Analyzing the Network Traffic Requirements of Multiplayer Online Games

Enrique Asensio, Juan M. Orduña, Pedro Morillo
Departamento de Informática
Universidad de Valencia
Avda. Vicent Andrés Estellés, s/n
46100 - Burjassot (Valencia), Spain
Juan.Orduna@uv.es

Abstract

In recent years, Distributed Virtual Environments (DVEs) have become one of the major network applications, mainly due to the enormous popularity of multiplayer online games in the entertainment industry. Although the workload generated by avatars in a DVE system has already been characterized, the actual network traffic requirements of multiplayer online games are usually limited (hidden) by the available network bandwidth.

In this paper, we propose the measurement of the network traffic requirements of the most popular MOGs by monitoring the network traffic generated by different game tournaments in a LAN Party. The network infrastructure was explicitely designed and implemented for that event by a Network Service Provider, achieving a sustained bandwidth of 100 Mbps for each network interface. Therefore, the potential bandwidth bottleneck was moved from the network to another element of the system or application. The results show that the aggregated bandwidth required by these applications is not higher than 1600 Kbps. Also, the results show identical variations in the network traffic sent to some of the clients by the game server. These results can be used as a basis for an efficient design of MOGs infrastructure.

1. Introduction

The enormous popularity that multiplayer online games have acquired nowadays has allowed a huge expansion of Distributed Virtual Environments (DVEs). These highly interactive systems simulate a 3-D virtual world where multiple users share the same scenario. Each user is represented in the shared virtual environment by an entity called *avatar*, whose state is controlled by the user through the client computer. The system renders the images of the virtual world that each user would see if he was located at that point in the virtual environment. Hundreds and even thousands of client computers can be simultaneously connected to the DVE system through different networks, and even through the Internet. Although DVE systems are currently used in many different applications such as civil and military distributed training [15], collaborative design [25, 18, 13] or e-learning [4], the most extended example of DVE systems are commercial, multiplayer online game (MOG) environments [30, 6, 11, 22, 2].

Different architectures have been proposed in order to efficiently support multiplayer online games: centralized-server architectures [28, 22], networked server architectures [12, 3, 19, 20] and peer-to-peer architectures [17, 10, 7, 24, 23]. Figure 1 shows an example of a centralized-server architecture. In this example, the virtual world is two-dimensional and avatars are represented as dots. In DVEs based on a centralized-server architecture, there is a single server and all the client computers are connected to this server. The server is in charge of managing the entire virtual world. As a result, it becomes a potential bottleneck as the number of avatars in the system increases. In fact, the DVEs based on this architectures are the ones that support the lowest number of clients.

Figure 2 shows an example of a networked-server architecture. In this scheme there are several servers and each client computer is exclusively connected to one of these servers. This scheme is more distributed than the client-server scheme. Since there are several servers, it considerably improves the scalability, flexibility and robustness regarding to the client-server scheme. However, it requires a load balancing scheme that assigns clients to servers in an efficient way.

Figure 3 shows an example of a peer-to-peer architecture. In this scheme, each client computer is also a server. This scheme provides the highest level of load distribution. Although the first DVEs were based on centralized architectures, during the last few years architectures based on networked servers have been the major de-facto standard for DVE systems [12, 3, 8]. However, each new

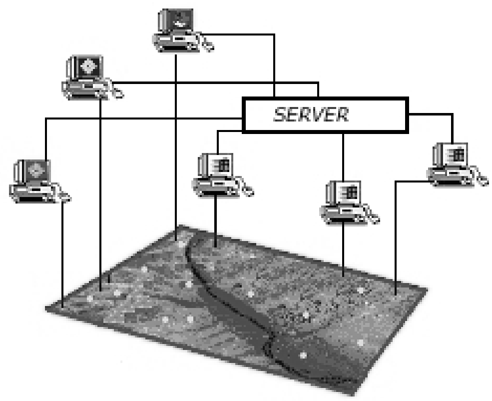

Figure 1. Example of a client-server architecture

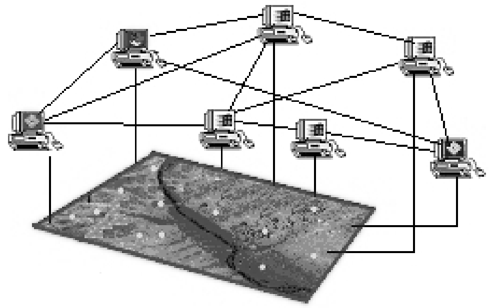

Figure 3. Example of a peer-to-peer architecture

avatar in a DVE system represents an increase not only in the computational requirements of the application, but also in the amount of network traffic [19]. Due to this increase, networked-server architectures seem not to properly scale with the number of clients, particularly for the case of MMOGs [1], due to the high degree of interactivity shown by these applications. As a result, Peer-to-Peer architectures have been proposed for massively multi-player online games[17, 16, 7].

Regardless of the underlying system architecture, DVEs are network-based applications, and therefore the network traffic requirements should be measured in order to design scalable systems that efficiently support Multiplayer Online Games. The reason is that on the one hand MOGs are the most popular real-time applications, making up around half the top 25 types of non traditional traffic for some Internet links [14]. On the other hand, MOGs have special network requirements, due to their real-time characteristics [26]. Despite the video game market is currently divided

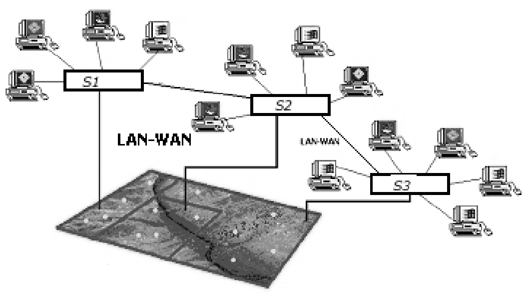

Figure 2. Example of a networked-server architecture

in more than ten types of sub genres (fighting games, role-playing games, simulation games, etc.) [29], first person shooter games (FPS) have focused the attention of the research community. The reason for such interest is due to the fact that FPS games impose the most restrictive requirements on both network traffic and QoS, because of their fast-paced nature [27]. In FPS games, the game is visualized in a first person perspective by users, who are located in a 3D virtual scene (called map) holding a gun or similar weapon [29].

In this paper, we propose the analysis of the IP traces generated by different MOGs during different game tournaments that took place within a LAN Party [21]. A LAN Party is a great meeting of Internet fans. During some days, they meet with their computers in a big place and they share their passion for computers and Internet. Some of the main events in the LAN Party consists of Multiplayer Online Game Tournaments, where different teams play among them. The network infrastructure for the LAN Party was explicitely designed and implemented for that event by a Network Service Provider (Telefónica, S. A.), achieving a sustained bandwidth of 100 Mbps for each network interface. Therefore, the potential bandwidth bottleneck was moved from the network to another element of the games. We traced all the traffic exchanged with the game servers, in order to measure the traffic requirements of Multiplayer Online Games. The results show that the aggregated bandwidth required by these applications is not higher than 1600 Kbps. The results also show that there are two kinds of games. One of them shows an inbound traffic bandwidth lower than the outbound traffic bandwidth. The other kind of game shows an inbound traffic bandwidth whose average values are very similar to the output traffic bandwidth values. Finally, the results show identical variations in the network traffic sent to some of the clients by the game server. These results can be used as a basis for an efficient design of MOGs infrastructure. The proposed characterization is essential to develop efficient optimization techniques for these systems and also to design methods for providing Quality of Service. Last years several proposals have been made about characterizing the high-level abstraction of networked games [26, 27, 9]. These studies have shown that multiplayer online games have distinctive features in terms of the effects of latency on users and user behavior. Nevertheless, to the best of our knowledge no proposal has been still made about low-level workload characterization in networked games.

The rest of the paper is organized as follows: section 2 describes the characterization setup for obtaining real traces from the different game tournaments. Next, section 3 shows the network requirements shown by the obtained traces. Finally, section 4 shows some conclusions and future work to be done.

2. Evaluation Setup

During the Murcia LAN Party 2004 event [21], the game servers were put in a subnet different from the one used by the users. The different subnets were linked by a switch. We monitored the switch where all the game servers were connected, obtaining all the traffic (IP headers) generated by the game tournaments.

Concretely, the MOGs played in the different tournaments were the following ones: Quake III Arena, Counter Strike, Unreal Tournament 2004, Warcraft3, The Frozen Throne, Need for Speed Underground, and FIFA 2004. Not all the games were active at the same time, although some of them overlapped during certain periods of time.

In order to capture the traffic, we used the Ethereal application [5]. This application allows to export in text format the acquired data and use shell scripts for processing the traces as desired. We captured traces from the noon of October the twenty third to the noon of October the twenty fourth, with a total period of around twenty four hours. We have filtered the trace file, selecting those traces belonging to different game protocols, and we have identified the servers (both the IP addresses and the server ports (a single physical server can host different game servers) and the clients involved in each game. At that point we have computed the bandwidth required in each game, as well as other relevant network parameters.

3. Evaluation Results

In this section, we analyze the network requirements shown by the multiplayer online games. The main parameter to be analyzed is the network bandwidth required by both the servers and the clients. Although we have analyzed different games and different servers, due to space limitations we show here only the results for two representative games and a representative server. The rest of the results were very similar to those shown here.

3.1. Server Bandwidth Requirements

Figure 4 shows the traffic sent and received by a server of the Counter Strike tournaments. In this figure, the Y-axis shows the bandwidth (measured in Kbps) computed for the server. The X-axis shows the time in units of ten seconds. Figure 4 shows three different plots, one for the outbound traffic bandwidth (packets going from the server to the clients), one for the inbound traffic bandwidth (packets going from the clients to the server) and another for the aggregated traffic (outbound and inbound traffic). Each peak in the figure can be considered as a different match of the game tournaments. Figure 4 shows around each traffic peak the number of clients participating in that match. In this way, we can compute the average bandwidth required per client.

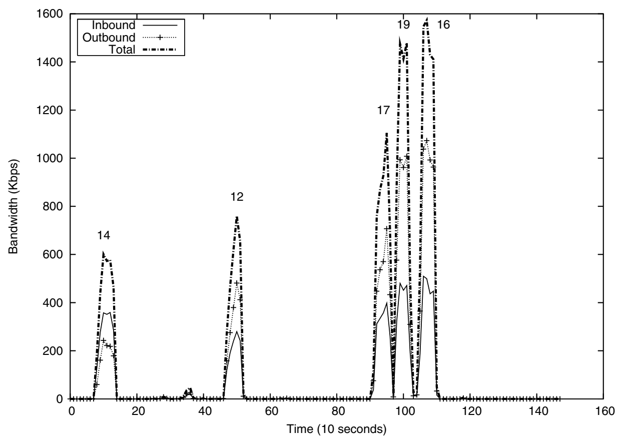

Figure 4. Bandwidth requirements for a CStrike server

Figure 4 shows that for the Counter Strike game the maximum aggregated bandwidth required is 1600 Kbps (the match played from around 5'00 a.m. to around 6'00 a.m. on October the 24th.), while the minimum is 600 Kbps. The number of players in the former match is sixteen, while the number of players in the latter one is fourteen. Another general feature obtained for this game (Figure 4 only shows one exception) is that the inbound traffic (received traffic) bandwidth is significantly lower than the outbound traffic bandwidth.

In order to summarize the results obtained for all the Counter Strike tournaments, Table 1 shows the average values for all the Counter Strike matches. These average results correspond to eight different tournaments, and they show not only the required bandwidth but also the average bandwidth required per client computer.

Aggr. Bandwidth	1600 Kbps
Inbound B.	200 - 450 Kbps
Outbound B.	400 - 1100 Kbps
Inbound B. per client	30 Kbps
Outbound B. per client	74 Kbps

Table 1. Summary of results for Counter Strike tournaments.

Figure 5 shows the traffic sent and received by a server of the Quake Arena tournaments. These results are quite different from the ones shown in Figure 4. On the one hand, the aggregated bandwidth required by this game is significantly lower than the one required for the Counter Strike tournaments. Effectively, Figure 5 shows that the aggregated bandwidth is not greater than 300 Kbps, while for the case of the Counter Strike game it reached 1600 Kbps. On the other hand, Figure 5 shows that the inbound traffic (received traffic) bandwidth is significantly higher than the outbound traffic bandwidth. This is the opposite behavior to the one shown by the Counter Strike game, where the traffic sent by the clients to the server required a lower bandwidth.

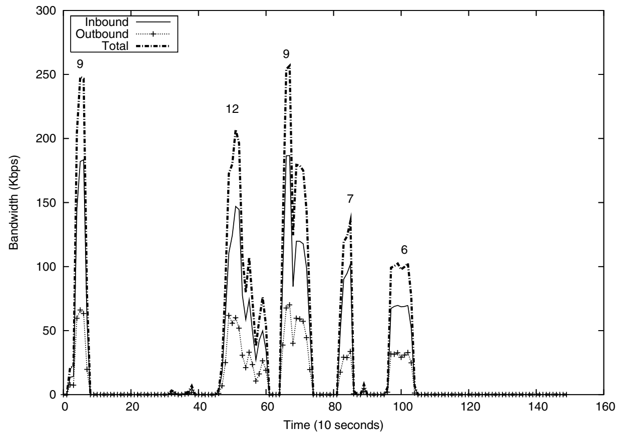

Figure 5. Bandwidth requirements for a SQuake server

Finally, Table 2 shows the average values for all the Quake matches. This table show that, as an average, the traffic exchange that takes place in this game is even. Effectively, the inbound and the outbound traffic bandwidth (both the overall and per client) show equal average values.

Aggr. Bandwidth	1000 Kbps
Inbound B.	550 Kbps
Outbound B.	550 Kbps
Inbound B. per client	25 Kbps
Outbound B. per client	25 Kbps

Table 2. Summary of results for Quake tournaments.

3.2. Client Bandwidth Requirements

We have also measured the total aggregated bandwidth required by the different client computers participating in the tournaments. Figure 6 shows the aggregated bandwidth required by ten of the clients (for the sake of clearness we have filtered six of the plots) during a single match. This figure shows on the X-axis the time, measured in tens of seconds. On the Y-axis, the figure shows the total aggregated traffic bandwidth consumed by the clients, measured in Kbps.

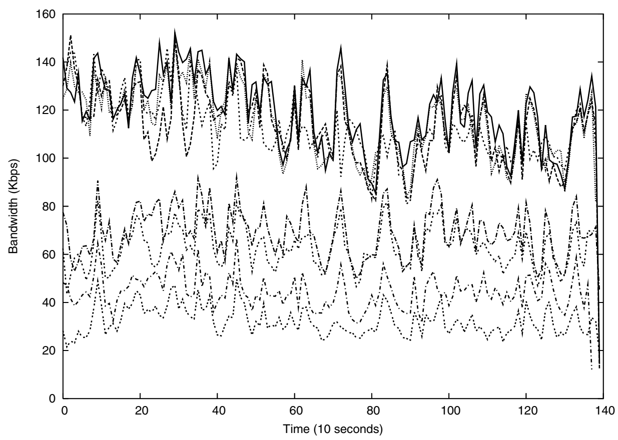

Figure 6. Bandwidth requirements for different clients playing Counter Strike

Figure 6 shows that all the plots have a sawtooth shape with an almost flat slope. That is, all the clients require different bandwidth at different time instants, but the long term bandwidth required is quite stable. Figure 6 also shows that the considered clients can be divided into two groups. The five clients that require a higher bandwidth shows almost identical requirements, indicating that perhaps all of them belong to the same team. The required bandwidth for these clients is around 120 Kbps, while for the rest of clients the required bandwidth ranges from 20 Kbps to 100 Kbps. It is also worth mention that the instantaneous peaks suffered by all the plots are produced at the same time for all the clients. This behavior indicates that the bandwidth required by a single client can be related to the bandwidth required by the rest of clients. In order to find ti which extend this assumption is true, we have studied separately the inbound that the outbound traffic bandwidth requirements.

Figure 7 shows the outbound traffic bandwidth requirements for the same clients whose requirements are shown in Figure 6. This figure shows plots whose shapes do not overlap among them as they do in Figure 6. Also, the peaks in all the plots are smaller than the peaks shown in Figure 6. These results indicate that the output traffic generated by the clients significantly differs from each other, not only in the long-term required bandwidth, but also in the instantaneous variations.

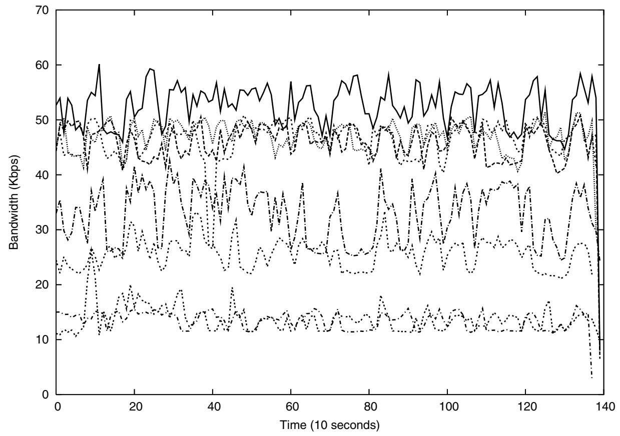

Figure 7. Outbound bandwidth requirements for the Counter Strike clients

Figure 8 shows the inbound traffic bandwidth requirements for the same clients whose requirements are shown in Figure 6. This figures shows a behavior very similar to the one shown in Figure 6. That is, the overlapping of the plots in Figure 6 is due to the traffic generated by the server. It seems that the server generated exactly the same traffic for all the clients playing in the same team.

The behavior shown in the previous figures is explained because the game clients should send short messages containing basically the location of the client within the virtual world and its movement direction. However, the server should send the clients information about the state of the system (statefull information about the virtual environment of the client).

Table 3 shows a summary of the average results for all the clients monitored in all the tournaments. This table shows not only the results for the bandwidth (inbound, outbound and aggregated bandwidth) but also the average packet sizes and the average transfers performed (measured as packets per second).

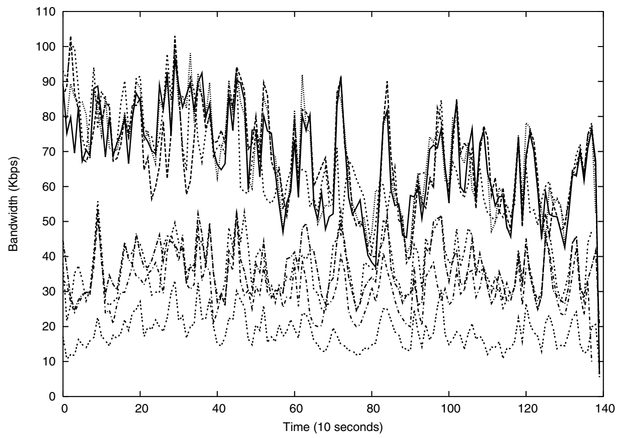

Figure 8. inbound bandwidth requirements for the Counter Strike clients

	Total	Outbound	Inbound
Bandwidth	115 Kbps	47 Kbps	69 Kbps
Pack. Size	83 bytes	61 bytes	110 bytes
Transfers	173 pack./s	94 pack./s	78 pack./s

Table 3. Summary of client results

Table 3 shows that the bandwidth required for the traffic sent by the clients is lower than the one for the traffic generated by the server. Also the size of the packets sent by the client is lower than the size of the received packets. This feature is due to the fact the the server should sent clients information about the state of the match. The different size of the packets is the reason for the higher number of transfers from the clients to the server than the number of packets in the opposed way.

4. Conclusions and Future Work

In this paper, we have proposed the measurement of the network traffic requirements of the most popular MOGs by monitoring the network traffic generated by different game tournaments in a LAN Party.

The results show that the aggregated bandwidth required by these applications is not higher than 1600 Kbps. Also, the results show that there are two kinds of games. One of them shows an inbound traffic bandwidth lower than the outbound traffic bandwidth. The other kind of game shows an inbound traffic bandwidth whose average values are very similar to the output traffic bandwidth values. Finally, the results show identical variations in the network traffic sent by the game server to the clients.

These results can be used as a basis for an efficient design of the network infrastructure for supporting MOGs. Thus, for the Counter Strike Game the network infrastructure should provide a sustained bandwidth of 47 kbps for the outbound traffic of each client computer, and a sustained bandwidth of 85 kbps for the inbound traffic. For the Quake III Arena a sustained bandwidth of 60 kbps is required for the outbound traffic of each client computer, and a bandwidth of 200 kbps is required for the inbound traffic of each client computer. On the server side, in order to provide good quality each server of the Counter Strike Game needs a network infrastructure that provides the server with an outbound channel of 85 kbps of sustained bandwidth for each potential client. Each server of the Quake III Arena needs an outbound channel of 60 kbps of sustained bandwidth for each potential client.

5. Acknowledgments

This work has been jointly supported by the Spanish MEC and European Commission FEDER funds under grants Consolider-Ingenio 2010 CSD2006-00046 and TIN2006-15516-C04-04.

References

[1] T. Alexander. *Massively Multiplayer Game Development II*. Charles River Media, 2005.

[2] Anarchy Online: : http://www.anarchy-online.com.

[3] N. Beatrice, S. Antonio, L. Rynson, and L. Frederick. A multiserver architecture for distributed virtual walkthrough. In *Proceedings of ACM VRST'02*, pages 163–170, 2002.

[4] C. Bouras, D. Fotakis, and A. Philopoulos. A distributed virtual learning centre in cyberspace. 4th International Conference on Virtual Systems and Multimedia, VSMM98, Gifu-Japan, 1998.

[5] Ethereal: : http://www.ethereal.com.

[6] Everquest: http://everquest.station.sony.com/.

[7] L. Gautier and C. Diot. Design and evaluation of mimaze, a multi-player game on the internet. In *Proceedings of IEEE Multimedia Systems Conference*, page 233, 1998.

[8] C. Greenhalgh, A. Bullock, E. Frecon, D. Llyod, and A. Steed. Making networked virtual environments work. *Presence: Teleoperators and Virtual Environments*, 10(2):142–159, 2001.

[9] T. Henderson and S. Bhatti. Modelling user behaviour in networked games. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 212–220, New York, NY, USA, 2001. ACM Press.

[10] S.-Y. Hu, J.-F. Chen, and T.-H. Chen. Von: a scalable peer-to-peer network for virtual environments. *IEEE Network*, 20(4):22–31, 2006.

[11] Lineage: http://www.lineage2.com.

[12] J. C. Lui and M. Chan. An efficient partitioning algorithm for distributed virtual environment systems. *IEEE Trans. Parallel and Distributed Systems*, 13, 2002.

[13] M. Matijasevic, K. P. Valavanis, D. Gracanin, and I. Lovrek. Application of a multi-user distributed virtual environment framework to mobile robot teleoperation over the internet. *Machine Intelligence & Robotic Control*, 1(1):11–26, 1999.

[14] S. McCreary and K. Claffy. Trends in wide area ip traffic patterns - a view from ames internet exchange. In *Proceedings of ITC Specialist Seminar*. Cooperative Association for Internet Data Analysis - CAIDA, 2000.

[15] D. Miller and J. Thorpe. Simnet: The advent of simulator networking. *IEEE TPDS*, 13, 2002.

[16] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical report, Technical Report HPL-2002-57, HP Laboratories, Palo Alto, 2002.

[17] S. Mooney and B. Games. *Battlezone: Official Strategy Guide*. BradyGame Publisher, 1998.

[18] P. Morillo, J. M. Orduña, and M. Fernández. *Encyclopedia of Networked and Virtual Organizations*, chapter The Quality of Service Issue in Virtual Environments, pages 1333–1340. Information Science Reference, 2008.

[19] P. Morillo, J. M. Orduña, M. Fernández, and J. Duato. Improving the performance of distributed virtual environment systems. *IEEE Transactions on Parallel and Distributed Systems*, 16(7):637–649, 2005.

[20] P. Morillo, S. Rueda, J. M. Orduña, and J. Duato. A latency-aware partitioning method for distributed virtual environment systems. *IEEE Transactions on Parallel and Distributed Systems*, 18(9):1215–1226, 2007.

[21] Murcia Lan Party: : http://www.murcialanparty.com/.

[22] Quake: http://www.idsoftware.com/games/quake.

[23] S. Rueda, P. Morillo, and J. M. Orduña. A saturation avoidance technique for peer-to-peer distributed virtual environments. In *Proceedings of International Conference on Cyberworlds 2007 (Cyberworlds'07), Hannover, Germany.*, pages 171–178. IEEE Computer Society Press, 2007.

[24] S. Rueda, P. Morillo, J. M. Orduña, and J. Duato. On the characterization of peer-to-peer distributed virtual environments. In *Proceedings of the IEEE Virtual Reality 2007 (IEEE-VR07), Charlotte, NC, USA.*, pages 107–114. IEEE Computer Society Press, 2007.

[25] J. Salles, R. Galli, and A. C. A. et al. mworld: A multiuser 3d virtual environment. *IEEE Computer Graphics*, 17(2), 1997.

[26] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu. The effect of latency on user performance in warcraft iii. In *Proc. of 2nd workshop on Network and system support for games (NetGames'03)*, pages 3–14. ACM Press New York, NY, USA, 2003.

[27] S. A. Tan, W. Lau, and A. Loh. Networked game mobility model for first-person-shooter games. In *Proceedings of the 2nd workshop on Network and system support for games (NetGames'05)*, 2005.

[28] Unreal Tournament: http://www.unrealtournament.com/.

[29] M. Wolf and B. Perron. *The Video Game Theory Reader*. Routledge Publisher, 2003.

[30] World of warcraft: http://www.worldofwarcraft.com.