Is not hard to notice that peer-to-peer connections would be better for a minimum number of players, and a server to bigger numbers, however it must be considered that in client-server architectures the server will have more work than the clients, as well as it must have a larger bandwidth to handle with all the data exchanges.

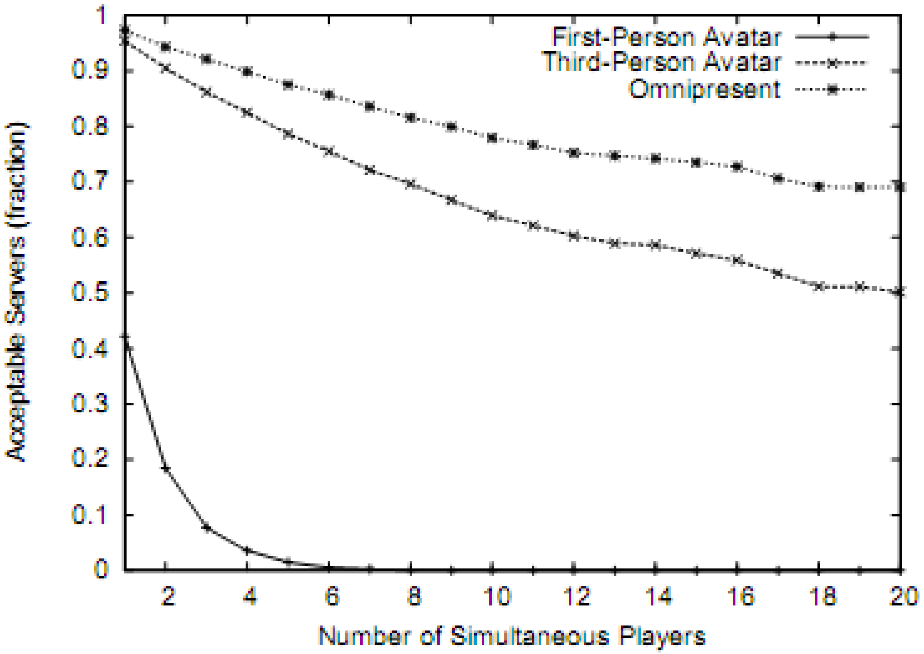


Figure 21 - Fraction of Acceptable Server versus Number of Clients Playing Together for Different Classes of Games

However the server-network architecture can be hold to multiple kinds of games, since a single server cannot hold the data processing and the response time is not one of the priorities of the game in question. The Figure 21 shows a graph from [VII] that evaluates the capacity of increase the number of games on the server while maintain a good gaming experience.

Not far from these architectures also there is the data architecture of the match. In the Figure 22 is possible to observe three kinds of architecture: Centralized, Distributed and Replicated. In fact this is apart the organization of the network because they can be mixed in several ways.

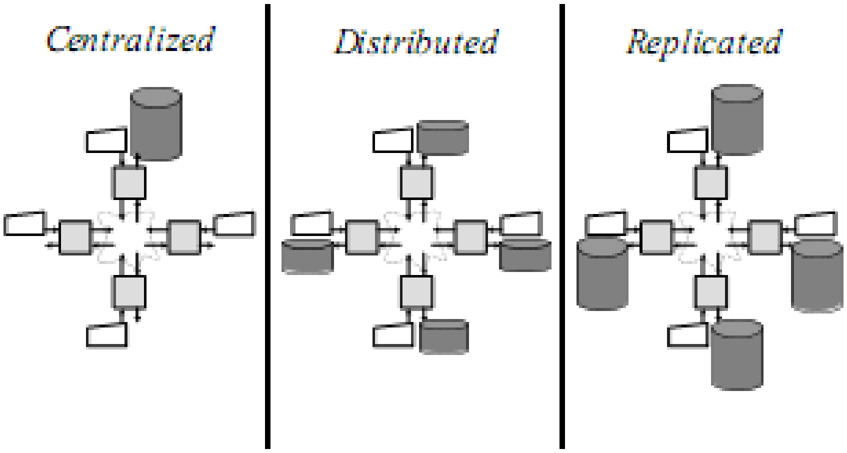


Figure 22 - In centralized data architectures, on (data server) the node stores all data. In replicated architecture, each node manages a replica of all data. In distributed architecture, the data is distributed among the nodes.

Is not uncommon that the design mixes the client-server architecture with the replicated, in this way the system can maintain a player's drop verification and noticing if the host of the match just loses his connection. Then the system starts a process to change the host of the match to another player, since they all share the same information.

However, the developer must be aware that the data consistency will be, and must be, controlled through software methods, and that process will have a great impact in the network bandwidth adding many data to be exchanged every time. In fact these two attributes, consistency and responsiveness, defines the models for data and control architectures.

In order to achieve high consistency the architecture must guarantee that all the process must be running tightly coupled. Usually this would require a high bandwidth, low latency and a small number of remote nodes. On the other hand, to achieve high responsiveness (or timeliness [VI]), the data must be processed as fast as possible, which leads to loosely coupled data. Not only, in order to get high responsiveness are applied several kinds of data compression and algorithms (to reduce the data sending), leading to a higher computational weight.

The Figure 23 states the problem. The game in execution is in a local node, it sends a control message through the relay and get data messages from it. In turn, the relay exchanges data with other nodes via a network (in specific the Internet). Here, a relay is a logical concept which expresses how the control would affect the data.

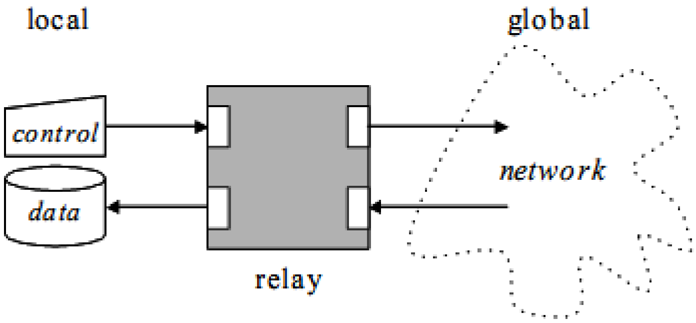


Figure 23 - Architectures defines how messages are relayed between local and remote nodes.

Nevertheless, the amount of the bandwidth is also measured depending on the transmission technique as you see in the Figure 24. Early in the times of LAN implementations it was common to use the Broadcast technique (b), sending messages to all users on the network [VIII]. It is clear that it leads to major problems when the number of participants increases. However the messages commonly have more than a unique receiver, so being a waste to send Unicast (a) to multiple receivers.

Hence in the 1990's the Multicast architecture(c) [IX], which is the communication between Broadcast and Unicast, appeared; allowing to user to join groups that are from his interest. The user sends a message to a group as in Unicast and the group receives the message as in Broadcast; the multicast is strongly applied into the Games Network Systems.

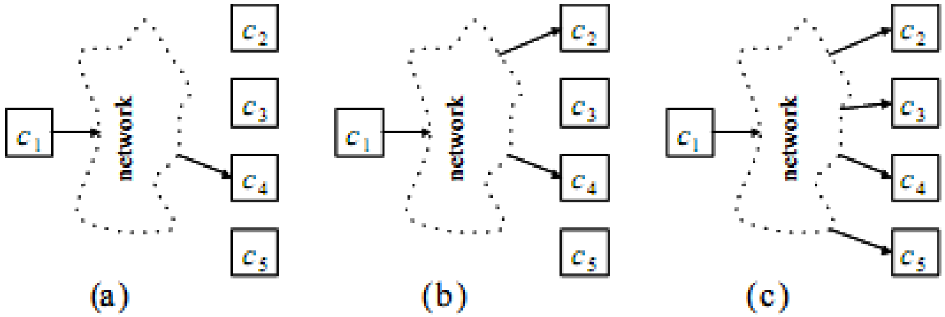


Figure 24 - Transmission techniques

Plus, a very common problem that is found when handling with Internet connection is to address the specific player in order to deliver the message. The problem is given since the fact that many Internet users uses Routers or proxy servers to split they connection, method that has been even more applied in order to connect Wi-Fi devices. The Figure 25 shows us an example of different game players in a networking using (or not) Proxy servers.

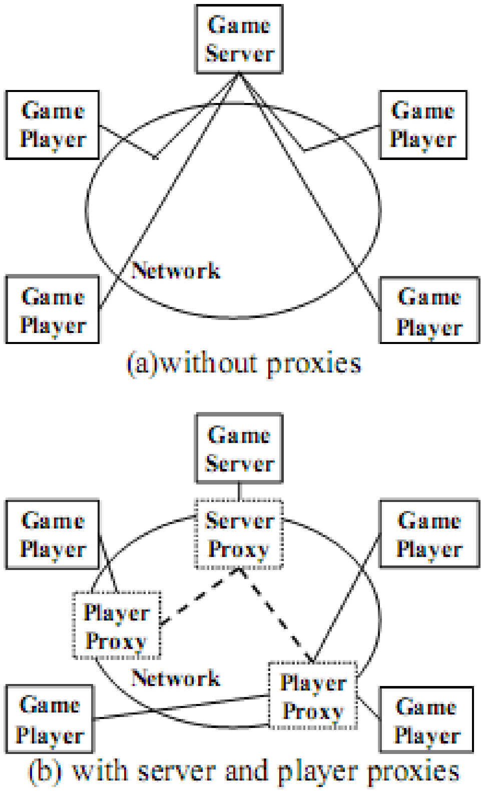


Figure 25 - Distributed game environment

The method that rules the translation of the Private Network Address to the IP address (internet) is the NAT. Is defined in [X]: *Network Address Translation is a method by which IP addresses are mapped from one realm to another, in an attempt to provide transparent routing to hosts. Traditionally, NAT devices are used to connect an isolated address realm with private unregistered addresses to an external realm with globally unique registered addresses.*

Network applications need the external IP to communication between themselves, for this purpose it was created the STUN (Simple traversal UDP over NATs) which classifies the NAT implementation as: Full Cone (Fig. 26), Restricted Cone (Fig. 27), Port Restricted Cone (Fig. 28), and Symmetric(Fig. 29).

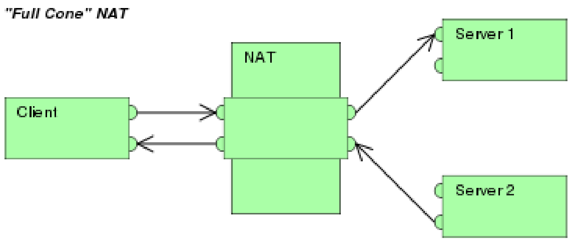


Figure 26 - Full Cone NAT

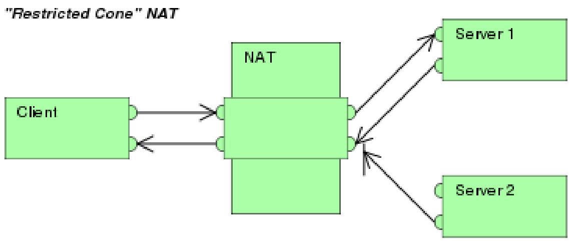


Figure 27 - Restricted Cone NAT

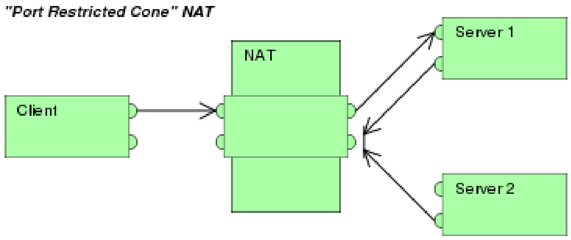


Figure 28 - Port Restricted Cone NAT

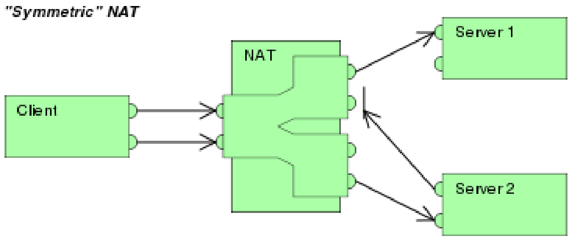


Figure 29 - Symmetric NAT

However, these procedures have since been deprecated from standard status, since many of the methods has proven to be inefficient and inadequate to assess many devices. A new standard has been formalized in the RFC 5389 in October of 2008 and now the acronyms STUN means: Session Traversal Utilities for NAT.

Luckily the network systems in question of this thesis are able to work around the most common NAT uses and provide a communication as a unicast way to consoles. Nevertheless it must be said that a Port Forwarding configurations maybe need to be done.

3.2 Connection Socket Samples

In this section is explained the basics of establishing a connection between two devices using a Socket. As defined in [XI]: “A socket pair for a TCP connection is the four-tuple that defines the two endpoints of the connections: the local IP address, local port, foreign IP address, and foreign port. A socket pair uniquely identifies every TCP connection on a network.”

Nevertheless all the concept about sockets in TCP can be extend to UDP, stated also in [IX], and this is very common since for games one of the main points is to reduce the data package and minimize the use of the bandwidth, besides the loosing of a package would not be so problematic since there is a continuous flow of data, overwriting the old packages with new in a small time gap. However an ACK message must be used to get essential messages.

The sample here presented is showing the basic use of sockets to build a simple Client-Server application that will be needed to understand the low level communication that happen in the Network Systems. Further details are described in the Section 4 - Results.

First, we must understand how the socket server should work. For that we will be using some socket structures that are defined in a C/Unix environment, depending on the platform some changes should occur, nevertheless the concept does not change. On the server side a connection must be open and it must be waiting a connection from a client, for this purpose a port is specified, the Figure beneath shows us an example.

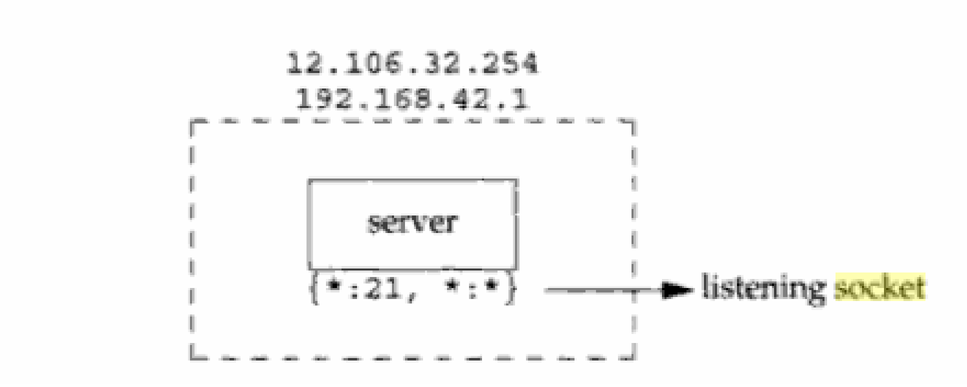


Figure 30 - Socket server with a passive open on port 21.

In order to accomplish the listening process a socket must be created and after all the parameters have been set the method *bind* must be called, this method in fact binds a socket to a port as we can see in the code below[XII].

```
    hints.ai_family = AF_UNSPEC; // set to AF_INET to force IPv4
    hints.ai_socktype = SOCK_DGRAM;
    hints.ai_flags = AI_PASSIVE; // use my IP

    if ((rv = getaddrinfo(NULL, MYPORT, &hints, &servinfo)) != 0) {
        fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
        return 1;
    }


    // loop through all the results and bind to the first we can
    for(p = servinfo; p != NULL; p = p->ai_next) {
        if ((sockfd = socket(p->ai_family, p->ai_socktype,
                p->ai_protocol)) == -1) {
            perror("listener: socket");
            continue;
        }

        if (bind(sockfd, p->ai_addr, p->ai_addrlen) == -1) {
            close(sockfd);
            perror("listener: bind");
            continue;
        }

        break;
    }
    if ((numbytes = recvfrom(sockfd, buf, MAXBUFLEN-1 , 0,
        (struct sockaddr *)&their_addr, &addr_len)) == -1) {
        perror("recvfrom");
        exit(1);
    }
```

In this case we have created the socket in the first highlighted function, and after that we have bind the socket to an address that is using MYPORT defined as the port to listen the connection. The next step is to see how the client will join to this server, having the parameters of IP and port well defined, the third highlight shows that the server will be waiting for a connection on the specific socket created.

In the Figure 31 we have an example of what happens when the client request a connection to the server. For that the client must know a priori the server address and the port that have been opened. After that he must also open a socket, with all the information gathered of the server, at this moment the client has bound his socket to the server, creating the socket pair.

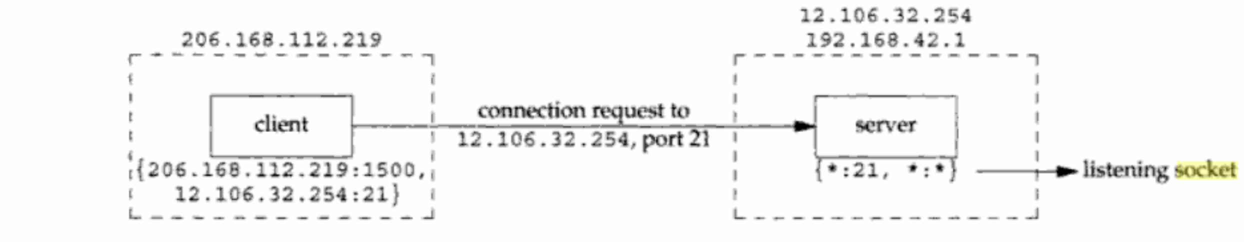


Figure 31 - Connection request from client to server.

The follow code is a simple implementation of a client that can supply us the necessary requirements. Keep in mind that these codes are aiming the UDP connections, so ever using datagram instead of using a streaming TCP server-client model.

```
    memset(&hints, 0, sizeof hints);
    hints.ai_family = AF_UNSPEC;
    hints.ai_socktype = SOCK_DGRAM;

    if ((rv = getaddrinfo(SERVERIP, SERVERPORT, &hints, &servinfo)) != 0) {
        return 1;
    }

    // loop through all the results and make a socket
    for(p = servinfo; p != NULL; p = p->ai_next) {
        if ((sockfd = socket(p->ai_family, p->ai_socktype,
                p->ai_protocol)) == -1) {
            perror("talker: socket");
            continue;
        }

        break;
    }

    if ((numbytes = sendto(sockfd, argv[2], strlen(argv[2]), 0,
             p->ai_addr, p->ai_addrlen)) == -1) {
        perror("talker: sendto");
        exit(1);
    }
```

In fact it must be very clear that the request do not necessarily is made to the same server that will close the connection. It is common that when a server catches a request it calls a *fork*, creating another process that will handle with the connection (as can be noticed in the Figure below).

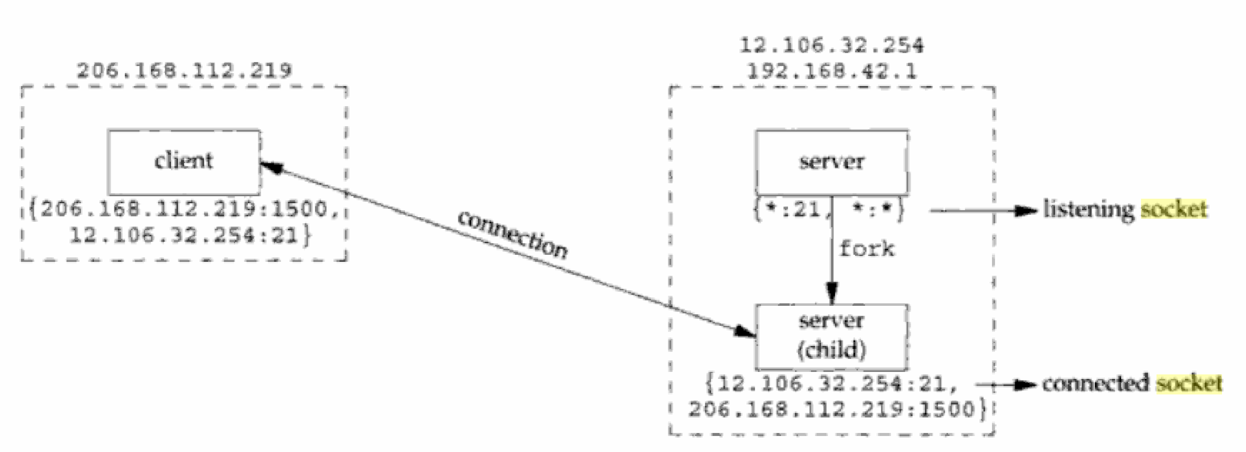


Figure 32 - Current Server has a child handle the Client

This method is then repeated for each new client that tries a new connection with the server (Figure 33). In this way the server can answer individually to each client and it has the control of the senders of the messages that are arriving to him. It is suggest to use design patterns [XIII] as singleton to handle with multiple threads in the application.

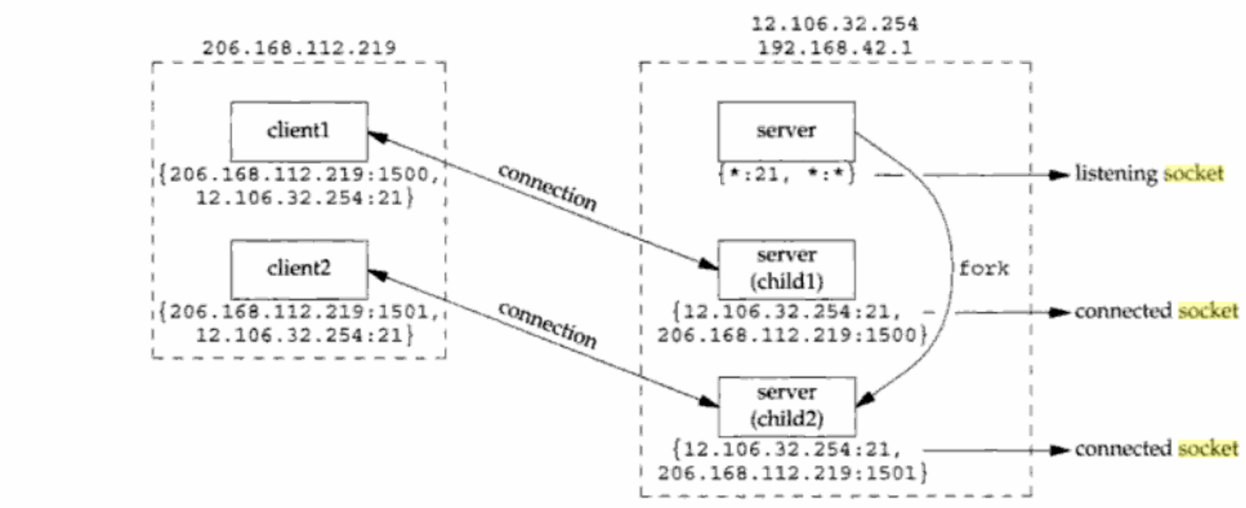


Figure 33 - Second Client connection with same Server

3.3 Organization of network sessions

To begin the Matching is organized in levels, to be more precise four levels: Server, World, Lobby and Room. They are organized as in follow scheme:

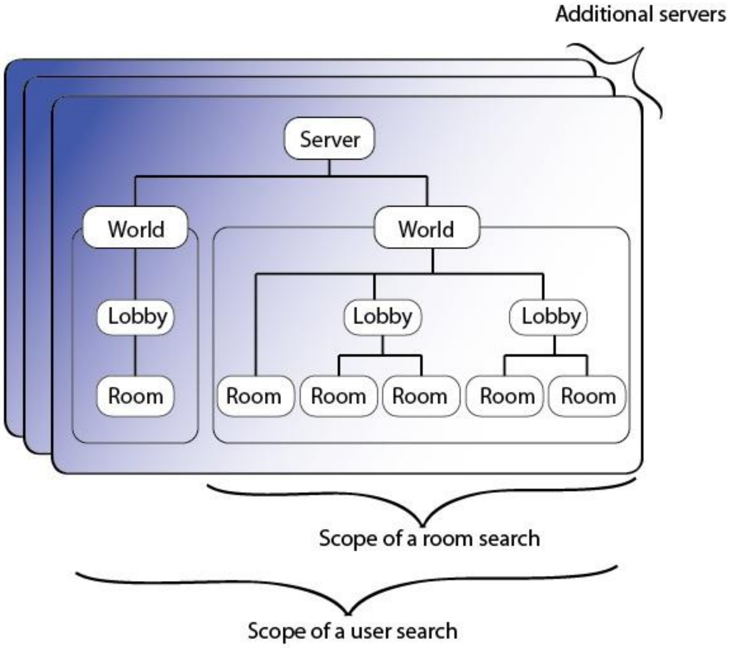


Figure 34 - Components of a network system

User: A user refers to the user of networking system.

Server: The server comprises the highest level of the networking system. It provides match making functions to be used by the applications. One or more servers can be allocated to each application. One server must be selected to be used for the execution of the application; two or more servers cannot be used at the same time.

Server Context; A server context must first be created on the server of the networking system by the user when he/she accesses the system. A user who has created a server context will subsequently be able to access the networking system.

World; A world represents a match making space. Functions of networking system (such as lobby obtainment and room search) are executed for the lobbies and rooms, respectively, which belong to a world.

Session; A session is a collective term for lobbies and rooms.

Lobby and Lobby Rooms; A lobby belongs to a world. It is a space that a user can join, communicate with other members in, and make use of the provided match making functions. A lobby provides functions to enable communication between its members. A lobby can also contain more members than a room. A user in a lobby is referred to as a lobby member.

These several levels allow the development of several different configurations to provide the better experience for a gamer taking care of the specific title. For instance there are three configurations that can be hold: “Searching for a Room”, “Using Lobbies” or “Using All Level”.

Searching for a Room

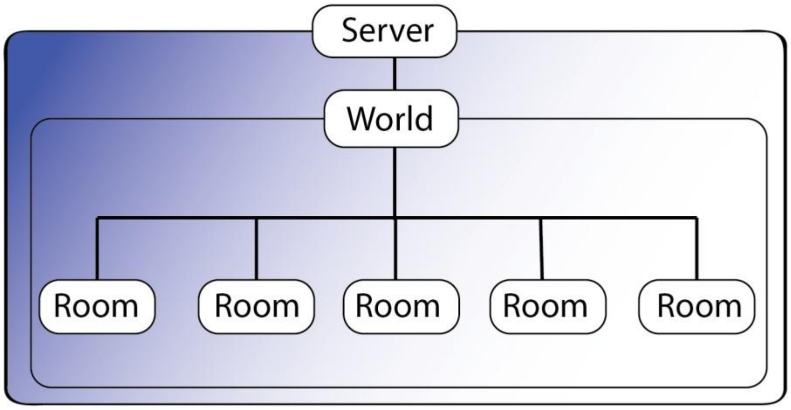
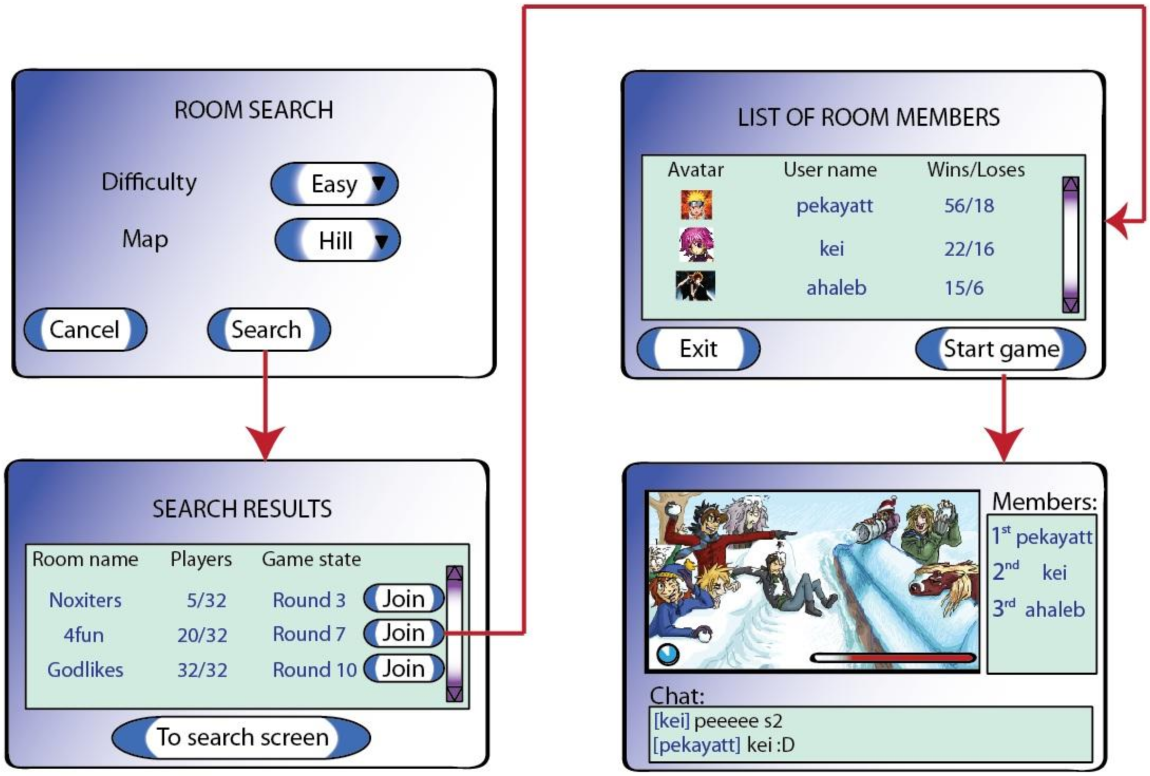


Figure 35 - Configuration when using Room Search Function

At this configuration the user is not aware of the many worlds or servers are used. It is the one who is near from the kind of matchmaking hold by the TrueSkill (see section 2.1.4). Basically the user must set a certain number of attributes of a room and then search for a room that is compatible with these attributes.

The system replies with a list of the many rooms that satisfy the criterion established on the search giving the final answer to the player as selecting the desired room. At this point relevant information, plus the ones gave on the search, are visible in order to show all attributes of the rooms.

From the point that a user is connected to the same room, it is possible to establish a per-to-per connection with others users, and so on exchange data relative to the game. The Figure below is showing a schema that should resemble the basic operation of searching for a room.



ROOM SEARCH

Difficulty: Easy ▼

Map: Hill ▼

SEARCH RESULTS

Room name	Players	Game state	
Noxiters	5/32	Round 3	Join
4fun	20/32	Round 7	Join
Godlikes	32/32	Round 10	Join

LIST OF ROOM MEMBERS

Avatar	User name	Wins/Loses
	pekayatt	56/18
	kei	22/16
	ahaleb	15/6

Members:

1st pekayatt
2nd kei
3rd ahaleb

Chat:

[kei] peeeee s2
[pekayatt] kei :D

Figure 36 - Design of an Application that Uses the Room Search Function

Using Lobbies

The lobbies are in an up level from the rooms, that is, a user must connect first to a lobby and then it will be able to enter in a room. A very popular example of the use of lobby can be found in the Battle.net system [XIV], which was created by Blizzard to support network playing for many of the most popular strategic games.

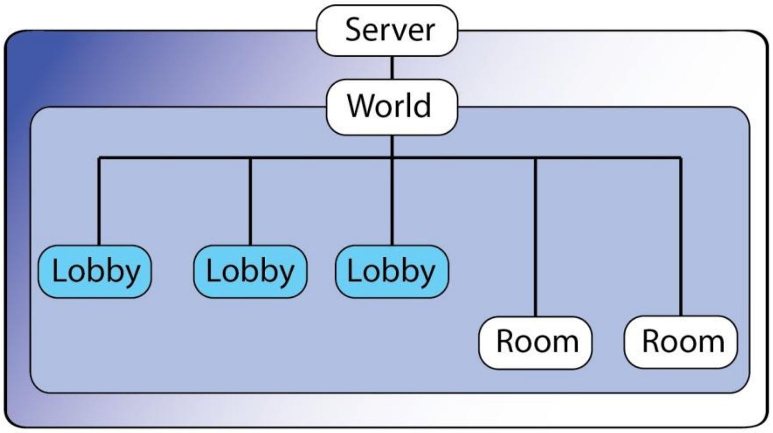
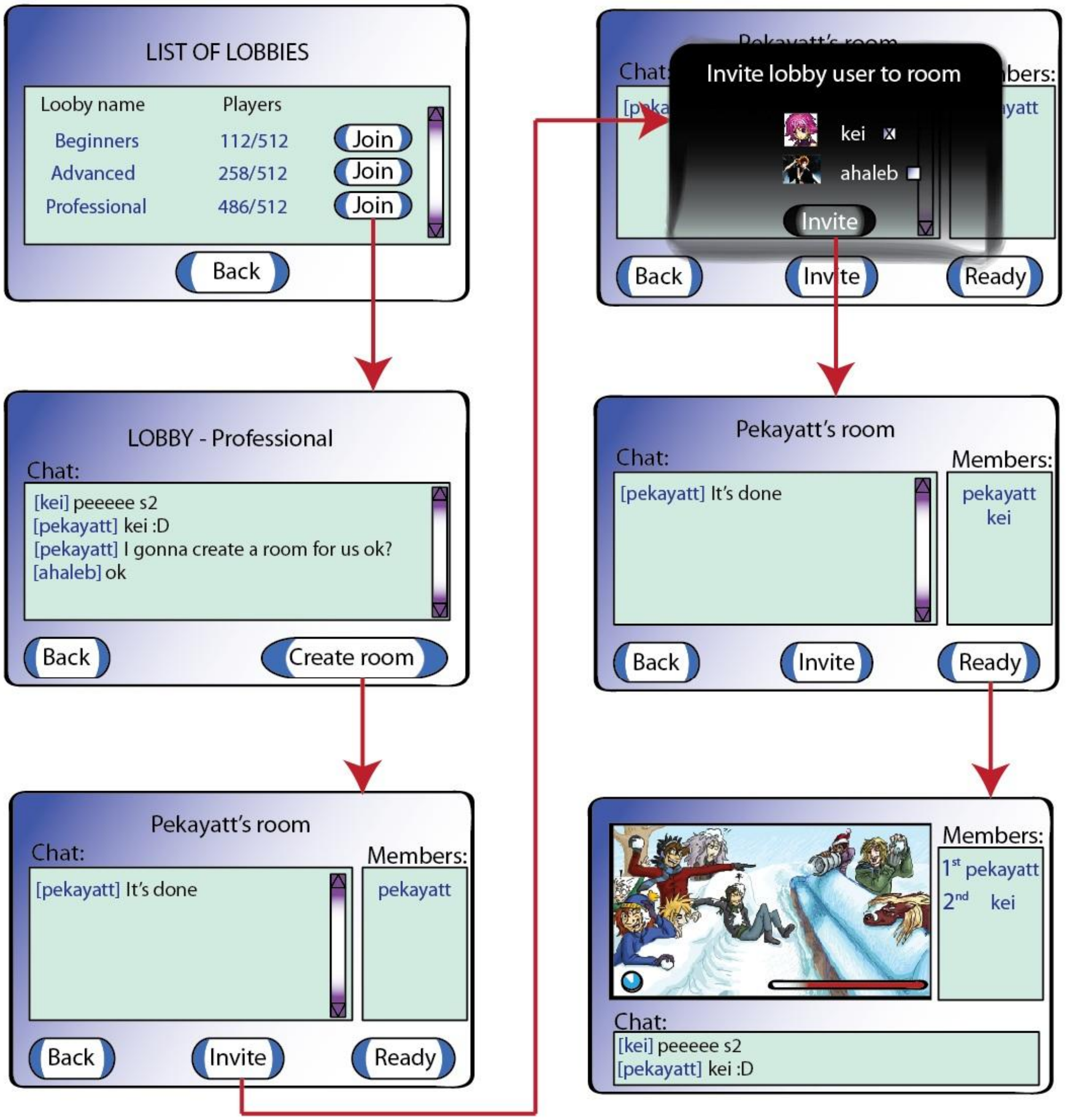


Figure 37 - Configuration Example 2: Configuration when Using Lobbies

After the selection of a lobby, the user may be able to see all the other users that are connected to that lobby and then interact with them, as chatting, and get basic information about each one. The option to create a room must be available, and from the creation of the room interface other users can be invited to the new room.

From this point and further the negotiation runs out in the same way it did before (section *Searching for a Room*) and the players in the same room can start P2P connections and start the exchange of data. The Figure below shows a scheme how it should proceed.



[kei] peeeee s2
[pekayatt] kei :D
[pekayatt] I gonna create a room for us ok?
[ahaleb] ok

[pekayatt] It's done

[kei] peeeee s2
[pekayatt] kei :D

Figure 38 – Configuration Example 2: Design of an Application that Uses Lobbies

Using All Levels

This kind of configuration is hardly recommended if and only if the game is about to have a large number o players online. For instance MMORPG (Massively Multiplayer Online RPG) which have numbers over thousands of players, or even very popular shooter games (i.e. Call of Duty MW2) that requires a low latency to play.

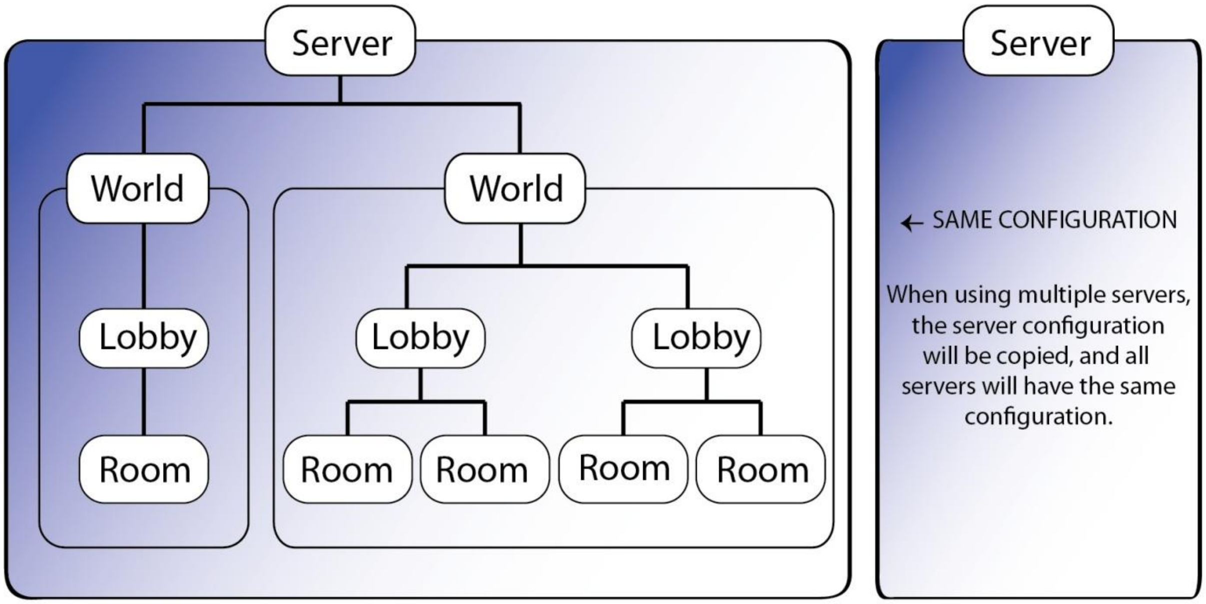
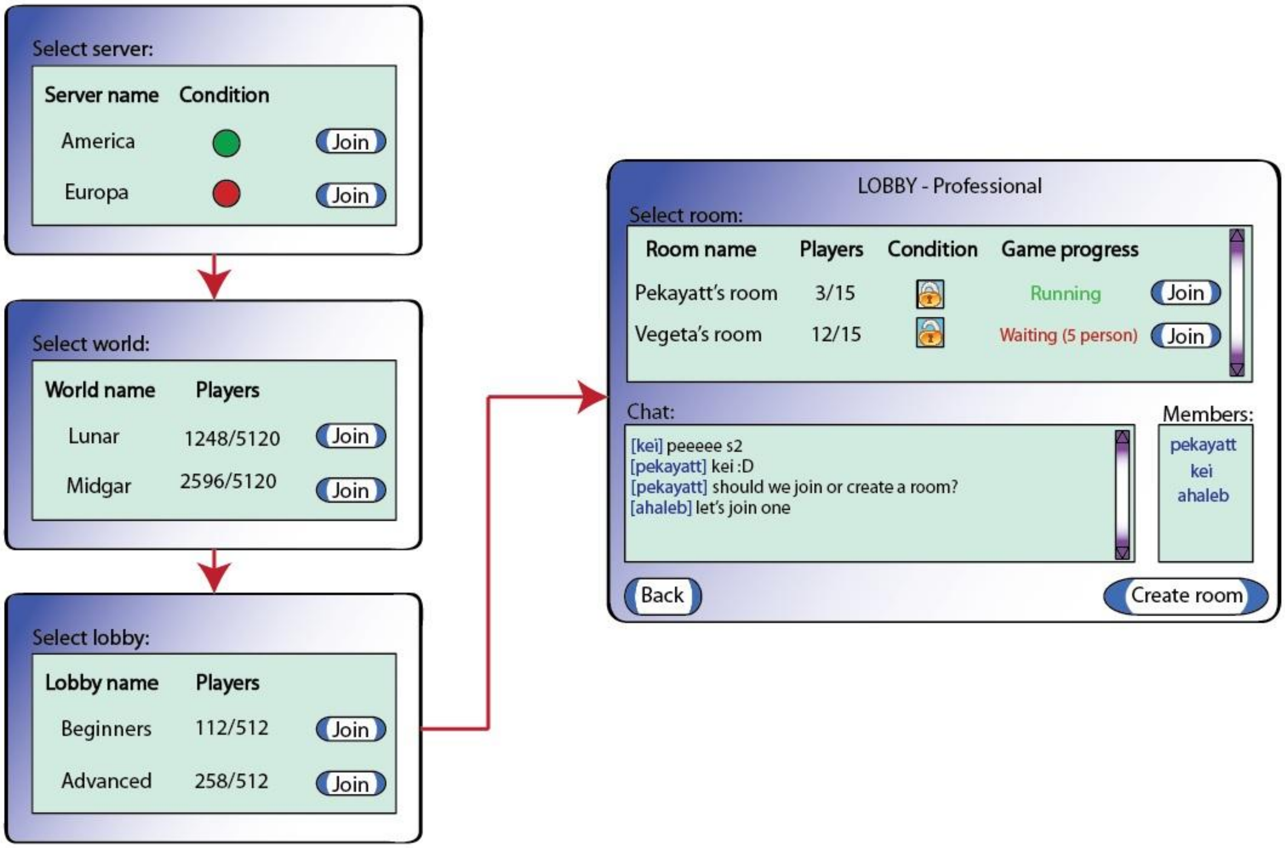

Server
World
World
Lobby
Lobby
Lobby
Room
Room
Room
Room
Room
Server
← SAME CONFIGURATION
When using multiple servers, the server configuration will be copied, and all servers will have the same configuration.

Figure 39 - Configuration Example 3: Configuration When Using All the Levels

In this way server can be created to distingue different locations, as can be observed in the Figure 40, in order to organize players with a low latency between themselves. Furthermore, lobbies that are expressed as the gamers' level (beginner-professional) or different virtual places to enhance the user experience.

The player, then, must select the server, world, and session to join in this order:

- Selects the server from a list of servers.
- Selects the world from a list of worlds belonging to the selected server.
- Selects the lobby from a list of lobbies belonging to the selected world.
- Creates a room that belongs to the selected lobby, or selects and joins a room by performing search among the rooms that belong to the selected lobby.



Select server:

Server name	Condition	
America	🟢	Join
Europa	🔴	Join

Select world:

World name	Players	
Lunar	1248/5120	Join
Midgar	2596/5120	Join

Select lobby:

Lobby name	Players	
Beginners	112/512	Join
Advanced	258/512	Join

LOBBY - Professional

Select room:

Room name	Players	Condition	Game progress	
Pekayatt's room	3/15	🔒	Running	Join
Vegeta's room	12/15	🔒	Waiting (5 person)	Join

Chat:

[kei] peeeee s2
[pekayatt] kei :D
[pekayatt] should we join or create a room?
[ahaleb] let's join one

Members:

pekayatt
kei
ahaleb

Figure 40 - Configuration Example 3: Design of an Application that Uses All the Levels

4 Results – The Network Library

In this section the work developed in the Novecento Games will be explained, some details, however, cannot be revealed both because of the Intellectual Property of the company or because the information is part of the confidential documents of SCE (Sony Computer Entertainment) or Microsoft.

Furthermore, this project was aimed to develop a Network Library that could extend a modified version of the PhyreEngine. PhyreEngine is a game engine of high quality that is distributed by SCE to his official developers and is open to modifications as can be seen in [XV]. Some of the games build over this engine that are worth to be said are: Flower, Race Driver GRID and DiRT (this last two using a modified version called EGO Engine).

Figure 41 - The game Flower by thegamecompany® uses the PhyreEngine

Without entering into details can be said that this engine is designed to be a very modular platform, in special it uses a method to include different functionalities to its normal usage, it takes advantage of a Utilities library.

Each utility extends the features of the engine in order to provide the necessary functions and methods to satisfy the game development. In the case of this project the Network library was developed as a PhyreEngine's utility. And it primary target was the PlayStation 3, extending it after to the Xbox platform.

First is good to clarify that the PlayStation 3 is one of the hardest platforms to program with. The SCE CEO Kaz Hirai has state in 2009: “We don't provide the 'easy to program for' console that [developers] want, because 'easy to program for' means that anybody will be able to take advantage of pretty much what the hardware can do, so then the question is what do you do for the rest of the nine-and-a-half years?”.

Hirai explained in the interview, hold on the 2009’s February issue of the Official PlayStation Magazine, that the reason to be complicated is because it has too much potential: “So it's a kind of - I wouldn't say a double-edged sword - but it's hard to program for, and a lot of people see the negatives of it, but if you flip that around, it means the hardware has a lot more to offer”.

Nevertheless, fortunately the networking programming in the PlayStation is very similar to the socket samples that have been explained in the section 3.2, on the other hand the matching making (Section 2.2.3) is a specific way and is hardly based on system callbacks and it is aimed to be work as a state machine[XVI] set of callings.

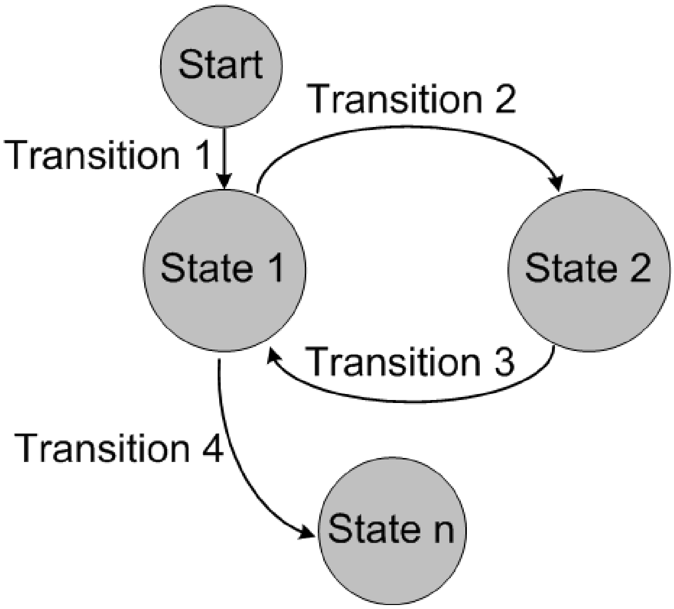


Figure 42 - Example of a Finite State Machine (FSM)

This means that to express the functionalities of the matching making plus the ones of signaling (the exchange of messages by itself) in a utility would request to include a finite state machine into the utility or to kill some features in order of automatic change of these states.

The second option was taken, the development of the library follow in a way that it would have two main levels: an interface, which handle the calling of the desired action of the game; a background state machine, which would be handling with the process and callbacks of the specific platform. Later this choice has showed to be even more suitable since it also could be applied to the Xbox without problems.

The utility was then written based in a stream, in the sense that it would work as a network stream connection, when the game developer wants to open a connection it simple call the function "*open*", this function on its own would take care of login process and establishing a connection with the platform server.

Thus, the follow steps were required in order to connect to another player: join or search a room (we configure the server to use the room directly under the world as is detailed in 3.3); find a player; ready to exchange messages with him. The method open, hence, has received a flag that allow the developer to ask for an automatic join/create, in this way just after the login the utility would search for a room and join the fittest, if no room could be found it would create a room and wait for the next player to join.

Nonetheless at this exactly point it is important to be able to retrieve the information about the actual state of the system, in other words to gather the number of rooms and the attributes of each one of them. For that precisely reason a function *listRoom* needed to be created, as well as a new class that could arrange the main attributes of each room retrieved.

Moreover, inside the class Room a list of players has set, since the information about the players within that room could be very useful for the game developer, coming from getting a ranking average level of the room until the simple reason of the display of the player's name for the user.

In this way another class has been created to collect the player information and main attributes, as done with the rooms, and a *listPlayers* method was added to the room class. As a result the game developer is able to connect to the platform server, list the available rooms, and selection a room he can show a list of players into that room and get their attributes.

After the game programmer complete the actions required to enter in a room the methods *send* and *read* are already available, performing a broadcast message to all the other players that are in the same room as him. The method gets as parameters a void pointer and the length of the data that must be sent, this is to make possible to the game programmer exchange the type of data structure that he wants.

An example of that is that the game programmer could build a structure that has, for instance, a variable *ID* with a type byte, and another *message* as a character limited array. Hence the data must be sent specifying that the size of the message is the *sizeof* of this structure, setting the amount of bits that will be set for the message package.

Another feature that was implemented on the utility was the possibility to send messages directly to a user, using the type Player collected through the *playerList* method. In this way, one another architecture of message could be used, as we have seen in the section 3.1, giving more freedom to the game designer. Is important to keep in mind that these are logical interpretations and no necessarily the message itself is going to be delivered in other ways, since the utility is working in an abstraction level over the networking implementation of the platform.

Besides, the reading of a message can be also done by specifying a user from who you want to receive, adding further option to gather information that can be essential mainly in the handshake process when a player joins a room.

After the library was done, it was added to a sample and with the properly modification it was expected that a player connected by other PlayStation 3 could enter in a room of other player and take the control of an animation. The network was set under the same network sub-net mask, in other words, into LAN and, even if the PSN was needed to access the matchmaking and, thus, the internet, everything was gone well.

Then the test was set to be more close to real-life situation, one of the consoles was then connected to a VPN locally set far then 100km and using other ISP, the latency between the consoles were around 70 to 100 ms, having peeks of 300ms. When tested under this situation the utility start to have some re-connection problems, after playing a match the user hardly could start another match with the same player.

Was noticed that the utility had a problem when closing the application (or rebooting the console to be said). The matter was the non closing of the socket connection, what is not done automatically, and it was solved by simple adding to the destructor a method to close the socket connection.

Also to be sure that no messages would not be lost in the case that the receiver could not process it before arriving a new one (call the method read, before a read data array changes), a circular linked list was implemented using a void array of limited size, keeping in the memory the number of non-zero data set by a define (default was ten). In this way this list would be updated in each new message and it would be cleaned when the read function of the network utility called.

Furthermore, the utility once completed and well tested was then adapted to work with the Xbox network library. Differently from the PlayStation SDK, it has many internal libraries do smooth the progress of development, including a very high level network library. In this way the utility has the same interface that it had for the PlayStation, but for the Xbox it has another layer, between the real socket connections and the utility. This interface has the purpose of simplify even more the set of action taken by the Network utility and is the middle-ware between the Xbox particular library and our game engine.

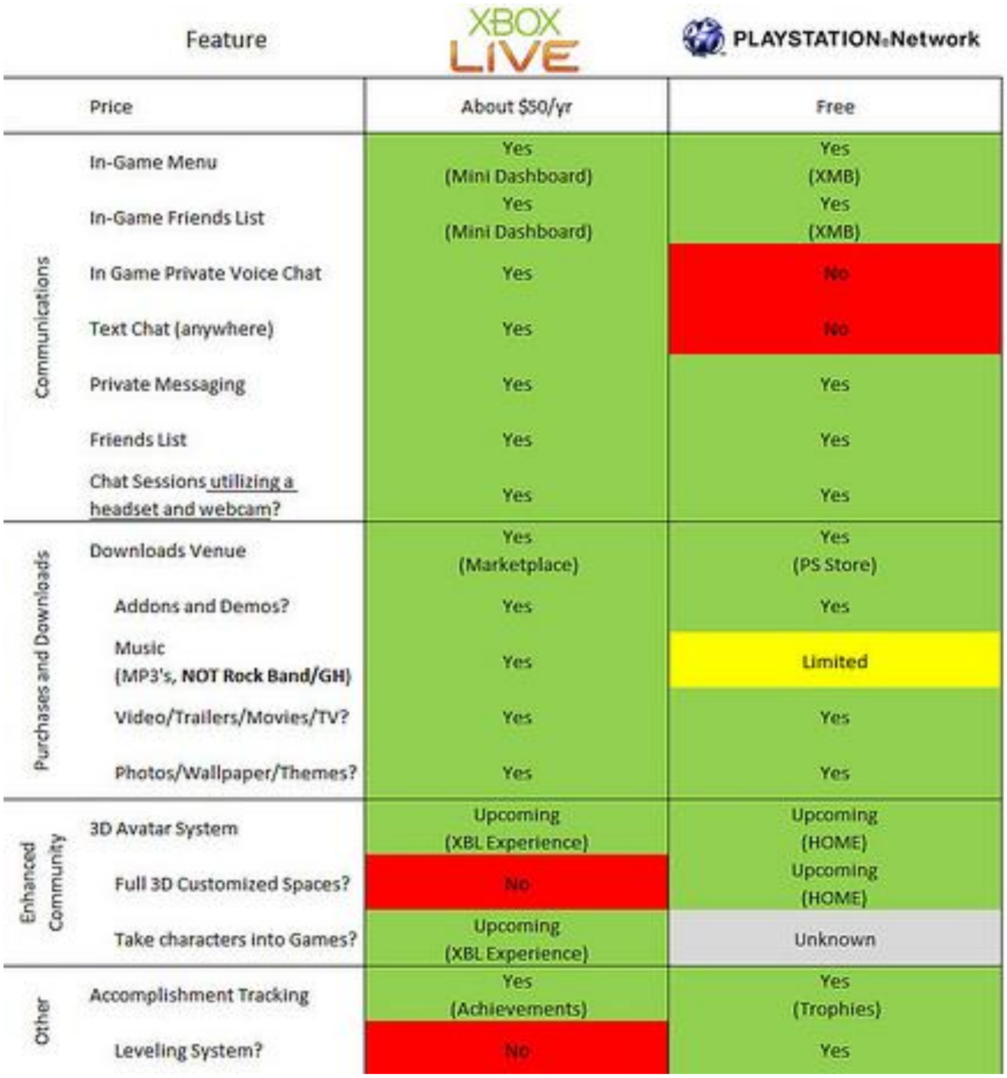
In addition, the Xbox utility has to change some properties of the Room and Player classes, since the main attributes and the interface still the same, for obvious reasons of compatibility, but concepts as the Player ID has changed from the player login of the PSN to the similar into the Xbox Live (see section 2.1 and 2.2).

Initial work have been done analyzing the potential to port the utility once more to the Wii system, which has showed to be worth and suitable to work in the same way as both of the two actual implementation. Unfortunately due time constrains problems it could not be finished.

5 Conclusion

From our top analysis was not hard to pick that the main characteristics of the systems really overlap each other. In one hand we have more sophisticated matchmaking (and ranking) algorithms running over the Live system, and in the other we have a well defined number of configurations that the PSN support, fetching as good as possible to your application.

Even when the both of the systems are compared in the full set of features they are really comparable, as we can see in the Table 5. Each one of then try to keep up creating their “unique” characteristics, such as the PSN Home, or the in-game voice chats of the Xbox Live.



	Feature	XBOX LIVE	PLAYSTATION Network
	Price	About $50/yr	Free
Communications	In-Game Menu	Yes (Mini Dashboard)	Yes (XMB)
	In-Game Friends List	Yes (Mini Dashboard)	Yes (XMB)
	In Game Private Voice Chat	Yes	No
	Text Chat (anywhere)	Yes	No
	Private Messaging	Yes	Yes
	Friends List	Yes	Yes
	Chat Sessions utilizing a headset and webcam?	Yes	Yes
Purchases and Downloads	Downloads Venue	Yes (Marketplace)	Yes (PS Store)
	Addons and Demos?	Yes	Yes
	Music (MP3's, **NOT Rock Band/GH**)	Yes	Limited
	Video/Trailers/Movies/TV?	Yes	Yes
	Photos/Wallpaper/Themes?	Yes	Yes
Enhanced Community	3D Avatar System	Upcoming (XBL Experience)	Upcoming (HOME)
	Full 3D Customized Spaces?	No	Upcoming (HOME)
	Take characters into Games?	Upcoming (XBL Experience)	Unknown
Other	Accomplishment Tracking	Yes (Achievements)	Yes (Trophies)
	Leveling System?	No	Yes

Table 5 - PSN vs. Xbox Live feature comparison (source: Leimeisel 2008)

Unfortunately other systems could not be directly compared for the reason that they have a different purpose or focus. For instance the Nintendo Wi-Fi Connection is not responsible to fulfill many features that both PSN and Live claim to be essential, in the mean time the Nintendo has develop other systems to supply their clients, like NintendoWare or Virtual Console.

Another good point that must be stressed is that even if the consoles have a totally different architecture, they are not machines of general purpose as a PC (personal computer); they still have the same basic network principles. This is due the fact that the Internet is rule by the same protocols wherever it is being accessed.

Thus, all the architectures and studies accomplished to computer systems can be applied into the development of a console network library. Nevertheless to take a game’s genre is really important into the considerations of the library given that the trade-off between the latencies and data’s concision is well defined for each choice.

Notwithstanding, it was showed that is possible to create a balanced middleware in an abstraction layer above the specific console libraries that handles with the basic network. The development of a well structured middleware like that can improve the development process of a game, in exchange of some performance.

In the actual scenario it is clear that the online gaming cannot be ignored. Not only the major consoles are investing in that field, but many systems start to appear in other games platforms, which are worth to be cited, as the Steam [XVII] (Windows and Mac OS) or even the Apple Store (IPhone). Some new endeavor are also being made into the clouding game network systems [XVIII], such as the OnLive and the Gaikai; that claims to make possible to play any game anywhere and with any platform.

6 References

I - James K. Galbraith and Travis Hale - Income Distribution and the Information Technology Bubble, 2004

II - Doom 3 Team Development Story - http://www.youtube.com/watch?v=X8xvTJ6msqY

III - "Cost of making games set to soar" - *BBC News*. November 17, 2005.

IV - “Official PlayStation Website: PlayStation Network - PSN” - http://uk.playstation.com/psn/

V - Ondrejka, Cory R. - A Piece of Place: Modeling the Digital on the Real in Second Life

VI - S. Singhal and M. Zyda - Networked Virtual Environments: Design and Implementation. Addison Wesley, 1999

VII - Mark Claypool - Network Characteristics for Server Selection in Online Games

VIII - Michael R. Macedonia and Michael J. Zyda - “A taxonomy for networked virtual environments,” IEEE Multimedia, vol. 4, no1, pp. 48–56, 1997.

IX - Steve Deering - “Host extensions for IP multicasting,” - Internet RFC 1112, Aug. 1989, ftp://ftp.isi.edu/in-notes/rfc1112.txt

X - P. Srisuresh & M. Holdrege - IP Network Address Translator (NAT) Terminology and Considerations – Internet RFC 2663, Aug. 1999

XI - W. Richard Stevens,Bill Fenner,Andrew M. Rudoff - UNIX Network Programming: The sockets networking API

XII - Brian Hall - Beej's Guide to Network Programming

XIII - Stephen D. Huston, James C. E. Johnson, Umar Syyid - The ACE programmer's guide: practical design patterns for network and Systems Programming

XIV - Blizzard Entertainment. *Battle.net* - http://www.battle.net, 2010

XV - "SCEI March 2009 (GDC) PhyreEngine Press Release" – SCEE - Mar 24, 2009.

XVI - Minsky, Marvin (1967). Computation: Finite and Infinite Machines (1st ed.). New Jersey: Prentice-Hall

XVII - Valve, Inc. - "Steam," http://www.steampowered.com/, 2005.

XVIII - A. Shaikh, S. Sahu, M.-C. Rosu, M. Shea, D. Saha - On demand platform for online games - IBM Systems Journal, 2006 - Citeseer