Experimental study and modelling of Networked Virtual Environment server traffic

Juan Luis Font
Dept. of Computer Tec. and Architecture,
University of Seville, Spain
Email: juanluisfont@us.es

Christian Callegari
TLC NETworks research GRouP
University of Pisa, Italy
Email: christian.callegari@iet.unipi.it

José Luis Sevillano-Ramos
Dept. of Computer Tec. and Architecture,
University of Seville, Spain
Email: jlsevillano@us.es

***Abstract*—The paradigm of virtual world environment arises as an useful tool in diverse fields such as e-Health or education, where they provide a new way of communication and interaction with end users. Networking capabilities play an important role in these systems, which motivates the study and understanding of the gaming network traffic. The present work focuses on Open Wonderland, a system that provides the basis for the development of Networked Virtual Environments with educational or health purposes. The goal of this paper is defining a testing environment and modelling the behaviour of the outgoing network traffic at the server side.**

***Index Terms*—modelling, network traffic, Open Wonderland, Networked Virtual Environment**

I. Introduction

Most multi-player on-line games rely on the virtual world paradigm. This “virtual reality” is a simulation shared and synchronized among multiple players over a data network. The massive popularization of these games in recent years has meant the emergence of a new business model in the video game industry and has attracted the interest of the research community, focused on the study and better understanding of the network requirements and user experience, as well as on the idea that other fields such as the *e-Health* may benefit from the introduction of new tools. For example, the possibility of several users interacting in a Networked Virtual Environment (NVE) has been seen as a suitable basis for the development of the so-called “persuasive systems”, systems that focus on motivating healthy lifestyle habits [1].

Open Wonderland (OWL) is an Open Source NVE, which has been previously used as basis for the development of a pervasive system over a NVE, “Virtual Valley”[1], [2]. There are several reasons that make OWL a suitable testing bench: a GPL v2 license, usage of Java technologies to ensure portability and freedom to deploy a OWL instance on a wide range of configurations and infrastructures.

The network traffic generated by Wonderland is mainly due to three sources. First, object synchronisation that allows all users to have a coherent view of the virtual world (including moving objects like avatars). Second, messages intended to support communications among users, including voice traffic (the main source of traffic) but also text messages (chat). And, finally, traffic due to the execution of applications shared among different users. The latter proves to be very difficult to model, as it depends on the particular application.

This paper aims to study and model the behaviour of the server outgoing traffic. This is a first step to understand the network behaviour and accomplish further study of the impact of the underlying network over the performance of NVEs.

The paper is structured as follows: Section II provides a brief summary of related literature, Section III defines the methodology employed in the current work as well as the mathematical and statistical tools. Section IV provides a detailed description of the test-bed and the performed testing sessions. Then, Section V shows the results extracted from the experimental data, while Section VI concludes the paper with some final remarks.

II. Previous Work

The early 2000s witnessed the proliferation of studies focused on the most prominent multi-player genres, First Person Shooter (FPS), Real-Time Strategy (RTS) and Multiplayer Online Role-Playing Game (MMORPGS). In [3] there is a complete survey about the gaming ecosystem.

Regarding the analysis and modelling of gaming network traffic, the study about FPS games performed in [4] laid the groundwork for the study of gaming traffic, proposing a methodology adapted to the specific nature of this traffic, specifically highlighting the study or packet inter-arrival time and size. Later studies such as [5] and [6] followed similar approaches and metrics: the first one focusing on Starcraft (RTS genre) and also paying attention to the autocorrelation as a defining characteristic of the traffic and the second one keeping the study of up-to-date FPS titles.

As stated above, OWL falls within the Networked Virtual Environments (NVEs), which are closer to genres like MMORPGS and RTS due to the network requirements. Literature has covered broader spectrum topics about NVEs: usage of NVEs as basis for persuasive systems ([1]), integration of NVEs with low cost sensors ([2]). On the other hand, [8] has focused on the network traffic aspects of OWL, proposing several models for the outgoing traffic of OWL clients. To the best of our knowledge there is no work that deals with server traffic modelling for NVEs like the one presented in the present paper.

III. Methodology

A. Testing gaming sessions

Most authors have chosen to base their studies on real human-driven gaming sessions [6] due to the complexity of using reliable models for simulating human player behaviour. In the testing environment defined in this work, human players have been replaced by scripts and the player behaviour has been simplified to maximise the interaction rate and its associated network traffic. Although this simplification may seem not representative of the traffic obtained during real gaming sessions, the results from [8] suggest that the impact of lower rates of activity in OWL sessions translates into greater inter departure and arrival times and therefore more heavy-tailed distributions while keeping an analogous nature to those proposed in this study.

Previous studies have distinguished between "active" and "inactive" players, focusing on the first ones considered as more representative about the normalised gaming experience to be studied and modelled [9]. In [8], it was determined that the inactivity periods of the so-called "inactive" clients translated into absence of traffic from client to server. The most basic forms of interaction that a player can perform in OWL are avatar movement and voice transmission. Each of these interactions translates into traffic exchange between clients and server. Audio traffic shows a periodic nature, while traffic associated to avatar movement is tightly linked to the events triggered by the OWL player.

All the testing sessions were performed in the same scenario, a minimalistic virtual world where each user was within the field of vision of the rest of the players. An instance of the whole virtual world with all its static elements is loaded at the beginning of the session, after that, there is no network traffic associated to such objects unless they perform any kind of special activity, which is not the case in this work.

Scripting has been used to remove the human factor in the testing gaming sessions [10] [11]. The real keystrokes have been replaced with scripts which simulate keystrokes of the cursor keys, responsible for the movement of the avatar. Thus, the need of a human player and a physical keyboard are bypassed. These OWL clients functioning without human intervention are defined as "automated clients" in this paper. To avoid any kind of correlation in their behaviour, the script determines the direction of the movement: forwards, backwards, turn right or turn left. This decision is made using a pseudo-random number generator following an uniform distribution [12]. The amount of time during which the avatar performs the selected movement is also determined using an uniform distribution. Independently of the erratic movement performed by each "automated client", they constantly generate updates about their respective position which are propagated all over the OWL system as TCP packets.

B. Network traffic study

The present work studies the server outgoing TCP traffic, focusing on its patterns, and the impact of an increasing number of concurrent users. The TCP traffic on OWL server is mostly generated by the propagation of updates that notify other clients about the changes in the virtual world. This work aims to model the behaviour of two traffic parameters: Inter Departure Time (IDT, time between consecutive packets leaving the server) and Packet Size of Outgoing traffic (PSO, size of the payloads contained into the TCP packets).

Weibull and Exponential distributions have been chosen to fit the observed Empirical Cumulative Distribution Functions (ECDF) for the data sets (data sets will be described in the following). Maximum Likelihood Estimation (MLE) was used to estimate the parameters for the best fitting distributions and λ^2 test was chosen to evaluate the goodness of fit between the empirical data and the proposed theoretical distribution with their estimated parameters. The λ^2 Discrepancy Measure quantifies the goodness of fit between a data set and a given theoretical distribution. It is based on the chi-square (χ^2) and it is independent from the number of observations and bins used during the calculation process, so it can be used for a quantitative comparison of the fitting of different data sets from a population. This metric was originally proposed in [13]. Several other works from the traffic analysis literature, [5] or [6], have also used λ^2 as discrepancy metric. The present paper uses the definition of λ^2 and its alternative, $\hat{\lambda}^2$, described in [4]. Both discrepancy metrics have been evaluated in this study, using Q-Q plots to determine their significance.

IV. Experimental Test-bed

The results shown in this work are based on experimental data obtained from a test bed running an instance of Open Wonderland. This test bed was composed of a server machine and a variable number of client computers, ranging from 1 to 10, depending on the number of players involved in each of the performed testing sessions.

The OWL instance was deployed on the hardware equipment of a Computer lab of the University of Seville, so all the machines shared the same hardware specifications. Each machine was equipped with an Intel i7-2600K CPU clocked at 3.4 GHz; 8 GB of 1333 MHz Dual Channel DDR3 RAM; GPU nVidia GForce GTX 550Ti equipped with 1024 MB of dedicated RAM; and integrated Gigabit Ethernet network cards. The underlying network was based on Gigabit Ethernet.

Despite all the machines sharing a common hardware configuration, a distinction was made between server and client software regarding their underlying operating systems. On the one hand, computers playing the role of OWL players ran under Microsoft Windows 7 64-bit with SP1. This choice makes sense in a technological context where Microsoft operating systems has over 87% of the domestic market, with a share of 53% for Windows 7 alone [14]. On the other hand, OWL started as a Sun Microsystems project with a tight relation with server technologies. In the same way that Windows is the most frequent choice on the client side, trends point to a Unix-like environment for server deploy. GNU/Linux is the rule in this context and the most convenient choice for system administration purposes. The server machine ran an Ubuntu

12.04 distribution updated with its official security patches and a Linux kernel 3.2 optimised for i686 architecture.

It must be highlighted that OWL is a project based on Java technologies, so any influence of the operating system is mitigated in some degree for the intermediate software layer comprised for those Java technologies. Both Windows clients and GNU/Linux server used Java 32-bit technology. Clients used the Oracle Java Runtime Environment (JRE) v.1.7. OWL server requires the use of a complete Java Development Kit (JDK) which also includes a JRE. The JDK selected for the server was the OpenJDK v1.7.0 32-bit implementation, which follows the Java 7 specification. The usage of an Intel hardware architecture of 64-bits (amd64) running both 32 and 64-bit software has a negligible impact over the network traffic results. The network communication in our study is agnostic regarding to the underlying CPU architecture.

V. Results

The results were obtained running 10 testing sessions using "automated clients". The number of concurrent users ranged from 1 to 10. Each session lasted 14 minutes. Network traffic captured during the first and last minute of each session was discarded to avoid patterns generated by the log in and log out process. All the traffic was captured using *libpcap*.

A. Inter Departure Time for TCP traffic

A preliminary study of the autocorrelation of IDT of TCP traffic showed certain degree of autocorrelation for sessions with only 1 or 2 concurrent users. Figure 1 contains several *autocorrelograms*. Each autocorrelogram shows the Auto Correlation Function (ACF) for a given numerical sequence. Due to space constrains, Figure 1 only contains autocorrelograms for a subset of testing sessions (1,2,4 and 8 client sessions), the remaining plots do not differ significantly but confirm the presented observations. The traffic sent to a single user presents certain autocorrelation, but this trend fade away in the 4-client session. Beyond this number of players, the IDT values can be considered as randomly distributed [15]. The OWL server does not send updates on regular basis, on the contrary, it propagates the position update packets when they are received. The alternating between big update propagation packets and small ACKs is the reason for the high degree of inverse autocorrelation observed in the 2-player case.

The TCP server traffic is mostly reactive to OWL request, without signs of periodicity nor bursty behaviour [8]. This supposes a difference in relation to other games analyses in the previous literature such as [6] and [3]. Weibull and Exponential distributions are those who best describe the outgoing TCP server traffic. The respective parameters for these distributions (λ *rate* for Exponential; and α *shape* and β *scale* for Weibull) were calculated using Maximum Likelihood Estimation (MLE). Table I contains the estimated parameters for each one of the testing sessions.

The evolution of the estimated parameters is consistent with the increase of concurrent players. An increment in the value of exponential λ means a higher frequency of packets and

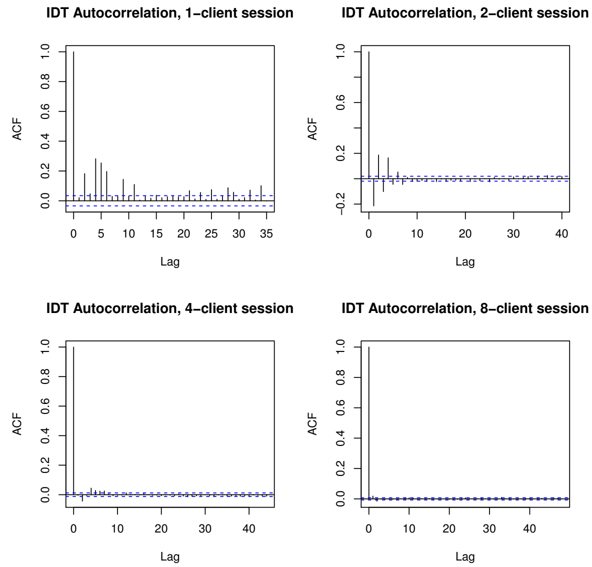

Fig. 1: Autocorrelation results for IDT

lesser IDTs. On the other hand, the decreasing trend of the Weibull scale parameter β suggests a narrower spreading, while values of the shape parameter α close to 0 indicate that small IDT values prevail over the rest. A gaming context where the number of players increases will require the server to attend more request and will mean a bandwidth increment and a reduction of IDT times between consecutive packets.

TABLE I: MLE parameters for Exponential and Weibull

	Exponential		Weibull			
N.Users	λ	$\sigma(\lambda)$	α	$\sigma(\alpha)$	β	$\sigma(\beta)$
1	4.54	0.0795	0.772	0.00955	0.18249	4.35e-03
2	14.22	0.1405	0.695	0.00576	0.05830	8.62e-04
3	23.70	0.1814	0.671	0.00422	0.03334	3.97e-04
4	35.38	0.2218	0.585	0.00299	0.01927	2.15e-04
5	47.47	0.2569	0.510	0.00223	0.01177	1.29e-04
6	61.31	0.2920	0.485	0.00185	0.00826	8.26e-05
7	80.00	0.3335	0.448	0.00147	0.00513	4.59e-05
8	99.44	0.3717	0.429	0.00124	0.00366	2.78e-05
9	122.46	0.4124	0.420	0.00108	0.00284	1.69e-05
10	126.06	0.4184	0.434	0.00110	0.00293	1.72e-05

The goodness of fit provided by the estimated parameters in Table I has been measured using λ^2 family of statistics. The original formulation of λ^2 proposed in [13] may face divide-by-zero situations for intervals where the expected value for the theoretical distribution is 0. Moreover, it is quite sensitive to extreme values and specially to the tail of the empirical distribution as can be observed in Table II, in the columns labelled as λ^2 for both Exponential and Weibull distribution. While providing values quite low for 1, 2 and 3 client sessions, λ^2 greatly increases several orders of magnitude for greater number of players. Thus, the value for the 4-client session is 2398, reaching 10^7 for 6 and 7 clients. It must be noted that negative values and values close to 0 mean a high degree of fitting between the theoretical and empirical distribution. On the other hand, values greater in several orders of magnitude are an evidence of lack of fitting, but cannot be considered an absolute fitting metric, but as a way to compare the fitting of samples from a common population. Thus, a λ^2 value in the order of thousands indicates a worse fitting that a value close to 0, but does not translate into a several orders of magnitude worse distribution. A proof of this is given by the huge values of λ^2 observed for the Exponential distribution in Figure 2. A close inspection of the empirical data reveals that λ^2 is not representative of the deviation between the theoretical and empirical distribution. When the more extreme values from the empirical distribution tail were removed (less than 1% of the observations), all the new λ^2 values were in the range of tens, as it was also observed in [5]. On the other hand, discrepancy results for Weibull parameters were close to 0 for all the testing sessions, indicating a very high degree of fitting between the empirical data and the proposed theoretical distribution.

TABLE II: λ^2 discrepancy for estimated parameters

	Exponential		Weibull	
N.Users	λ^2	$\hat{\lambda}^2$	λ^2	$\hat{\lambda}^2$
1	1.420e-01	11.471141	0.25207	11.741701
2	3.372e-01	1.176080	0.20700	1.348291
3	1.156e+01	1.166329	0.11331	1.146745
4	2.398e+03	1.304024	0.10902	1.179470
5	5.651e+06	1.426090	0.08907	1.201576
6	1.570e+07	1.565878	0.07524	1.139104
7	1.136e+07	2.791518	0.07025	2.108351
8	3.392e+05	2.811499	0.08808	2.097059
9	8.192e+03	3.929104	0.06901	3.069065
10	1.597e+00	3.741701	0.29100	3.078933

To avoid the divide-by-zero situations mentioned above, [4] proposed an alternative discrepancy metric, $\hat{\lambda}^2$, which shared the same theoretical basis and analytical behaviour that original $\hat{\lambda}^2$. This new metric also deals better with tail values and outliers, as can be observed in Table II, in the columns labelled as $\hat{\lambda}^2$ for both Exponential and Weibull. Discrepancy values for exponential distribution were sensibly smaller and stable in this case. The discrepancy slightly increases with the number of users. The trend for Weibull fitting is similar to the Exponential one, the resulting $\hat{\lambda}^2$ values are even greater that those observed for Weibull using the original λ^2 metric.

Trends for the estimated parameters can be better appreciated in the Figures 2 and 3. Figure 2 is composed of three plots. The upper one reveals that rate λ for the exponential distribution increases quadratically with the number of users. This behaviour is plausible considering that in a n-client session, each client update implies propagating $n-1$ updates to the rest of users. The one at bottom left shows λ^2 for estimated values of Exponential rate λ, values for 5, 6 and 7-client sessions reach values of several orders of magnitude over the rest due to the presence of outliers and tail values in the empirical distribution. The bottom right plot shows the $\hat{\lambda}^2$ for estimated Exponential parameters. The distribution of these values is much more uniform showing a slight increase in discrepancy with the number of users.

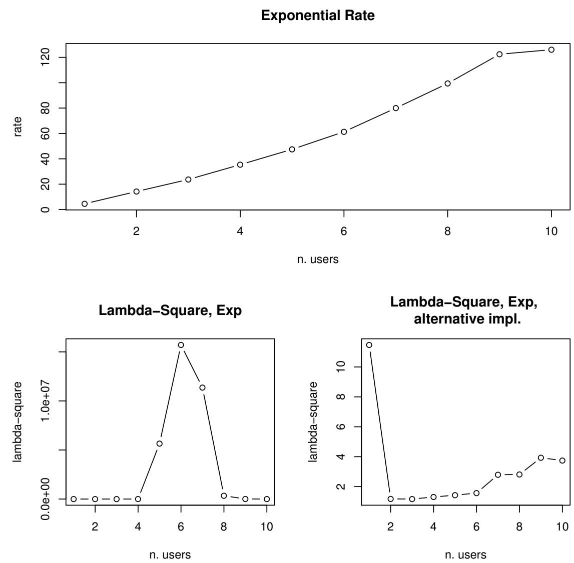

Fig. 2: Estimated Exponential parameters and λ^2 discrepancy

Figure 3 is divided in four plots. The two situated at the top of the figure show the evolution of the Weibull parameters shape α and scale β estimated by MLE. Both of them show an asymptotic decrease and are susceptible of being fitted by analytical curves. The bottom left plot displays the λ^2 values for Weibull fitting, it suggests that the discrepancy values follow a decreasing trend with some exceptions. The bottom right plot corresponds to the $\hat{\lambda}^2$ values, their shape is quite similar to its equivalent in Figure 2.

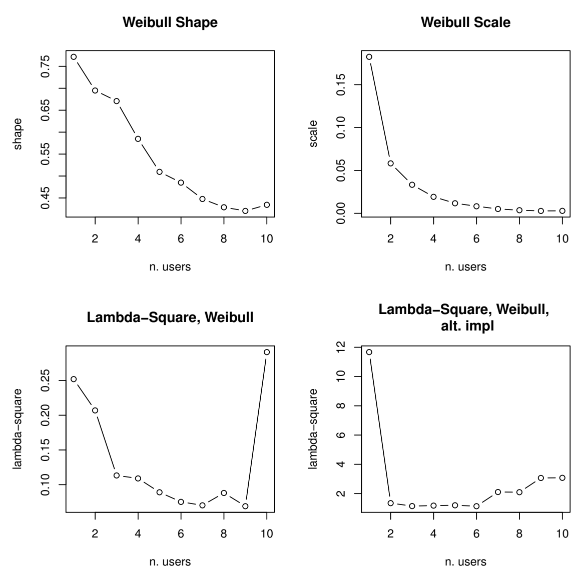

Fig. 3: Estimated Weibull parameters and λ^2 discrepancy

λ^2 and $\hat{\lambda}^2$ plots in Figures 2 and 3 provided distorted or even contradictory information about the goodness of fit between the empirical data and the proposed theoretical distributions. To determine which of the metrics were more suitable for the case of study, Q-Q plots were used. The closer the values are from the gray line, the better fit. Due space restrictions, Figure 4 shows quantile comparison only for 2, 4, 6 and 8 client sessions, the obviated plots do not differ significantly while confirming the results shown above. These Q-Q plots point the distribution tails as the regions of maximum discrepancy for both Exponential and Weibull distributions. Discrepancy for exponential models increases with the number of clients, so their increasing $\hat{\lambda}^2$ values are coherent with these plots. On the other hand, Weibull distributions show smaller discrepancy with the increase of users, this trend is not well captured by the $\hat{\lambda}^2$ values in Figure 3 which may led to think that Weibull discrepancy increases with the number of users. On the contrary, λ^2 values suggest the same decreasing discrepancy trend observed in the Q-Q plots.

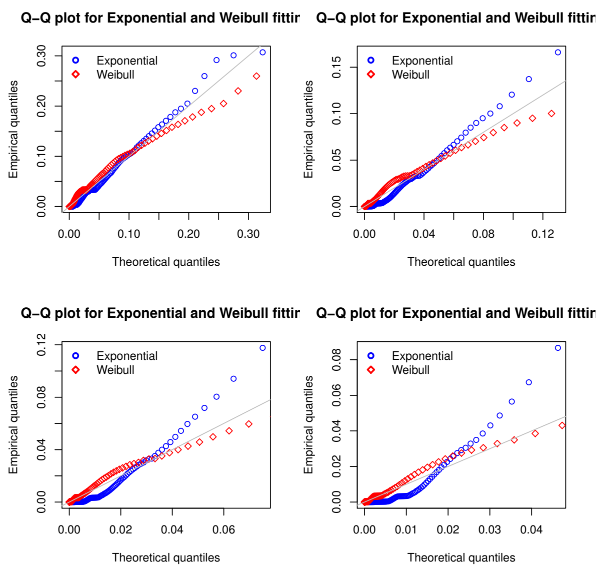

Fig. 4: Q-Q plots for IDT, Exponential and Weibull curves

B. *Packet Size of Outgoing traffic*

The autocorrelograms generated for PSO suggested a slight increase of inverse autocorrelation with the increase of the number of users. Figure 5 shows four ACFs for 2, 4, 6 and 8-client sessions. The 4-client ACF shows signs of inverse autocorrelation for lag 1. This trend is confirmed in the plots for sessions with more than 4 clients. This can be interpreted as a certain probability of sending a big packet followed by other significantly smaller or vice versa.

Outgoing TCP traffic shows a discrete set of packet sizes, only some of them with a significant frequency. Figure 6 contains several histograms showing the overall amount of packets of each size. For space constraints, these histograms correspond to 2, 4, 6 and 8-client sessions respectively. An important part of all the observed packets sizes fall within the intervals [180-235] B and [290-340] B. To avoid increasing the

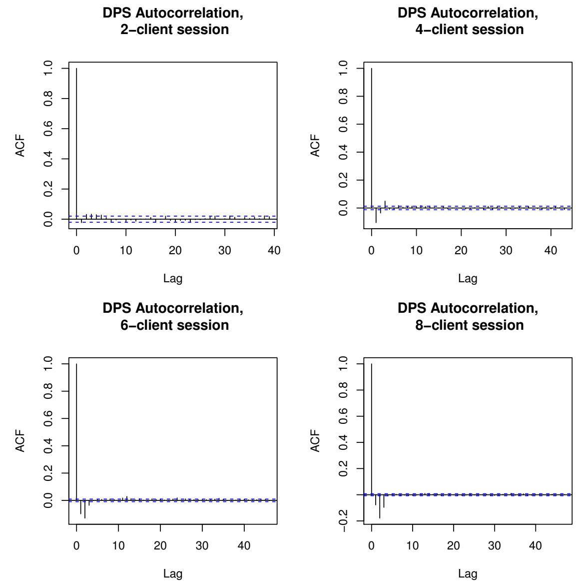

Fig. 5: Autocorrelation results for PSO

complexity, these intervals have been represented by a pair of single values calculated by weighted mean: 217 B and 325 B. Packets of 1460 B size resulted the most frequent along the testing sessions.

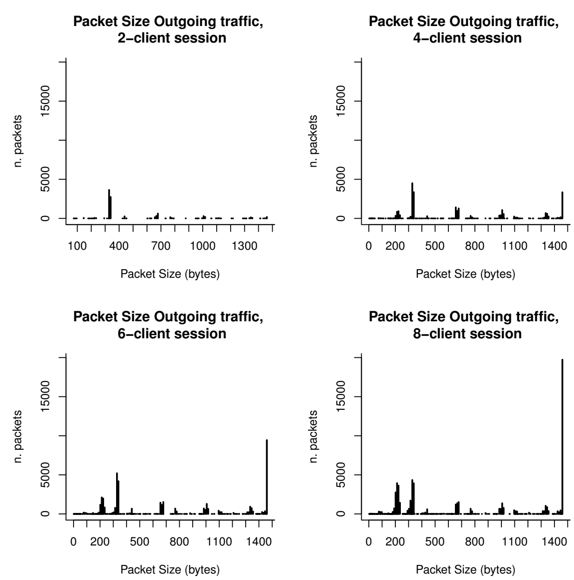

Fig. 6: Packet size distribution for 2, 4, 6 and 8-client sessions

The three first graphs in Figure 7 (top left and right, bottom-left) show the evolution of packets represented by 217 B, 325 B and 1460 B (solid black line). Dashed red lines in represent the best fitting polynomial curves for each packet size. These curves have been calculated by Gauss-Newton algorithm for non-linear least-square (weighted) estimation. The grade of each polynomial has been chosen keeping a balance to avoid over-fitting as well as inconsistent estimations such as negative number of a determined packet size. The final plot (bottom-right) displays the relative frequency of each packet size for each testing session.

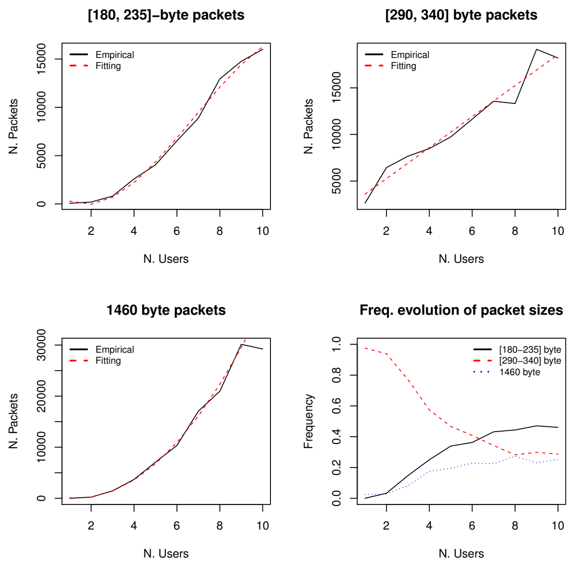

Fig. 7: Number of packets evolution and size frequency

The fitting curves calculated by least-square estimation have been used to elaborate the deterministic model proposed in Equation 1. Depending on the number of concurrent users n, the model assigns a specific probability for each one of the contemplated packet sizes. The polynomials used to determine this probability are showed in Equation 2.

$$P(x;n) = \begin{cases} 217bytes, & p_1(n)/p_t(n) \\ 325bytes, & p_2(n)/p_t(n) \\ 1460bytes, & p_2(n)/p_t(n) \end{cases} \tag{1}$$

$$\begin{cases} p_1(n) = -35.8n^3 & +726n^2 - 2232.0n & +1825 \\ p_2(n) = & + 1664.0n & +1930 \\ p_3(n) = & +502n^2 - 1343.5n & +927 \\ p_t(n) = -35.8n^3 & +1228n^2 - 1911.5n & +4661 \end{cases} \tag{2}$$

VI. Conclusions

The present paper has performed a study of the outgoing TCP traffic of an Open Wonderland game server during several testing sessions. This TCP traffic is mostly due to avatar position and movement updates within the virtual world. The number of concurrent users ranged from 1 to 10, while keeping a high interaction rate to maximise network load. Inter Departure Time (IDT) and Packet Size of Outgoing traffic (PSO) have been the studied parameters, given their significance for the QoE.

The IDT revealed that the OWL server does not transmit update packets on periodic basis nor has bursty traffic. The theoretical distributions which best described the IDT were Weibull and Exponential ones. A set of parameters for them were calculated by Maximum Likelihood Estimation (MLE). To determine the goodness of fit of each distribution λ^2 and $\hat{\lambda}^2$ approached were used, together with Q-Q plots.

The PSO showed a slight increase of inverse autocorrelation with the number of users. The user increasing implies a quadratically increasing in the number of larger packets, which translates in greater payloads for position update packets. The experimental results have been used to propose a deterministic model that depends on the number of concurrent users to determine the probability of the most significant packet sizes.

Future work involves checking the validity of the proposed models with more realistic player behaviour and confirming the impact over the tail of the proposed IAT/IDT models.

Acknowledgment

The authors would like to thank Prof. Michele Pagano, from the University of Pisa; and Prof. Daniel Cascado, from the University of Seville for their support to the presented work. This work has been partially supported by Project GameUp: Game-Based Mobility Training and Motivation of Senior Citizens (AAL-010000-2012-10) and project Vulcano: TEC2009-10639-C04-02.

References

[1] D. Cascado, S. Romero, S. Hors, A. Brasero, L. Fernandez-Luque, and J. Sevillano, "Virtual worlds to enhance ambient-assisted living," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Sep. 2010, pp. 212 –215.

[2] S. J. Romero, L. Fernandez-Luque, J. L. Sevillano, and L. Vognild, "Open source virtual worlds and low cost sensors for physical rehab of patients with chronic diseases," in *Electronic Healthcare*, P. Kostkova, Ed. Springer Berlin Heidelberg, Jan. 2010, no. 27, pp. 76–79.

[3] S. Ratti, B. Hariri, and S. Shirmohammadi, "A survey of first-person shooter gaming traffic on the internet," *IEEE Internet Computing*, vol. 14, no. 5, pp. 60 –69, Oct. 2010.

[4] M. S. Borella, "Source models of network game traffic," *Computer Communications*, vol. 23, no. 4, pp. 403—410, Feb. 2000.

[5] A. Dainotti, A. Pescape, and G. Ventre, "A packet-level traffic model of starcraft," in *Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems*, ser. HOT-P2P '05. Washington, DC, USA: IEEE Computer Society, 2005, p. 33–42.

[6] P. Branch and G. Armitage, "Extrapolating server to client IP traffic from empirical measurements of first person shooter games," in *NetGames' 06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, vol. 24, 2006.

[7] M. Gardner, G. A. Gánem-Gutiérrez, J. Scott, B. Horan, and V. Callaghan, "Immersive education spaces using open wonderland: From pedagogy through to practice," 2011.

[8] J. L. Font, D. Cascado, J. L. Sevillano, F. Díaz del Río, and G. Jiménez, "Network traffic analysis and evaluation of a multi-user virtual environment," *Simulation Modelling Practice and Theory*, vol. 26, no. 0, pp. 1–15, Aug. 2012.

[9] W. Feng, F. Chang, W. Feng, and J. Walpole, "Provisioning on-line games: a traffic analysis of a busy counter-strike server," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 2002, p. 151–156.

[10] "AutoHotkey," http://www.autohotkey.com/.

[11] "HotkeyNet: home," http://hotkeynet.com/.

[12] "Random," http://www.autohotkey.com/docs/commands/Random.htm.

[13] S. P. Pederson and M. E. Johnson, "Estimating model discrepancy," *Technometrics*, vol. 32, no. 3, p. 305, Aug. 1990.

[14] "Top 5 operating systems on oct 2012 | StatCounter global stats," http://gs.statcounter.com/#os-ww-monthly-201210-201210-bar.

[15] P. Branch, "Measuring the autocorrelation of server to client traffic in first person shooter games," in *in Australian Telecommunications, Network and Applications Conference ATNAC*, 2006.