Traffic Analysis Beyond This World: the Case of Second Life

Stênio Fernandes
Centro Federal de Educação Tecnológica de Alagoas
Maceió, AL, Brazil
stenio@gprt.ufpe.br

Rafael Antonello, Josilene Moreira, Djamel Sadok
Universidade Federal de Pernambuco
Recife, PE, Brazil
jamel@gprt.ufpe.br

Carlos Kamienski
Universidade Federal do ABC
Santo André, SP, Brazil
cak@gprt.ufpe.br

ABSTRACT
Virtual Worlds (VW), such as Massive Multiplayer Online Social Games, have been gaining increasing attention in the last few years, mainly due to the new way users interact with them. However, little effort has been devoted to understand their traffic profile and the implications to the traffic management area. With the current growing rate of VWs' usage, their traffic demand could eventually impose a significant burden on the operation of a typical Internet Service Provider (ISP) network. In this paper, we seek to understand the traffic behavior of an increasingly popular VW application, namely Second Life, from both the connection and network level perspectives. We also show results of a traffic analysis of a Second Life client, when an avatar performs different actions in the virtual world, at different places and under different network conditions. Our results show that Second Life makes intensive use of network resources (mostly bandwidth), since the capacity needed for having a full second life experience (listening to live music) may reach 700 Kbps.

Categories and Subject Descriptors
C.2.5 [**Local and Wide-Area Networks**]: Internet; H.4.3 [**Information Systems Applications**]: Communications Applications; K.8.0 [**Personal Computing**]: General - *Games*

General Terms
Measurement, Virtual Worlds

Keywords
MMORPG, Second Life

1. INTRODUCTION
A Virtual World (VW) is an interactive simulated environment accessed by multiple users, represented by avatars, through an online interface [1]. Virtual worlds (or digital worlds) provide new levels of socialization, where users can experience sensations and interact with others in a similar way to real life. New forms of expressions for human behavior, fun and amusement, and most importantly, they can conduct business offered by such environments. Furthermore, there is the advantage of preserving privacy and anonymity. Second Life (SL) [2] is a popular VW application that has been gaining a lot of attention from the news media since last year and is also one that keeps growing in the number in terms of financial dealings and users (also called residents).

Virtual worlds (like SL) offer their residents the opportunities for building new places, often comprised of terrain, buildings and objects where users seek services just like in real life. They differ from traditional online games, where players have a particular goal to achieve and are limited to interact with the preexistent environment. Second Life provides a scripting language (called LSL) that allows residents to add objects with additional features to the world.

These features make Virtual Worlds very exciting for designers, programmers and users, but raises concerns for transport and content providers who have to provide adequate quality of service levels to guaranty a good interaction experience. In addition to the traffic sent from the virtual world's server to the clients, programmers may trigger the streaming and download of real-time of other types of traffic coming from different servers from all over the Internet cyber space. In Second Life, many places allow users to activate music streams, coming from servers maintained by independent owners and sites. Such capability has the potential to generate many different traffic profiles (e.g., voice, music, video, and data) from a single virtual world, causing considerable impact on a typical ISP network. Nonetheless, little effort has been devoted to understand Second Life's traffic profile and the implications to the traffic management area.

In this paper, we take a first step to understanding the traffic behavior of Second Life, from both the connection and network level perspectives. We show results of a detailed traffic analysis of a Second Life client, while an avatar performs different actions in the virtual world (standing still, walking and flying), at different places (both popular and unpopular), using different access network technologies (ADSL and university LAN) and with or without external traffic sources (e.g., while listening to music or not). We collected information at different weekdays and hours, summarizing more than 100 hours of Second Life usage with different avatars. However, due to the space constraints, only the most significant findings are presented.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
NOSSDAV'07, Urbana, Illinois USA.
Copyright 2007 ACM 978-1-59593-746-9/06/2007…$5.00.

Our preliminary results show that Second Life makes intensive use of network resources (mostly bandwidth). The traffic bitrate arriving at the client with a full second life experience (flying with live music) reached sustained levels of about 500 Kbps and peaks of up to 700 Kbps. In some places, music accounted for about 200 Kbps (probably encoded at 192 Kbps), which comes to emphasize our argument that external data sources may generate unpredictable and potentially harmful traffic patterns.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 presents the architecture of Second Life, from a connection and client-server level point of view. Section 4 exposes the methodology of our work and section 5 the results we obtained. Finally, section 6 draws some conclusions and presents topics for future work.

2. RELATED WORK

With the current growing rate of Massively Multiplayer Virtual World (MMVW), specifically of Massively Multiplayer Online Role Playing Game (MMORPG), such as Everquest and World of Warcraft, networking researchers have been struggling to understand their network requirements through the analysis of their traffic profiles [8][9][10][11]. An in-depth knowledge of their traffic behavior will certainly assist Internet Service Providers (ISP) to provision and better design their network infrastructure. In [8], Kim et al present traffic measurements of a popular MMORPG (Lineage II) and provide its traffic characterization from the server side point of view. In [9], Feng et al analyzed the network traffic of several popular on-line games with the focus on First-Person Shooting (FPS) game traffic, namely Counter-Strike. Claypool and Claypool [10] evaluated the effects of Internet latency on online games. They argued that their results are useful for game designers (e.g., by applying different latency compensation techniques), network designers (e.g., by creating infrastructures providing quality of service (QoS) for interactive applications) and game players (e.g., by providing knowledge for QoS purchases). We observe that most MMORPG and FPS described in the literature have low bitrate requirements, due to its intrinsic characteristics of sending frequent but small packets, mainly for synchronization issues. Although the aggregate traffic at the server could eventually impose a high burden for ISPs, from the client side perspective, downstream traffic are typically far below the capacities of typical broadband connections. Indeed, most of these environments allow dial-up users to have a fair playing experience.

However, a slightly different type of MMORPG, called Massively Multiplayer Online Social Game (MMOSG), such as Second Life (SL), has recently become hype and, to the best of our knowledge, its communication patterns are still unknown. Our work reveals that the bandwidth requirements for SL are far beyond from those for popular MMORPG and FPS games. As of April 2007, SL has more than 5.5 million registered users and an average of 30.000 simultaneous on-line users[1]. With the recent widespread deployment of broadband access technologies, the increase of the number simultaneous online users is putting pressure on both SL servers and ISPs network infrastructure and demanding an in-depth traffic analysis of the SL's network behavior.

3. SL ARCHITECTURE

This section presents the roles of different servers in the SL architecture, their interactions between themselves and the way they exchange data with clients.

3.1 Servers

SL is based on an asymmetric client-server architecture, where each server is dedicated to a particular task such as login, instant messaging handling, or region simulation. The SL world consists of a great deal of interconnected and uniquely-named simulators (*sim*). Each *sim* is a process responsible for processing a 256x256 meter region, and communicates only with its four nearest neighbors, thus avoiding the transactional scaling problem as the world becomes really large [5]. A grid is a graphical division of units representing the servers which run SL, wherein one server supports one *sim* (although multiple *sims* may be, and actually are, executed on the same physical server). The user is known as viewer. As the viewer moves through the world it is handed off from one sim to another [3]. Figure 1 depicts the SL architecture, including the current known servers.

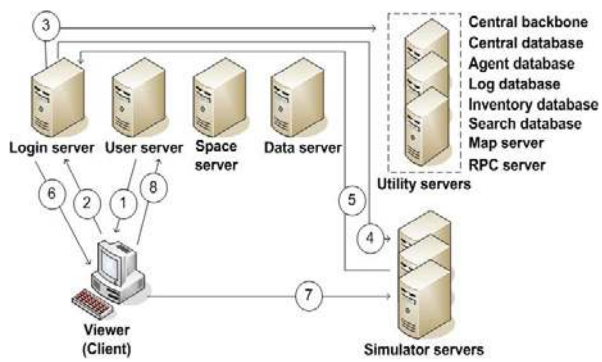

Figure 1 – SL Architecture

Login server: Server *login.agni.lindenlab.com* is responsible for the authentication of avatar names and passwords. It uses XML-RPC communication encapsulated in a TLS 1.0 (https) stream. Also, it determines the region where to connect the user, based on the last location, home location or a specific region chosen by the user at login time.

User server: Server *userserver.agni.lindenlab.com* handles instant message sessions, particularly for groups.

Space server: This server handles message routing based on grid locations. The *sims* register themselves with the space server and request their list of neighbors.

Data server: It handles connections to the SL databases: Central, Log, Inventory and Search.

Simulator server: This is the primary SL server process, responsible for storing object state, land parcel state, and terrain height-map state. They perform visibility

[1]SL Economic Statistics: http://www.secondlife.com/whatis/economy_stats.php

computations on objects and land and transmit these data to the viewers. Physics simulation is handled by the *Havok physics library* [4]. Simulators communicate with their neighbors via UDP connections.

Other servers: Central Database (CDB) contains a list of who owns what, e.g. used for billing. Agent Database keeps track of the mapping between metadata and item id (UUID), which is a globally unique identifier (128-bit number). Inventory database contains information about user's assets, which is a data resource such as an image, sound, script and object. Assets can be downloaded to the viewer or uploaded into the central asset store. Search Database is a replica of the Central DB used for search. The Map server renders the overall map with OpenGL. The RPC server behaves as an API for developers to manipulate Second Life without using the viewer. It translates XML-RPC server into in-world requests and communicates with the space and CDB servers.

3.2 Authentication Flow

Figure 1 depicts the 8 steps required to establish a user connection to Second Life, which are:

1. The viewer sends a secure message checksum request to the server (port 12036).
2. The viewer sends a XML-RPC function call, over https to login. The server can suggest an optional update.
3. The Login server queries the database server for authentication credentials.
4. The Login server decides which *sim* to send the viewer to. Then, requests session start.
5. The *sim* sends a reply to the login server with the verification that user is allowed on the region.
6. The login server sends a reply to the viewer with agent id, session id, secure session id, *sim* IP, *sim* port, global location and some inventory information.
7. The viewer sends user id and session id to the *sim*, as a handshake. The simulators address range is 64.128.0.0 to 64.129.255.255, and the port range is 13001-13050.
8. The viewer sends presence information to the User server, so that instant messages among group members may be exchanged. The User server checks group rights and sim session information.

4. DATA COLLECTION METHODOLOGY

We visited different places at different hours and days, summarizing more than 100 hours of experiments using SL. Data for this paper were collected from January 24 to 29 (2007). The methodology for our experiments and the configuration are explained in the next paragraphs.

Scenarios: After visiting a large number of different places in SL, we realized that the number of avatars and objects in a place makes a significant difference in the traffic. Therefore, we considered two places using the criterion of popularity. We used "Goddess of Love" and "Menglin II" for representing a popular and an unpopular place, respectively. In order to estimate the level of attractiveness of a place (popular or unpopular), we utilized the traffic index found in the SL client. According to SL documentation, such metric is a measure of the proportion of the in-world time that other avatars have chosen to spend in a specific place.

Network Connections: we repeated the same experiments for two types of network connections, namely a 100 Mbps link (called UFPE) from our University to the Brazilian Research Network, and a residential ADSL connection of 600 Kbps (called ADSL).

Actions and external sources: Each experiment was repeated for 3 different actions of the avatar, standing still, walking and flying. We also took into account external traffic sources (e.g., media streaming servers). Therefore, we repeated the experiments with and without music.

Metrics: We evaluated a number of metrics, although not all of them are presented in this paper, due to space constraints. We use the tcpdump protocol analyzer for capturing packets while using the SL viewer. The metrics considered were throughput, packet size (including TCP, UDP and IP headers), packet inter-arrival time, traffic volume (UDP/TCP).

For each experiment, we considered only a period of 10 minutes and the metrics were computed for each 10 second interval. In other words, for each 10 minute experiment we have 60 samples of the metric being evaluated. Most results present the average of the 60 samples and the 99% confidence intervals we also computed, although the pictures do not show them.

5. PERFORMANCE EVALUATION AND RESULTS

5.1 Throughput

Figure 2 and Figure 3 present the time series for the throughput of UFPE and ADSL, with the audio stream turned on or off, and only for a given popular place. Observing the throughput collected for those scenarios, one may notice that different network connections do not significantly affect the amount of traffic generated by the application. In other words, even in a scenario with an external music source, a 600 Kbps ADSL downlink is enough for providing a good experience for the users in most cases. Figure 2 shows that the average throughput is around 300 Kbps for the ADSL connection, whereas within the university network (UFPE) it is around 400 Kbps with peaks of up to 700 Kbps.

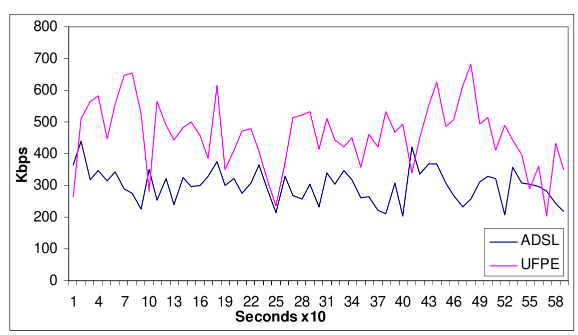

Figure 2 – Throughput (Flying, with music)

On the other hand, when the client has deactivated the audio stream, Figure 3 shows an even lower difference between ADSL and UFPE. Throughput for the former is around 180 Kbps whereas it reached 210 Kbps for the latter. From these data, it is clear that SL makes an intensive use of network resources and for enjoying a full experience, a user needs a broadband connection of at least 500 Kbps, if an external audio stream is desired. Such requirements are beyond most MMORPG and FPS games.

From now on, due to space constraints we will only consider the scenarios for the residential network connection (ADSL) and without an external data source (e.g., a media streaming).

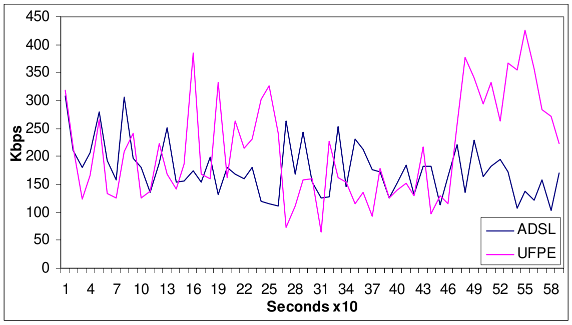

Figure 3 – Throughput (Flying, no music)

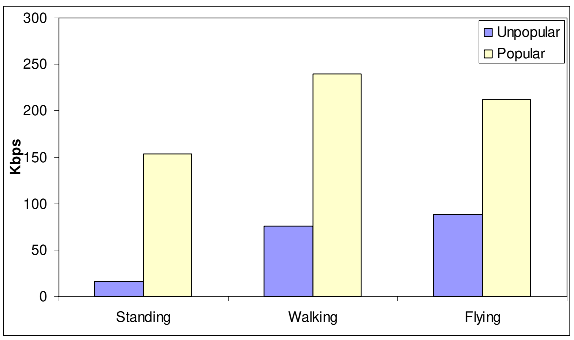

Figure 4 – Throughput (average)

Figure 4 shows the average bandwidth usage for 3 actions (standing still, walking and flying) and 2 places (popular and unpopular). Regardless of the action of the avatar, the popular place generates at least 2.5 times more traffic than the unpopular place, revealing that the bandwidth usage has a strong relationship with the particular place in SL where the avatar is.

There is also a clear correlation between the motion pattern of the avatar and the generated throughput. In Figure 5, an empirical cumulative distribution function (ECDF) shows that for an unpopular place, when the avatar is standing still, up to 97% of the time the throughput is below 20 Kbps, whereas with some form of movement it is between 60 Kbps and 110 Kbps in 87% of the time.

As far as SL places are concerned, a comparison between Figure 5 and Figure 6 points out that the throughput is much higher for the popular place (up to 400 Kbps) than for the unpopular place (up to 150 Kbps). Even when the avatar is standing still in the popular place, 93% of the time, the throughput is between 75 and 350 Kbps, which is much higher than that for the unpopular place.

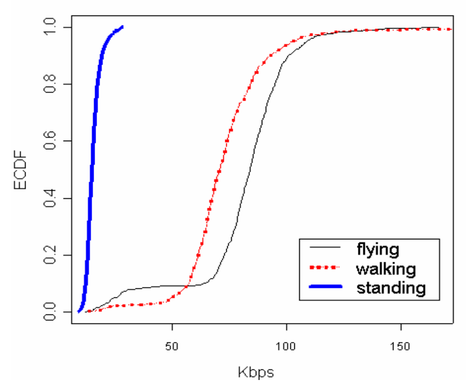

Figure 5 – Throughput distribution (unpopular place)

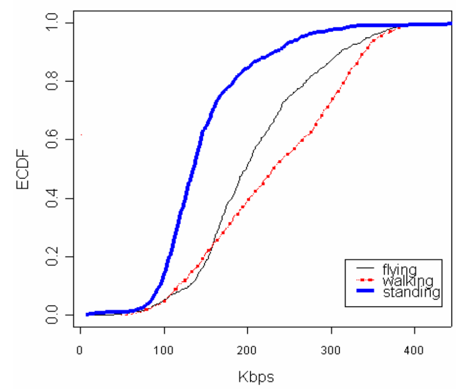

Figure 6 – Throughput distribution (popular place)

5.2 Packet size

Figure 7 presents the mean packet size for both upstream (client-to-server) and downstream (server-to-client) traffic. In addition, Figure 8 and Figure 9 show the distribution of the packet sizes for the same scenario. In this case, the mean packet size has a similar profile as that observed by Chen [11]: in general, server-to-client packets are bigger than the client-to-server ones.

Please note that the average packet sizes generated by both the server and the client at the unpopular place are similar. This is possibly due to the fact that in a unpopular (with few people and objects), the amount of information the server sends to the client is smaller due to the lack of details of the surrounding environment.

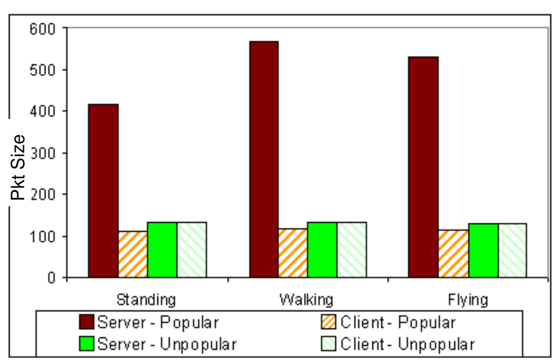

Figure 7 – Packet size (bytes)

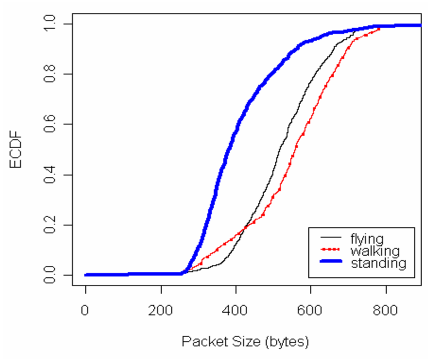

Figure 8 - Packet size distribution (popular place)

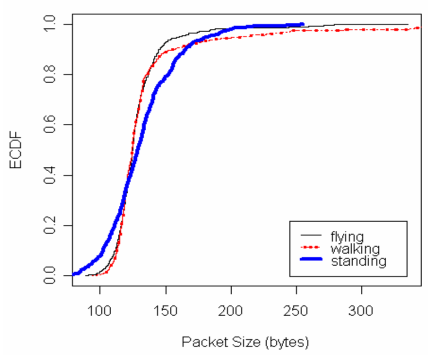

Figure 9 - Packet size distribution (unpopular place)

5.3 Traffic Volume

We found out that both TCP and UDP transport protocols are used by the application. The communication between SL clients and simulator servers use UDP, whereas the external audio streaming sources use TCP. Figure 10 shows that TCP packets are only exchanged in scenarios with music. The total volume of TCP bytes collected in our experiments is nearly the same (i.e., about 5 MB) for the 3 different actions: standing still, walking and flying. As expected, it is similar for both popular and unpopular places, which suggest the external servers may have used equivalent codecs for the audio stream. Although external traffic sources may generate unpredictable traffic patterns, as they are not controlled by SL, during our experiments they generated a steady sending rate (around 192Kbps).

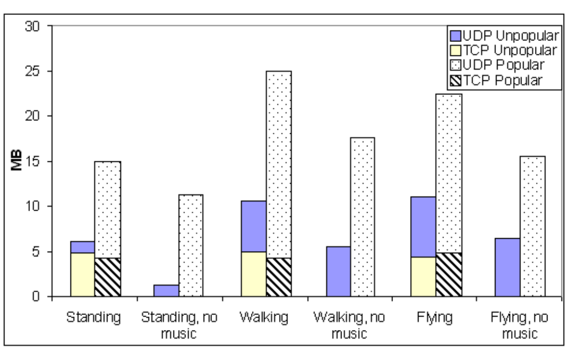

Figure 10 – Traffic volume for TCP and UDP

5.4 Packet Interarrival Times

The average packet inter-arrival times (ADSL) are presented in Figure 11. It may be observed that the interarrival time is shorter for more complex environments (i.e. popular places) and when the avatar is moving faster. Obviously, this happens because either information concerning location and objects (including obstacles) sent to the viewer need to be updated more frequently or more information is sent in each update. In a simple environment the number of objects and avatars is smaller and consequently fewer packets have to be sent per unit of time. Figure 12 and Figure 13 corroborate with our findings, since they scrutinize the packet size and interarrival times correlation.

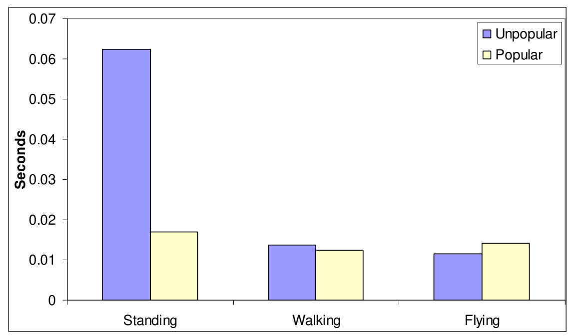

Figure 11 – Packet Interarrival time (average)

Figure 12 shows that when the avatar is standing still in an unpopular place, the client sends small packets (50 to 150 bytes) to the server, whereas the server updates the client with slightly larger packets (100 to 250 bytes), mostly below 200ms of interarrival time. On the other hand, a closer look at Figure 13 reveals that the profile for the client-to-server communications remains almost the same, whereas the server definitely increase the size of the packet payload, which we believe to contain additional information about other avatars and details of the surrounding objects. Although we do not show here, the motion pattern of the avatar also has a strong influence on both inter-arrival times and packet sizes from server to client. The client sends every movement to the simulator that in turn sends back updated information of objects, textures, other avatars, 3D viewpoint and so on.

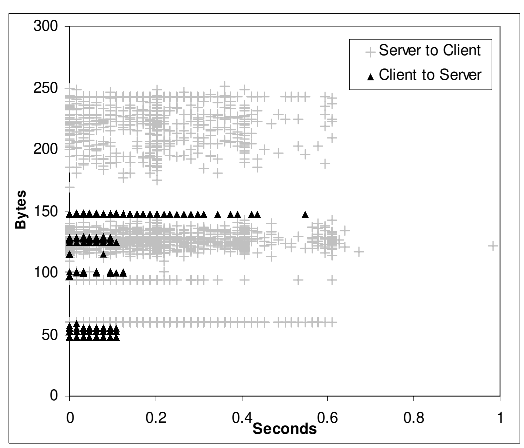

Figure 12 - Packet Size x Interarrival Time (unpopular, stand still, no music)

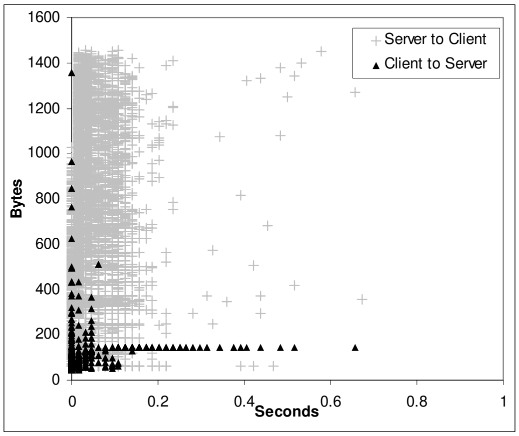

Figure 13 - Packet size x Interarrival time (Popular, standing still, no music)

6. Discussion

By filling a niche that is of interest of the Internet research community, this paper reveals that the traffic profile from the SL client's point of view, can generate diverse network patterns. This is mainly due to the fact that within SL, users are allowed to customize the virtual world, by building an assortment of objects with distinct network requirements. Each object may have unique characteristics that require additional bandwidth in order to be rendered properly by the SL client. In addition, users can attach external traffic sources within their land boundaries, such as live streaming radio. Traditional MMPORG games differ from SL given that most of them have fixed scenarios and objects, and are not customizable. Such characteristics allow a more steady and predictable traffic behavior between client and servers.

By the analysis of some selected metrics, collected at the client over a long period of time, our work provides a first picture of how developers, designers and researchers in both networking and virtual environments fields can improve the performance of their systems or networks. For instance, by the knowledge of traffic patterns of the SL clients, a local ISP could monitor and forecast the aggregate bandwidth requirement to plan a link capacity upgrade. Alternatively, ISP could apply traffic shaping and policing mechanisms.

Although we did not focus on the comparison of SL traffic patterns with other 3D online games, it is clear that SL requires more stringent network parameters. In fact, most games of different types (e.g., FPS, RPG, and Social) share common behavior with SL, such as position update messages between client and servers. However, in most games the 3D scenarios are built within the client software, which alleviate the exchange of information about the surrounding objects near the avatar. In this case, clients and servers only need to exchange information about avatars' position. For instance, a study of a MMORPG game [11] concluded that the average bandwidth required per client is small (e.g., around 7 kbps), whereas for Counter Strike it is about 40 Kbps [9]. SL requires bandwidth around 400 Kbps for the popular places and up to 150 kbps for unpopular places.

7. Conclusions

Second Life (SL) is a virtual world that has been increasingly gained the attention from the news media and Internet community since the last year. This paper takes a first step on understanding the SL architecture and profiling the traffic generated by its servers and clients. Our results show that SL makes intensive use of network resources, and for enjoying a full experience, a user needs a broadband connection of at least 500 Kbps (with external audio stream). Without a live audio stream, the average server-to-client throughput is about 200 Kbps for a popular place and below 100 Kbps for an unpopular place. This is within the possibilities of most broadband residential users and currently does not represent a threat for ISPs network planning. However, since SL permits developers to attach different external traffic sources, some SL places may generate higher and unpredictable traffic patterns in the future. Furthermore, the increase of the number simultaneous online users is a potential concern for network management

As future work, we intend to derive traffic models for SL and comparing it with another similar virtual world. We envisage presenting a breakdown of the SL traffic by showing how many percent of traffic comes from chat messages, avatar position updates, transmission of 3D objects. Since SL client went open source and its messages are not encrypted, it opens up a number of possibilities for profiling SL traffic within SL client source code.

8. References

[1] Book, B., "What is a Virtual World", Virtual World Review, 2006, http://www.virtualworldsreview.com
[2] Rymaszewski, M. et al., "Second Life: The Official Guide", Sybex, 2006.
[3] Second Life, Inc., "Open Source Portal", 2007, https://wiki.secondlife.com/wiki/Open_Source_Portal.
[4] Havok Inc, "Havok Physics", 2006, www.havok.com.
[5] Rosedale, P. & Ondrejka, C., "Enabling Player-Created Online Worlds with Grid Computing and Streaming", Gamasutra, Sept. 2003, http://www.gamasutra.com
[6] Ondrejka, C., "Changing Realities User Creation, Communication, and Innovation in Digital Worlds", 2005, http://lindenlab.com/whitepapers/
[7] Ondrejka, C., "Escaping the Guilded Cage User Created Content and Building the Metaverse" http://lindenlab.com/whitepapers/
[8] Kim, J., et al, "Traffic characteristics of a massively multi-player online role playing game", 4th ACM SIGCOMM Workshop NetGames '05, October 2005.
[9] Feng, W., Chang, F., Feng, W., and Walpole, J. 2005. "A traffic characterization of popular on-line games", IEEE/ACM Trans. Netw. (13) 3, June 2005.
[10] Claypool, M. and Claypool, K., "Latency and Player Actions in Online Games", Comm. ACM, (49) 11, pp 40-45, November 2006.
[11] Chen, K.-T. et al, "Game Traffic Analysis: An MMORPG Perspective", NOSSDAV'05, June 2005.
[12] Zander, S. et al, "Dissecting Server-Discovery Traffic Patterns Generated By Multiplayer First Person Shooter Games", 4th ACM SIGCOMM Workshop NetGames '05, October 2005.