The Effect of Latency and Network Limitations on MMORPGs
(A Field Study of Everquest2)

Tobias Fritsch
Freie Universität Berlin
Takustrasse 9,
D-14195 Berlin
+0049 838 75250
fritsch@inf.fu-berlin.de

Hartmut Ritter
Freie Universität Berlin
Takustrasse 9,
D-14195 Berlin
+0049 838 75214
hritter@inf.fu-berlin.de

Jochen Schiller
Freie Universität Berlin
Takustrasse 9,
D-14195 Berlin
+0049 838 75212
schiller@inf.fu-berlin.de

ABSTRACT:

Networked games are becoming more and more important in the last years. One of the current game evolutions is creating massive virtual environments, so called MMORPGs (massive multiplayer online role-playing games). Those game types exist for more than 10 years, but the market and the number of products grew incredibly strong in the last two years.

With such an evolution comes the need for new techniques to cope with special problem fields such as effects of latency with 200 and more players. Thus understanding and pointing out the new problem fields will help to design fitting solutions for expansions like a mobile MMORPG version.

This paper will give a brief introduction into the MMORPG history. Furthermore it will evaluate certain latency problems in a field study of Everquest2, referring to the previous work at "Overviewing Scientific Research for (Mobile) Gaming" and point out possibilities to design MMO(RP)Gs in a mobile content.

General Terms

Measurement, Documentation, Performance.

Keywords:

Network Games, MMORPG. Everquest2, Mobile Gaming, User study

NetGames'05, October 10–11, 2005, Hawthorne, New York, USA.

1. INTRODUCTION

In the last years the gaming market significantly became more and more important. Nowadays, games are already being one of the major parts of the electronic entertainment industry, with their importance still growing.

One major growing part of the gaming industry is the MMOG (Massive Multiplayer Online Game) Sector. The first MMOG was released in September 1996 [1]. In April 2003 there are 51 MMOGs available for game players worldwide, today more than 120 MMORPGs exist. Most of them focus on a fantasy theme and are so called MMORPGs (Massive Multiplayer Online Roleplaying Games) [2].

Those game types open a completely new set of problem fields due to their massive amount of players, such as hosting the game environment on multi server architectures or coping with latency at several game events with 200 or more players. To solve the current problems and combine the ideas of massive and mobile gaming it is necessary to understand the current network structure and the effect of latency on the game.

Another aspect is the huge size of the game worlds, taking multiple servers to host the complete virtual environment (VE). Thus, techniques for separating the world into different parts are needed. Two main game structures exist since the first MMOGs. The simpler attempt towards implementation is the zone based structure, which basically separates the whole game world into different zones (smaller parts). Those are small enough to host them on a single server. The zone borders are connected to each other and the player will connect to the next server once he or she is changing zones. A big disadvantage of this attempt is the loading time between each zone.

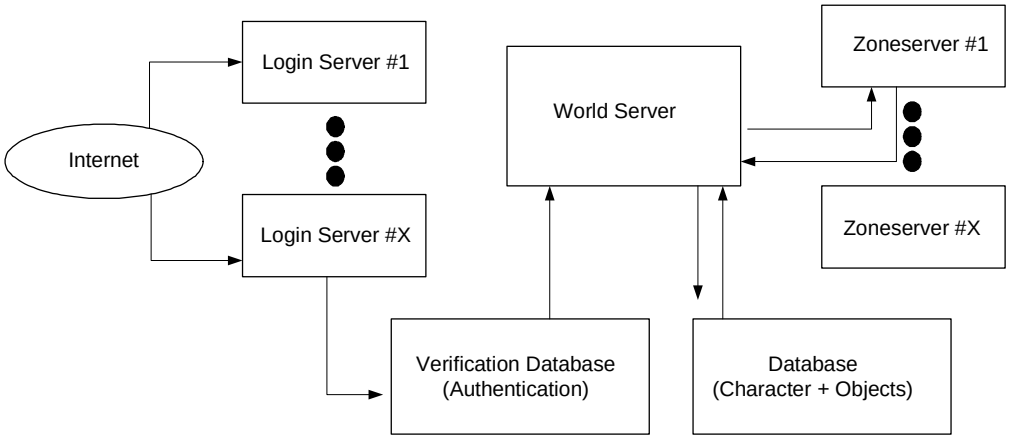


Figure1: An example of a multi-server architecture.

The other attempt of implementation is the seamless environment, where zone borders do not exist. This will give the player the feeling of a completely seamless world that never ends. The game world never the less is separated. The important connecting areas are hosted on two different servers. Once the player leaves the area of server #1 there will be a zone border area, which is synchronized. The handover to server #2 does not happen immediately, it happens after the player has completely crossed the border zone. The big disadvantage of this attempt is the high programming complexity and the growing traffic in case of events in the border zones. Another problem might be a lack of AI that allows the player to "trick" its environment by standing in the area of another server, while static monsters might not be able to pass this border.

This paper focuses on Everquest2 as an example for a 2nd Generation MMORPG, giving a brief overview about the game's background in Section 2. The related testbed, its design and the results are discussed in Section 3. In Section 4 further problems will be analyzed, including security related issues, such as the possibility to create a distributed architecture for a MMOG. This will lead to Section 5, which includes a conclusion and outlook for further work.

2. BACKGROUND

Everquest2 is a so called 2nd generation MMORPG. This means its evolution has been influenced by previous knowledge of common issues in the MMOG scene. Thus, it has several aspects of its predecessor Everquest and shares the same game universe. It also tends to improve known issues such as instancing (creating instances of important areas that allow the players to fight certain monsters without the fear of getting a kill stolen by others) or sharding (a shard is a copy of a zone, which enables a certain amount of load balancing and helps not to overfill parts of the world [3]).

Like any other MMOG Everquest2 has no LAN or single player support. Therefore, it focuses completely on Internet gaming with a zone-based approach. The design fits an expectation of 200 Player per shard (zone copy). As a result of the world's size the network structure is a basic multi-server approach (see figure 1). There are multiple servers with a common total load of 10.000 to 15.000 players on each realm.

The goal of Everquest2 is to gain experience and a higher character level (you start at level 1 and you can reach up to level 50), better equipment (certain monsters drop better loot than others, the more players are needed to take down a monster, the better the loot is) and access to restricted areas. A good player therefore has the skill to gain a high amount of experience in a short time and obtain good items faster than others can. Further discussions about the measurement of player skill will follow in section 3.

As the game is designed for a raid environment (Raid = massive amount of players formed up to beat a single challenge) the relevant scientific topics refer to the effects of latency for multiple players, the strategy to cope with massive amounts of players at the same time and the consistency of the game world. Like with any other real-time application, latency has a huge impact, because it can influence the outcome of fights or the reaction of the game environment.

Figure 2: A Raid Force is fighting an epic battle (24 Players vs. the environment)

3. RESEARCH APPROACH

In order to scientifically understand the effects of latency on certain game situations we employed the following methods:

3.1) Point out the important game interactions for Everquest2, categorize them and build valid test situations for all of them.

3.2) Build a fitting test environment that can influence the latency and other aspects if needed. Monitor the effects on the game.

3.3) Perform a fitting test for each important game aspect. Each test series should be independent to its own hypothesizes and results.

3.1 GAME INTERACTIONS

Although the game concept of Everquest2 has far more than a short time goal it is hard to point out a certain aspect which will make the player "win" this game. There exists no final goal in the game and thus there can be only some aspects mentioned which make a player "better". To understand what a "good" player in Everquest2 is, we concentrated on the endgame goal, which is PvE (Player vs. Environment) – based. Most of the available ingame content is developed for level 50 characters to keep them busy.

Understanding this idea of an "endgame" (the game at level 50) leads to the following characteristics:

A "good" Player

- knows the location of all monsters and items.

- dies less often because he or she moves more intelligent

- fights faster and survives more often because of a better combat management.

- understands his/her character's role and plays his/her character most effectively.

- can work with the game engine to build up cooperative groups faster than others.

In Section 3.3 we analyze those aspects and develop appropriate research approaches. Not all of the aspects can be measured by testing though, but still have an impact on the game. Thus they will be taken into account when redesigning the network structure (for detailed information refer to section 5).

3.2 THE TESTBED

The Testbed consisted of two PCs and a switch with a NISTNET controller. Both PCs are connected to a switch, which is controlled and monitored by a Linux running computer with NISTNET[4] installed. NISTNET has several options to modify the incoming and outgoing data stream such as delay, jitter, bandwidth and many more. That way you can freely add latency spikes, jitter and packet loss to the connection and monitor the results. Figure 3 illustrates the testbed setup. Because both computers were in different subnets it was possible to individually influence their data stream.

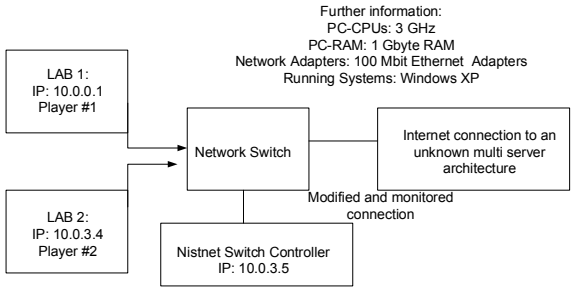


Figure 3: Testbed setup.

3.3 TESTS AND HYPOTHESIS

As already mentioned, some of the aspects of a "good" player are not analyzable. Because of that it is important to at least exclude the influence of those aspects by selecting matching players and environments.

We therefore decided to let both of our participants (expert players with high skill and the maximum available game experience) play on similar PCs and gave them the exactly same equipment and character class (except in group combat test).

The latency (connection) dependent game aspects are player movement and player combat behavior – both of them have a huge impact on the outcome of the game. Three main tests were set up:

(1) Movement & Combat:

The movement & combat test indicates the direct reaction of latency for the game world. Both participants started at the same point, had to slay exactly 3 monsters and then start to run through a hostile area until they reached the goal. Both of them were instructed to kill and run as fast as possible.

This test combines the idea of effective combat behavior with a movement component (dodging monsters in a hostile environment). After each run, the time of both participants was taken – there were runs with 0ms, 250ms, 500ms, 750ms, 1250ms, 2000ms, 3000ms and 5000ms latency – both players had the same latency induced on their runs.

Hypothesis: The more latency will occur, the longer the fights will take. A latency of 500ms was expected to be a "breaking point" from where the game turns into not-playable. It was also expected that the movement is less influenced by latency than combat is. Furthermore, the number of deaths and the damage taken while running through the hostile area will increase with the latency as well. Concluding this, it is expected that the time to finish the movement and combat test will proportionally increase with the induced latency.

(2) Group Combat:

This test focuses on the combat aspect. Therefore the participants received a healer and fighter combination (a basic class combination that can force encounters effective) and had the same

startup gear and level. Both disabled their experience gain (to prohibit level gain in the middle of the fight – which would make them stronger and/or completely heal them).

A fixed and challenging monster-encounter was picked, and the players were to fight them under different latency influences. Again the induced latency was 0ms, 250ms, 500ms, 750ms, 1250ms, 2000ms and 5000ms. After each fight, combat length, remaining health (hit points) and remaining mana (magic points that are needed to cast spells) of each player was taken.

Hypothesis: It was predicted that the latency will have a huge impact on the combat. The "breaking point" was assumed to be at 500ms again. The combat length will rise proportionally with the induced latency; meanwhile the remaining health and mana will have an antiproportional dependence on the latency.

(3) Movement/Environment:

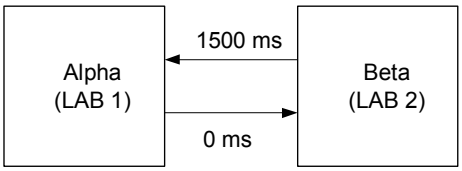
This research approach was designed to give more information about movement prediction and information hiding in Everquest2. In the first part, only one of the recipients suffered from induced latency. The other recipient was moving towards him and similar, time stamped screenshots were taken. The test setup was 1500ms incoming, 1500ms outgoing and 750ms both sided latency (see figure 4).

The second part had a similar setup. Both players took environment screenshots (time stamped, same perspective and same viewpoint) to investigate the effect of latency on the game environment. Again it was 1500ms incoming, 1500ms outgoing and 750ms both sided latency taken in account.

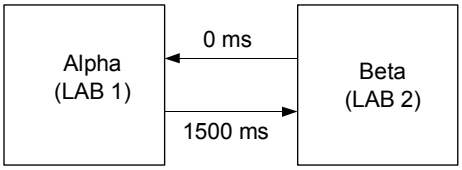
Hypothesis: A MMORPG will probably need a dumb client model, because the number of recipients that can see a player might differ from situation to situation. Especially similar gaming with 30 or more other players will cause serious movement issues if every single player needs to verify every move.

The environment on the other hand should be at least partially server sided, because of the information hiding approach. This helps to prevent the design of combat radars or custom tracking tools.

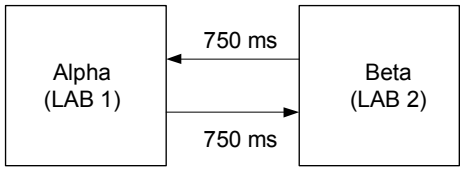
(3.1)



(3.2)



(3.3)



4. ANALYSIS

4.1 Movement & Combat:

Our hypothesis has proven to be partially correct. The game expectedly turned out to be non-playable at very high latency sequences. The "breakpoint" (point from where the game was no longer fluently playable) however occurred at 1250ms, which was more than two times higher than expected. Two main reasons are responsible for this effect. On the one hand the movement seems to be rendered locally (a detailed analysis of rendering and movement can be found in section 4.3). On the other hand the combat seems to react pretty accurate even at 500-1000ms. The Everquest 2 player combat system is based on a special queuing mechanism that allows the player to queue up to one combat move. This action is already sent towards the server, and as soon as the current combat move is done, the new ability is instantly executed. This makes even players with higher latency feel like they got no delay. Because most abilities take around 1 second to perform, it is easy to understand why the combat turns out to become very problematic at latency times of more than one second.

The time measurement (figure 4) shows a proportional grow of overall time with rising latency. At the player interview both participating testers stated that the effect of latency was drastically influencing the combat meanwhile the movement seems to be excellent up to 2000ms. With an average player latency at around 200-500ms the game influencing behavior obviously does only occur in special situations such as raid instances (24+ player fighting at the same time) or in crowded areas.

Figure 4 shows the overall time that both testers needed to finish the Combat & Running test. The test itself has been repeated three times with the exactly same starting conditions. The result for every single latency step is the median of all three test results. It clearly shows a proportional relation between the latency and the overall time.

4.2 Group Combat:

The combat part of the running test already indicated that the impact of latency on fights will be much higher. Thus our hypothesis has proven correct, except that the problems show up at 1250ms again. The combat time grew proportionally with the latency, the single parts of the fights took longer and especially the first contacts and placement was very much influenced by the latency. With growing latency (especially over 1000ms) the abilities were no longer used coordinated.

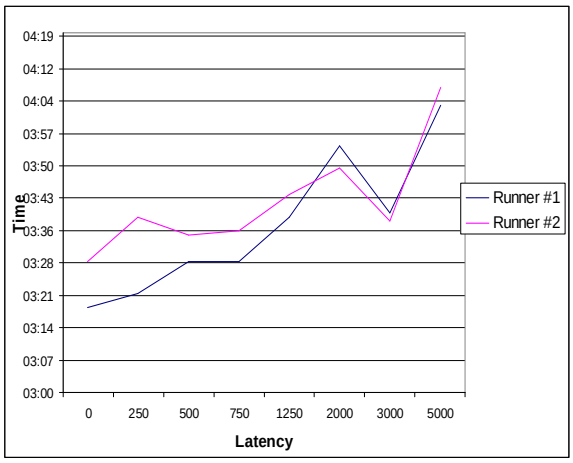


Figure 4: The overall time in relation to the growing latency.

Figure 5 shows this effect, especially at higher latency times (1250ms or more) the players' DPS (damage per second) went down massively. The main reason was the missing ability to control the monsters. As soon as the environment has started to move and dynamic fight elements occurred the players were no longer able to stay in fighting range because of the outdated placement data.

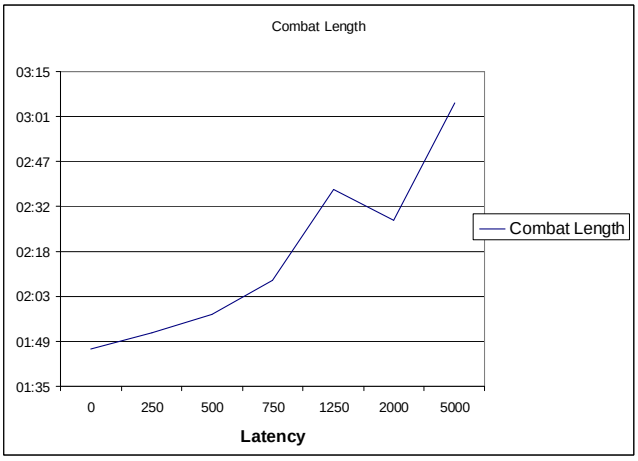


Figure 5: Overall combat time

Another result of the growing latency was the antiproportional relation between remaining resources (which had been protocolled after each fight) and induced latency. The overall results hint at such a relation, but the value of mana points compared to health points influenced the statistic. Figure 6 shows the remaining health and mana after each fight (Player #1 & #2 separated). Especially after the "breakpoint" of 1250ms there is a notable change.

Various game aspects, such as a healer being able to restore hit points in a 2:1 ratio (meaning that for every single mana point two hit points are healed) lead to the conclusion that manapoints have a greater impact on the fight. Thus we created an overall resource function **r(x) = (h(x) + 2*m(x)) / 3**, where r(x) denotes the remaining resources for X ms latency, h(x) the remaining health, and m(x) the remaining mana. Figure 7 shows the resulting function of the remaining resources. The new resource function shows the relation between latency and remaining resources quite clearly.

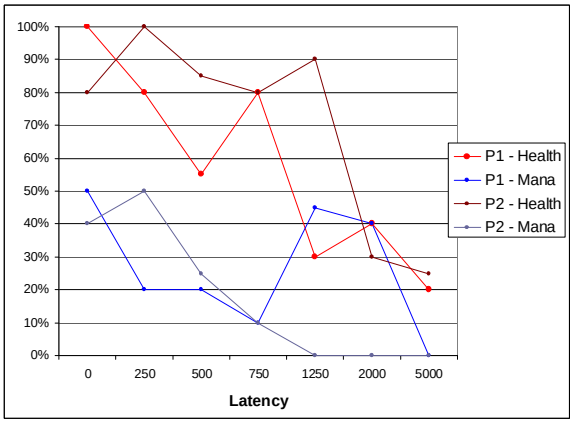


Figure 6: Remaining health and mana after each fight (player #1 and #2).

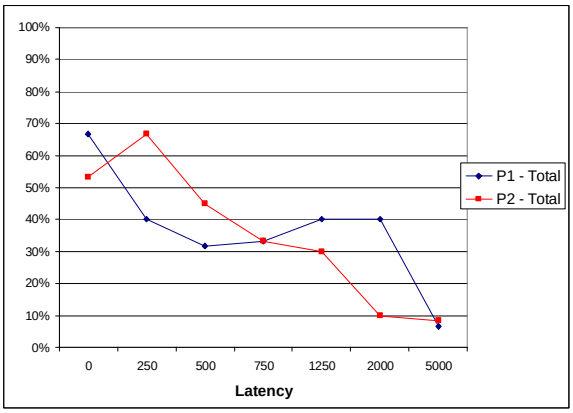


Figure 7: Remaining overall resources (r(x)) after each fight.

Both participating players knew about the encounter and were prepared (they knew the abilities and strength of their opponents). When summarizing the results, it is also important to mention, that the player behavior in combat has changed. The reason for a relation between latency and combat length/remaining resources is, that the players were not able to perform coordinated attacks with growing latency. As a result, the players had to rethink their tactics and had to adjust their playing style (e.g. cast healing spells even earlier or change to faster, less mana efficient spells). Thus an unprepared group with a latency that high, would probably have lost the battles. Figure 8 shows a typical latency situation: Player #2 has already sent the command for his healing spell several seconds ago and due to latency the spell has not yet executed – Player #1 is constantly taking damage in the meanwhile.

4.3 Movement and Environment

Besides the combat and bandwidth research approach it is important to understand why movement and rendering is still working even at very high latency times. Especially in a MMOG content it is important to keep accurate track of all players. Furthermore, an accurate measurement can provide useful information about possible improvements (like cheat protection or improved latency burst behavior), especially when analyzing different network structures (see section 5).

The first part of the movement test indicates that player Alpha has already moved much further on his screen than on Beta's screen. The screenshots were taken with 1500ms induced latency from the Internet to Beta. Figure 9 shows the difference in placement.

Figure 9: Different rendering with latency

This rendering difference indicates a dumb-client model, with the rendering performed locally and movement commands sent and acknowledged through the network afterwards. (see figure 10):

To validate this hypothesis, the same test setup has been repeated with 1500ms outgoing latency (from Beta to the internet) and with 750ms latency both sided. For more information about rendering and prediction models refer to [5], [6].

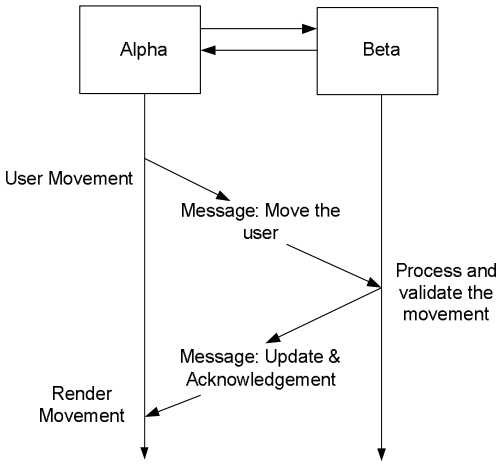


Figure 10: Dumb client model. (shows only clients for clarification)

The test with 1500ms outgoing latency validated the hypothesis, because the rendering is displayed the same position. The last part (750ms both sided) showed that Alpha was misplaced by about half the distance compared to the first approach with 1500ms incoming latency. This result confirms that Everquest2 is using a pure dumb-client model approach. This seems logical in a MMOG, because the own moving flow would be disrupted very much when 200+ acknowledgements are needed to proceed.

Nevertheless, the dumb client approach also has disadvantages, especially concerning a distributed network structure with local information management.

The second part (environment test) should determine if the environment is also rendered locally or server based.

The results show that with an incoming latency of 1500ms (latency from the internet to Beta) the environmental movement of Beta lags behind Alpha's. With 1500ms outgoing latency (from Beta to the Internet) both players' movements were accurate and showed no significant differences. With 750ms both sided latency the difference was again half as much as with 1500ms incoming latency. Still, the players own movement was rendered instantly.

This clearly indicates that the zone server is taking care of monster movement and placement, while the client only receives information within his viewing distance. This information hiding technique is based on development experiences in the MMOG sector and several vision and radar cheats in the predecessor Everquest2. Of course we did not want to let Everquest2 look like a simple peer to peer game. In fact figure 10 does not show the interaction of the server itself to simplify the results from one client to the other, but that does not make the game peer to peer based. The server is still part of the position updates and especially important when it comes to movement corrections or other position decisions.

4.4 Network Level

Another aspect is the overall bandwidth consumption of an MMOG. Especially when thinking about different network

structure approaches it is important to keep an eye on the required bandwidth. In a mobile content an increase of the end systems is expected, meanwhile the maximum data rates will not increase in a similar way.

We therefore traced thirty minutes of Everquest2 in a medium populated zone. Figure 11 shows that the average download rate was 0.9 kbit/s and the average upload rate was 0.4 kbit/s with a maximum burst of 4.2 kitb/s download and 4.7 kbit/s upload.

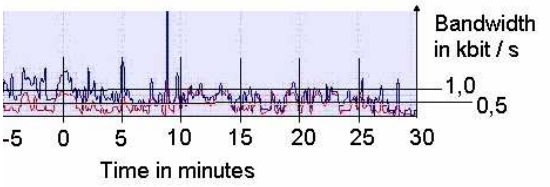


Figure 11: Bandwidth in an average crowded environment.

Afterwards the game was traced during heavy fighting in a crowded environment. The packet size remained the same, while the number of transferred packets almost doubled. The average download rate was 1.5 kbit/s and the average upload rate was 0.9 kbit/s. Furthermore the bandwidth graph shows multiple bursts with up to 2.6 kbit/s download and 1.4 kbit/s upload. Figure 12 illustrates the bandwidth.

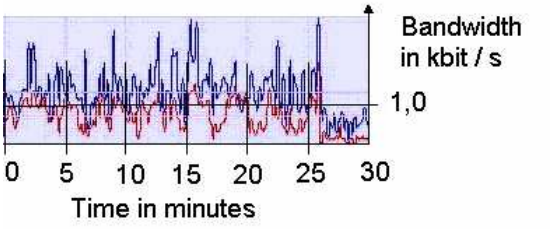


Figure 12: Bandwidth in a heavy crowded and fighting environment.

Concluding, Everquest2 does not use a bigger packet size in burst situations, instead each event seems to be handled within its own package. The low overall data transfer rate has advantages for mobile networks (further discussion in section 5).

4.5 User Qualities

The last aspect of each research approach is the user quality level (the knowledge and skill that the participating test-users have). Although the pre-knowledge gives a user a huge advantage in an MMORPG world, we decided to have expert players with the maximum pre-knowledge to play the game. This way both players had the exact same starting conditions and equal levels of combat behavior, as well as game speed. Especially in group combat situations the players' background experiences helped to get the most accurate time results (a big aspect of the game is a repetitive fighting behavior, which "optimizes" the players combat skills, thus the players did not need to test different tactics, they already knew how to proceed in a best possible way).

4.6 MMORPGs on a mobile content

Another important part of the research is to take the current games to another level. Several game types already exist in a mobile version (such as car-racing games or simple jump and run games). With the release of more complex handheld devices such as the Nintendo DS and Sony PSP new options are offered.
The bandwidth and latency results of the Everquest2 clients show, that even with limited network conditions (such as mobile networks) the game should run stable. Never the less the server side calculation (multi server architecture) is not taken into account within that conclusion. To completely understand this new game field one should both examine the first generation of so called mobile MMORPGs (like Fantasy World Rhynn – or Pan Asia Games) and implement given approaches about peer to peer MMORPG designs (such as [14]). Both of that are tasks for our further research.

5. RELATED WORK

There is plenty related work for Everquest and other MMORPGs. Most of the scientific approaches are members of one of the following three main approaches:

(1) Ingame social content and evolution:

This approach focuses mainly on social ingame behavior. Such games create a whole new interaction set, including some behavior rules and communications forms. That is because MMORPG - VEs have a huge amount of human players and several game tasks require cooperation. There are even special design approaches that support social interaction, giving it an overall high priority. The Ingame social content might be the most reviewed part of MMORPGs in a scientific way. The main difference is that most papers are completely focusing on socials aspects and thus leaving out technical details. For more information about ingame social content refer to [7], [8] and [9].

(2) MMORPGs and their impact on the real world:

Besides other specific games (like Diablo2) it is a well known fact that electronic items from MMORPGs have a real world value. Players tend to pay money for ingame advantages, new game features or rare items. There are also other forms of MMORPG and real world relations, ranging even to low budget 3rd world workers, who play for ingame cash. Such relations are focus of a series of papers, including [10] and [11]. The financial approach focuses on business aspects and leaves social and technical parts of MMORPGs beside.

(3) Gametype analysis / user experience:

This research approach includes both game reviews and improvement of certain game elements. A good set of game reviews is done by Mark Claypool, including [12], [15] and [16]. The approach mainly focuses on a user perspective of the chosen game(type) and tries to point out what common aspects games from a similar type share. Furthermore gametype analysis provide

further research projects with important up to date research data, such as maximum tolerated network limitations (from a end users point of view).

Even with a massively growing community there is no fitting MMORPG network review, especially not from a second generation MMORPG (so called second generation because the game designers learned from the predecessors). Other research approaches [13] are improving the network variety and therefore open new possibilities (like server independent MMORPG gaming).

(4) Mobile games:

Those research approaches aim on an improvement of mobile conditions to fit the game restrictions better. The harder the restrictions are the more difficult it is to provide a fitting solution for a mobile version of the game(type). Especially organization or the mobile network is becoming more and more important lately [17].

Concluding the related work, there are plenty of research approaches for the wide field of MMORPGs ranging from rather social to very technical reviews. This paper fills the gap of a nonexisting second generation MMORPG review, giving a brief overview about the MMORPG scene and the game mechanisms. Furthermore it tries to understand the underlying network structure, rendering schemes and bandwidth consumptions. Thus the data gained can be used for further game improvements, such as creating a middleware that can host MMORPG-related basic functions without depending on a certain network types.

6. CONCLUSIONS

Summarizing the game analysis of Everquest2 as an example of a second generation MMORPG, several different important aspects could be observed. The game itself already has several working strategies to cope with high latency. This means, as expected and already shown at other MMORPGs, the game can still run smooth with very high latency (1250ms). Especially in intense combat situations the game manages to establish a smooth virtual environment. However the very limited number of test clients does not provide enough information for generalized statements. Thus the results should be validated with a bigger test group (a perfect target that does not charge money for even big numbers of test clients would be an open source MMORPG).

We also observed the rendering mechanisms. The local rendering via dumb-client model certainly has its advantages; with multiple participants (like 100+) it is still possible to interact with your local copy of the environment, without waiting for the acknowledgements of each other player. Taking this model into other network structures though will certainly cause new problems. Furthermore, when thinking about a single server architecture or a distributed network such local rendering will cause problems. If no server is present to cope with cheating attempts, then movement capabilities can be abused (especially to pass walls or whole areas). Thus depending on the network type there will still be the need of an institution (such as a group of random clients or the server itself) to judge whether or not the client movement is legal. Else an instanced dungeon, which takes at least one hour to clear and fight to the boss-monster might suddenly get restructured on the local copy, and the players reach their goal within a couple of seconds. Another problem in distributed areas might be that some clients might not receive all important environment information (or at least receive them massively delayed).

Another important conclusion is that the packet size remained the same, even in crowded situations, only the number of packages changed. That means Everquest2 has a low-end client data rate, which is an important aspect in mobile network structures as well. As mentioned, the data rate in mobile environments will not increase in the same speed as the equipment of the mobile devices does. Thus such low client data rates would fit perfect into a mobile environment. Never the less there is still a lot of server-side game traffic that was not monitored. Database queries for items, characters, monsters, abilities, player inventory and so on create a high amount of bandwidth. We mentioned that one step of creating completely mobile MMORPGs would be a deeper understanding of peer to peer models for a huge number of clients [14]. Another approach would be the monitoring of current first generation mobile MMORPGs. Both of them are planed for the future.

Further research will focus on the creation of a middleware platform to support general game functions for different networks, such as peer to peer or server-client architectures. With such a middleware, the game design itself should be network independent and even with changing network conditions the game world itself should be untouched. Furthermore, we will evaluate the possibilities of creating a MMORPG in a mobile content.

7. REFERENCES

[1] Internet resource: “The first released MMOG ever was Meridian 59” – a fantasy setting: for more information refer to http://www.meridian59.com/, (Aug. 2005).

[2] Internet resource: Detailed market percentages are available at: http://www.game-research.com/. For a complete overview of the MMORPG community refer to http://www.mmorpg.com/ , (Aug. 2005).

[3] For detailed information about the content refer to http://www.everquest2.com.

[4] Further information about NISTNET is available at http://www-x.antd.nist.gov/nistnet/ . Status of 05-10-2005.

[5] Y.W Bernier. *Latency Compensation Methods in Client/Server In-Game Protocol Design and Optimization*. In Proceeding of the Game Development Conf., Feb. 2001.

[6] Jouni Smed, Timo Kaukoranta, Harri Hakonen. *Aspects of Networking in Multiplayer Games*. Turku Centre for Computer Science, 2002.

[7] Nicholas Yee. *Facets – 5 motivation factors for Why People Play MMORPG's*. Internet resource March 2002.

[8] Lisbeth Klastrup. *Interaction forms, agents and tellable events in Everquest*. Melbourne DAC 2003.

[9] Nicholas Ducheneaut, Robert J. Moore, Eric Nickell. *Designing for Sociability in Massive Multiplayer Games: an Examination of the Third Places of SWG*. Other Players conference, Denmark 6-8 December 2004.

[10] Cory Ondrejka. *Escaping the Gilded Cage - User Created Content and Building the Metaverse*. February 2002.

[11] Richard A. Bartle. *Pitfalls of Virtual Property*. The Themis Group, April 2004.

[12] Natha Sheldon, Eric Girard, Seth Borg, Mark Claypool, Emmanuel Agu. *The Effect of Latency on User Performance in Warcraft III*. Worcester Polytechnic Institute, May 2003.

[13] Chris Gauthier Dickey, Daniel Zappala, Virginia Lo. *A Distributed Architecture for Massive Multiplayer Online Games*. University of Oregon, April 2004.

[14] Yoshihiro Kawahara, Hiroyuki Morikawa, Tomonori Aoyama. *A Peer-to-peer message exchange scheme for large scale networked virtual environments.*The university of Tokyo, 2004.

[15] Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, Mark Claypool. *The effects of loss and latency on user performance in unreal tournament 2003®*. Netgames 2004.

[16] James Nichols and Mark Claypool. *The Effects of Latency on Online Madden NFL Football*. In Proceedings of the 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV).

[17] Kieran Mansley, David Scott, Alastair Tse, Anil Madhavapeddy. *Feedback, latency, accuracy: exploring tradeoffs in location-aware gaming.* Netgames 2004.