A packet-level Traffic Model of Starcraft

Alberto Dainotti, Antonio Pescapé, and Giorgio Ventre
University of Napoli "Federico II", Naples - Italy
Email: *alberto,pescape,giorgio@unina.it*

Abstract

Starcraft is a popular Real Time Strategy game that uses a Peer-to-Peer network communication model. In this paper we analyze its traffic and we provide a statistical characterization at packet-level obtained by varying the number of players. We examine the time dynamics between individual packets within a game session as well as the packet sizes. Also, we provide analytical models approximating the empirical distributions found, we study properties of the tails and of the auto-correlation function, and we investigate the presence of self-similarity. The results obtained show how traffic generated by such game has different characteristics from the traffic prevailing on the Internet in past years.

1. Introduction

Multi-player network games represent one of the most popular examples of real-time, interactive commercial Internet applications and traffic generated by them is rapidly increasing, becoming a significant contributor to overall Internet traffic. Traffic generated by network games is of interest not only because of its market potential but also because its characteristics are poorly understood, making it difficult to assess the impact of such traffic on large networks. Hence, a complete statistical descriptions useful to carry out a simple traffic model could be needed. In particular, we would like to find statistical characteristics of how a gaming host generates network traffic that can be parametrized for analytical models and simulation.

In [1] it is reported that nearly 4% of all packets in a backbone could be associated with only 6 popular games and in USA alone, they are currently worth a significant fraction of the 7 billion dollars computer games industry [2]. In [1] it is also reported that the multi-player network computer games, most of them based on a Peer-to-Peer paradigm, are predicted to make up over 25% of LAN traffic by the year 2010. U.S. computer and video game software sales grew 8% percent in 2003 to 7 billion dollars, a more than doubling of industry software sales since 1996. In 2003, more than 239 million computer and video games were sold, or almost two games for every household in America [16].

The amount of Internet traffic generated by computer games is expected to increase fast, especially because new players are entering the Internet with game consoles that support Internet connections. Indeed, the game console industry has also recognized the rapid growth of multi-player games and with the launches of Microsoft's Xbox, Dreamcast (from Sega Corp.), and Sony's Playstation II on-line game networks, and with the emergence of massively multi-player on-line games, it is clear that a further large increase in gaming traffic is imminent.

Obviously, Internet Service Providers are interested in being able to provide a quality and efficient service to the gaming community. But to make provisioning of network resources it is necessary to understand traffic properties. As we will see also in this work, interactive games traffic has different characteristics to the TCP-based traffic prevailing on the Internet in the last years and that has received most of the attention in the network research community.

In particular, network game traffic tends to employ small, highly periodic UDP packets. High periodicity is due to game's dynamics, which require frequent state updates from each peer. While UDP is often used as a transport protocol because of minor protocol overhead and because there is no time and usefulness in resending lost packets. Also, the extremely low latency demand of such applications makes message aggregation impractical, which leads to small packets.

We chose a representative game from a popular genre, Real Time Strategy games, that is Starcraft, which is based on a synchronous Peer-to-Peer paradigm. In this architecture, every computer calculates the position and actions of every player in the game. Computers do not send messages over the network like "I hit you for 10 points" that are associated to asynchronous events. Instead, they periodically send keyboard and mouse input. In our traffic analysis we consider modeling at a *micro scale*. In other words we examine traffic at the packet-level, estimating distributions and statistical properties of packet lengths and inter-packet times.

The remainder of this paper is organized as follows. Section 2 contains an analysis of the related work. In Sec-

Proceedings of the 2005 Second International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'05)
0-7695-2417-6/05 $20.00 © 2005 IEEE
0-7695-2426-5/05 $20.00 © 2005 IEEE

tion 3 we describe the statistical methodology used in modeling, whereas in Section 4 we describe the network scenario and the traffic traces we used. In Section 5 we present and discuss the results of our study. In Section 6 we compare our results with other previous works. Section 7 ends the paper with some conclusion remarks.

2. Related Work

Multi-player network games, and network traffic generated by them, have been subject to interest by the academic community only in recent years, while industry works have been more focused on the management aspects of game development or on latency and maximum bandwidth issues only ([3], [4], [5]). In the academic world, the first works related to traffic modeling of network games were presented by Gautier and Diot [6] and Borella [19]. Gautier and Diot [6] designed MiMaze, a distributed game for the Internet using a multicast communication system and conducted experiments to collect data on the network traffic generated. Borella [19] has provided an in-depth analysis of traffic traces from the popular multi-player first-person shooter game *Quake*. Empirical distributions of packet size and packet inter-arrivals have been found and analytical distributions approximating them have been obtained through statistical fitting. In [8] Feng *et al.* describe results of the analysis of a 500 million packet trace of a popular on-line, multi-player, game server. They found that the behavior of the traffic generated by the server was highly predictable, something that was attributed to the fact that the designs of the games involved target the saturation of narrowest last-mile links. They also found that observed on-line games provide significant challenges to current network infrastructure because of the presence of large, highly periodic bursts of small packets. In [9] Feng and Chang study and model the player session time distribution over a one-week trace of a popular on-line game server. In [7] Farber evaluates Counter Strike game from a 36 hour LAN party measurement and presents traffic models for client and server in terms of packet size and inter-arrival times. In [26] the network traffic patterns of Counter Strike and Starcraft were examined and documented. Analysis focused on bandwidth usage, packet size and inter-arrival times. Our study is based on the traffic traces that were collected and used in this work. In [27] a study on the characteristics of traffic generated by Starcraft, focusing on how the distributions of payload sizes and inter-arrival times, is presented. We give details on analogies and differences, in terms of both methodology and results, between our work and the last two cited papers in Section 6. In [10] a synthetic traffic model for Half Life is shown. A *ns*2 model simulation model for packet length, packet inter-arrival time, and data rates is also presented. In [11] four different classes of games were defined: action games, simulators, real-time strategy games, and turn based strategy games. Traffic generated by the representatives of these four different classes was measured and analyzed in terms of packet size distribution and packet inter-arrival time distribution. One of the main results of this study was that the amount of traffic generated by different games could vary heavily. The authors observed small packets of a few distinct sizes rather than continuous packet size distributions. In most cases, they modeled inter-arrival times by multimodal distributions consisting of extreme, normal or exponential distributions. In [12], in order to characterize traffic generation patterns, network traces generated by four network game applications were analyzed. In [13] a simple *ns*2 simulation model for Server and Client Xboxes was developed. Traffic characteristics observed were packet length, packet inter-arrival times, packets per second, and data rates. In [15] a per-player traffic modeling methodology is presented. In [14] an analysis of traffic generated by the popular Internet-based on-line game engine, Unreal Engine, is presented. Network parameters such as packet length, inter arrival time and aggregate data rate are observed for analyzing aspects of self-similarity.

We have not found in literature a detailed packet level traffic model of file-sharing applications, the most common Peer-to-Peer applications. All works focused most of their attention on flow level characterization. Authors of [31] found that eDonkey flows can be divided in mice and elephant and those TCP connections have a rather small bit rate and there is no evidence for long range dependence. Authors of [30] found similar results; they also found that Peer-to-Peer traffic of file-sharing applications (in particular eDonkey traffic) increases the presence of the “mice and elephant” phenomenon in the Internet traffic characteristics. They also found that there are no heavy tails in distributions of flow size and flow inter-arrival times. Authors of [32] present a characterization of Peer-to-Peer traffic in the Internet and develop several heuristics that allow them to recognize Peer-to-Peer traffic at nonstandard ports. They perform an analysis of block size and packet format for each distributed file-sharing application.

3. Background and Statistical Methodology

A statistical analysis of the measured samples in real traffic traces has been provided by setting up a methodology that integrates well-known established techniques separately found in different works, as distribution estimation, statistical fitting, study of the tails and of the autocorrelation function.

We represented the empirical distributions of the studied variables not only by estimating the corresponding Cumulative Distribution Functions (CDF), but also by estimating

Probability Density Functions (PDF). The latter ones have been obtained as density histograms.

For each of the distributions we performed a statistical fitting to find an analytical distribution approximating the empirical one. Indeed, we are interested in obtaining analytical models because they offer several advantages, as conciseness and ease of tractability. We proceeded as follows. Given a class of known statistical models (Exponential, Normal, Weibull, ...) we used the *Maximum Likelihood Parameter Estimation* (MLE) to determine, for each of them, the shape and location parameters that maximize the likelihood function of the sample data. After that, to choose the best fitting analytical distribution among them, we used the λ^2 discrepancy measure [20] which is based on the X^2 statistic. Goodness-of-fit statistics indicate whether lack of fit is statistically significant but do not directly measure the magnitude of the departure. The size of such departures, which is called discrepancy, is often of interest, for example when model fits of several different data sets must be compared. The X^2 statistic partitions the sample set into contiguous fixed-size bins, and compares the relative frequency of the samples from the empirical set falling into each bin with the expected number of observations for the analytical distribution. Let N be the number of samples in empirical distribution Y, which is partitioned into n bins. We have

$$X^2 = \sum_{i=1}^{n} \frac{(Y_i - N \cdot p_i)^2}{N \cdot p_i} \tag{1}$$

Where Y_i represents the number of samples from Y that fall in the i-th bin and p_i is the expected number of observations from the theoretical distribution Z falling into the same bin. If we then define $E_i = N \cdot p_i$, $D_i = Y_i - E_i$ and the quantity $K = \sum_{i=1}^{n} D_i/E_i$ we have

$$\lambda^2 = \frac{(X^2 - K - dF)}{(N-1)} \tag{2}$$

where dF represents the number of *degrees of freedom* in computing X^2 and K, which for our purposes is given by $dF = n - 1 - Est$. Est is the number of parameters used to estimate the analytical distribution Z.

One of the advantages of the λ^2 estimator is that, incorporating both the sample size N and the number of bins n, it is independent from such values and can be used to compare different sample sets. The *variance* associated with the estimate of λ^2 is given by:

$$\widehat{v}(\lambda^2) = \frac{[2dF + 4N\lambda^2 + 4N\lambda^4 + 4T]}{N^2} \tag{3}$$

where: $T = \sum_{i=1}^{N} \frac{[D_i^3 - 2D_iE_i + \frac{5}{2}D_i^2 + \frac{3}{2}Y_i]}{E_i^2}$. There are rare situations in which the probability associated with a bin is zero, for example when the analytical distributions that is considered is a deterministic distribution. In such situations the above formulas cannot be computed. Borella [19] suggests to modify them calculating D_i as $D_i = E_i - Y_i, \forall i : Y_i \neq 0$. Then we compute X^2 not as shown before but as: $X^2 = \sum_{i=1}^{n} D_i^2/Y_i, \forall i : Y_i \neq 0$ and $K = \sum_{i=1}^{n} D_i/Y_i, \forall i : Y_i \neq 0$. Borella does not suggest a corresponding formula to calculate the variance of the λ^2 estimate. Such variance is useful in comparing the discrepancy measures obtained for different analytical distributions, because when dealing with estimated quantities the error of estimation could be so large that a comparison using the "$<$" operator would be misleading. When the variances are available instead we use $\widehat{a} <_\sigma \widehat{b} \Leftrightarrow \widehat{a} + \sigma_a < \widehat{b} - \sigma_b$ [18].

Once the best fitting distribution has been chosen, we can provide visual displays of the quality of the fit by plotting the empirical an analytical PDFs and CDFs on the same graph, and comparing the distributions with quantile-quantile (Q-Q) plots. In practice we often find deviations in the fit. The power of the Q-Q plot is that we can easily determine where those deviations occur (i.e., in the main body, the upper tail, etc.). The Q-Q plot has been used extensively in networking literature for this purpose [22][21][19]. On the plot, corresponding quantiles of each distribution are graphed against one another; for example, the median is graphed against the median, the upper quartile is graphed against the upper quartile, and so forth. If the points follow the line with intercept 0 and slope 1, drawn on the plot, then the distributions are identical. When the Q-Q plot or the shape of CDF (PDF) indicates a deviation, we may prefer to model the data set with a *split distribution*. In this case, we model part of the data set with one distribution and the rest of it with another. Obviously, we can split a distribution as many times as necessary, but more than four splits results in a cumbersome analytical model.

Often a distribution's behavior in its upper tail can be crucially important. For example, Paxson [18] found that in FTP traffic the upper 2% tail is so heavy that rare bursts will often completely dominate FTP traffic. The λ^2 discrepancy measure does not give any special weight to the agreement between tails of two distributions, so we adopted a quantitative [18] analysis of how well a model captures distribution's tails behavior. Suppose we have an empirical model Y and an analytical model Z that we have found to be best fitting model. For our convention, we consider as extreme tail of the empirical distribution elements that in CDFs fall in ranges: $[90\%, 100\%]$ for upper tail. Let b be the number of instances of empirical distribution that lie in the range of values derived from the given tail of the empirical distribution and let a be the number of instances of analytical distribution that lie in the same range derived from the same tail of empirical distribution. Define: $\xi = \log_2 a/b$. Positive values of ξ indicate that the model *overestimates* (symbol $+$) the tail, negative values that it *underestimates* (symbol $-$) the tail. A value of $\xi = 0$

(or < 0.01) indicates that model perfectly estimates tail behavior and it is indicated with "ok". An acceptable value is for $0 < \xi < 1$ and, following the notation adopted in [18], is indicated with "+". A bad value is for $1 < \xi < \log_2 5$ and is indicated with "++". A very bad value is for $\xi \geq \log_2 5$ and is indicated with "+++". This quantitative evaluation will be presented in form X/Y where X will be the number of elements discarded from tails in analysis (indeed there can be some outliers not important for modeling) and Y one of symbols mentioned above.

Also, the presence of power-law behavior in the upper tail of a distribution has important implications. A random variable X follows a heavy-tailed distribution, with tail index α, if $P[X > x] \sim cx^{-\alpha}, as x \to \infty, 0 < \alpha < 2$. Where c is a positive constant, and where $\sim$ means that the ratio of the two sides tends to 1 as $x \to \infty$. This distribution has infinite variance. It is possible to estimate the α parameter by plotting the *Log-Log Complementary CDF* plot [24] (CCDF). We consider a distribution X and its CDF $(x, F(x))$; then we plot $\ln(1 - F(x))$ versus $\ln(x)$ for all x. If the upper tail on the plot has a linear behavior, an estimate for α can be obtained selecting a minimal value x_0 of x above which the plot appears to be linear and estimating the slope for values greater than x_0. A problem of such method is that one must determine some point x_0 in the tail at which power-law behavior begins. The "scaling method" proposed in [23] helps to identify the portion of a dataset's tail that exhibits power-law behavior. This method is based on a particular property of heavy tailed distribution: the tail index is unchanged when heavy-tailed random variables are summed or aggregated. By aggregating a data set of N observations $X_i, i = 1, \ldots, N$ we refer to the process of summing non-overlapping blocks of observations of size m: $X_i^{(m)} = \sum_{j=(i-1)m+1}^{im} X_j$. By observing some distributional properties of $X^{(m)} : X_i^{(m)}, i = 1, \ldots, \left[\frac{N}{m}\right]$ we can make inferences about where in the tail power-law behavior begins. Based on these determinations we have the basis to estimate the tail index α. In Figure 1 an example of the scaling property in the tail of the distribution for a Pareto distribution is depicted. Tails of successive data sets are approximately parallel, with slope approximately $-\alpha$.

For each studied variable we also evaluated the correlation between subsequent samples, that is, we estimated the auto-correlation function at lag 1, $r(1)$, also indicated with $ACF(1)$. $r(1)$ is a particularly significant value of $r(l)$, because if a random variable is correlated, often the correlation is greatest at a lag of 1. Also we reported the auto-correlation plots from lag 1 to 100 to infer possible properties of Long Range Dependence. In recent years there has been a lot of interest by the research community in investigating behaviors of traffic statistics on different time scales.

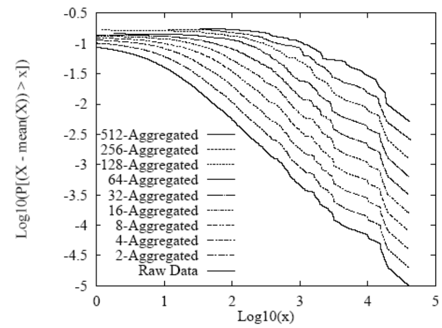


Figure 1. Scaling behavior in a synthetic data set (source [23])

In particular, the presence of self-similarity in network traffic and its bad impact on network nodes have been shown in several works [33][34][35]. Also, many techniques to evaluate the presence of self-similarity, by estimating the Hurst parameter, in a stochastic process have been developed. In this work we used the wavelet transform estimation [36], which is considered one of the most reliable techniques [25], applied to inter-packet times and packet rate.

4. Network Scenario and Traffic Traces

We analyze the traffic generated by *Starcraft*, a Real Time Strategy game based on a Peer-to-Peer communication structure among players (Figure 2(a)). In such game,

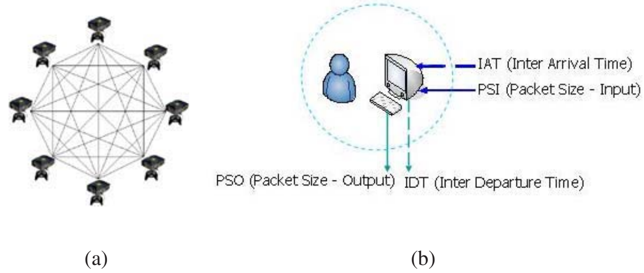


Figure 2. Peer-to-Peer topology for online games (a). Modeled variables (b).

players construct buildings and fighting units and issue commands that cause the units to move, engage enemy units and similar tasks. Every game is played on one of many possible *maps*, either provided with the game or custom built by users. There are three races from which a player can choose, and each of them has a balanced set of advantages over the others. There are a number of ways in which players can be competitively grouped. In a free-forall games, all players compete to have the last remaining army on the map. Players can also team up against each other and/or against automated "computer" players. *Starcraft* supports up to 8 players and uses the following communication model [27]. At the start of a game session, a listen server (a playing machine as well as a game hosting machine) is used to set up the current session. In this phase, there can be also TCP packets between the server and the participants. Once the session has been set up, every participating computer sends packet to all the others, irrespectively of the initial server used to set up the session. Due to low latency necessity, UDP is used as the transport protocol in this case. Unfortunately, UDP does not provide any built-in congestion control, presenting the risk of congestion collapse as the fraction of unresponsive UDP traffic increases.

As for the empirical data, in our analysis we use the traffic traces available at [17] and used for the traffic analysis made in [26]. The tests have been performed by using *Starcraft: Brood War* version 1.7. A local player has been logged on the *battle.net* server [28] by using the USEAST gateway. Thank to it, he has created the game sessions. Each game type has been made by top players vs. bottom players. The same map, called *Big Game Hunters* (found in the *maps/broodwar/webmaps* directory) has been used for each trial. Tests were structured by having two player teams of equivalent sizes: 2 vs. 2, 3 vs. 3 and 4 vs. 4. The point of observation is that of a single peer, as depicted in Figure 2(b). The local player played the same race in each game and employed the same building strategy throughout. Our approach is to model source traffic at *packet level* and not at *flow level*, therefore we examined IP packets inter-arrival times (IAT) and inter-departure times (IDT) as well as packet sizes of inbound (PSI) and outbound packets (PSO). As packet size we considered the byte length of the transport-level payload, because we want to model traffic as it is generated by the application. As for the time resolution adopted in measurements, in the analyzed traces [17] we have, both for IAT and IDT, a resolution of 1 millisecond. This means, for example, that an inter-arrival (inter-departure) time of *0 ms* represents a situation where the inter-arrivals (inter-departures) time are: $IAT\ (IDT)\ <\ 10ms$. We discarded packets with inter-arrival or inter-departure times > 1 second or packets with transport protocols different from UDP or TCP. We have chosen to discard these packets in order to remove traffic patterns resulting from player pauses or waiting time between matches or turns. As shown in Table 1 they represent a negligible portion of the total number. The hypothesis at the base of our work are the following: (i) We suppose our modeling independent from hardware resources of each player. This means that we do not study how differences in available processing resources and/or other resources may affect the traffic generated by each user [27]; (ii) We suppose that resource contention does not introduce variations in the packet inter-arrival or inter-departure times distributions and in the packet size in or out distributions. We assume that resource contention is low and therefore has no significant impact on the characteristics of collected data [27]; (iii) We suppose that all random variables studied are *i.i.d.*

5. Results and discussion

We applied the statistical methodology presented in Section 3 to each of the four analyzed random variables (*IAT*, *IDT*, *PSI*, and *PSO*) and for three different gaming scenarios (4 players, 6 players, 8 players). Because of space constraints, in this paper we report detailed tables and figures showing results only for the 6-players scenario, which is the one with the largest collected trace. In Table 2 the results of analytical modeling are shown. The columns in the table are to be read as follows: (i) random variable; (ii) best analytical model chosen for the variable with an offset to subtract (note that for IDT, PSI, and PSO we have used split distributions); (iii) parameters of the chosen distribution; (iv) discrepancy; (v) quantitative analysis of the upper tail; (vi) estimate of the α parameter; (vii) auto-correlation function at lag 1. Figures 3 and 4 show comparisons between the empirical and the analytical CDF and PDF respectively; Figure 5 presents heavy tail analysis.

5.1. Packet Size

To obtain analytical distributions approximating the empirical ones we chose, both for PSI and PSO, to split the distributions into few parts to capture the behavior of a main peak and other lower peaks, which were fitted with deterministic distributions. PSI and PSO have indeed almost identical distributions, with more than 70% of the samples presenting a packet payload 23 bytes long and with a maximum not negligible packet size of 33 bytes. Such a behavior is expected from a Peer-to-Peer game in which each player sends out multiple copies of its update packets to each other peer. The updates need to be small, to keep latency low, and frequent, so to give the illusion of real-time interaction. We note though, that with such small packets protocol overhead is high. Indeed if we count the IP header, which is at least 20 bytes long, plus 8 bytes of UDP header the sum is greater than the average payload size. For this reason, often this kind of games have support to run over IPX networks to be exploited in dedicated environments as *Internet cafes* hosting multi-player games sessions. We found identical results with different numbers of players, that is, for the 4 and 8 players scenarios. We can conclude that Starcraft produces very small pack-

Number of Players	Number of Packets	Log Time	UDP Packets	TCP Packets	Discarded Packets
4	281157	4h:20m:22s,422ms	281157	0	22
6	415107	4h:51m:20s,830ms	415107	0	12
8	60976	0h:27m:10s,964ms	60976	0	2

Table 1. Collected traffic traces

Var	Model	Parameters	λ^2	Tail	α	ACF(1)
IAT	*exponential*	$\mu = 0.043633$	0.37068	$0/--$	5.6572	-0.14095
IDT	determ $p = 66.2\%$ uniform $p = 27.8\%$ determ $p = 6\%$	$a = 0$ $a = 0.05$ $b = 0.17$ $a = 0.21$	0.337632	$0/-$	6.2967	-0.185757
PSI	determ $p = 3.2\%$ determ $p = 10.8\%$ determ $p = 72.4\%$ determ $p = 6.2\%$ determ $p = 7.4\%$	$a = 16$ $a = 17$ $a = 23$ $a = 27$ $a = 33$	0.0808382			0.0360888
PSO	determ $p = 6.2\%$ determ $p = 10.9\%$ determ $p = 74.2\%$ determ $p = 8.7\%$	$a = 16$ $a = 17$ $a = 23$ $a = 27$	0.1497652			0.503225

Table 2. Starcraft: Summary of Results for 6 Players scenario

ets, with small variance, and that the payload distribution is independent of the number of players. The fact that the distribution of packet size is similar for inbound and outbound packets and that we encountered approximately the same number of packets for both directions are a consequence of the symmetrical communication structure of a Peer-to-Peer game. As for correlation analysis, Table 2 indicates for PSI a low value of $ACF(1)$, while for PSO we find a considerable auto-correlation between subsequent packets. This is explainable with the fact that a player sends the same update information in the form of back-to-back packets, of the same length, destined to different peers. In [11] an $ACF(1)$ analysis has been made for 4 popular client-server games. It is interesting to note that the results reported in such work are quite different, with values close to 1 both for client and server packet sizes.

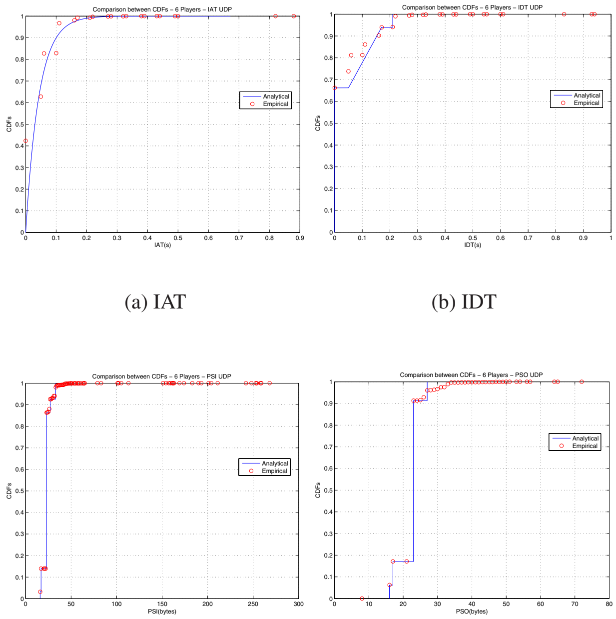
5.2. Inter-packet times

Figures 3(a) and 3(b) show comparisons between empirical and analytical CDFs for IAT and IDT. We can see how the analytical shapes capture the behavior of the empirical ones, confirming the goodness of the fit. For IAT we can see a more regular behavior that allows to choose a simple analytical model. Indeed IAT are well approximated by an Exponential distribution, as reported in Table 2. Contrariwise, for IDT we can see a very irregular behavior of CDF. We obtained considerably smaller values of discrepancy measure by splitting the distribution into three parts and separately fitting them to different analytical distributions (see Table 2).

Unlike what happens for payload sizes, we observe a visible difference between inter-packet time distributions of inbound and outbound packets. Even though both in IAT and IDT distributions 99% of the values are smaller than $200ms$, IDTs are more concentrated in zero (which we remind it corresponds to values $< 10ms$) counting for almost 70% of packets, while about 40% of IATs are $< 10ms$ and the corresponding CDF curve reaches 99% at $200ms$ more smoothly than for IDTs. We can explain such behavior with the Peer-to-Peer structure of the game: if we suppose that at an instant every player sends an update packet, from the point of view of the observed player we see a series of outbound back-to-back packets towards the other peers while in the inbound direction each update packet comes from a different peer and arrives through a different path (that is, with a different one-way delay). Also updates can be sent at different times by each peer. Both reasons explain a major variability in packet inter-arrival times.

As expected, smaller IATs and IDTs become more dominant as the number of players increase. In Figures 6 we show the CDFs of IDT and IAT for the 4 and 8 players scenarios. We have that values of IDT $< 10ms$ grow from 59.3% in the case of 4 players up to 77.8% in the case of 8 players. A similar increase can be observed for inter-arrival times.

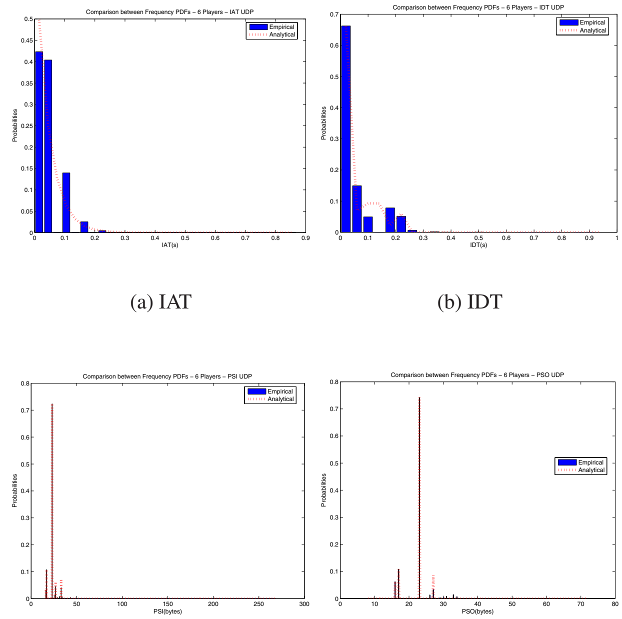
As for tail analysis, Table 2 indicates, for IAT and IDT respectively, a bad underestimation and a slight underestimation of tail. As for heavy tail estimation, refer to Figures 5(a) and 5(b) for IAT and to Figures 5(c) and 5(d) for IDT. As for IAT, the Log Log Complementary CDF shows a fast decay in final part (but it is difficult to estimate the slope because of the irregular shape of the CCDF), with an estimated slope far from -2. This is confirmed by the



(a) IAT (b) IDT

(c) PSI (d) PSO

Figure 3. Comparison between analytical and empirical CDFs



(a) IAT (b) IDT

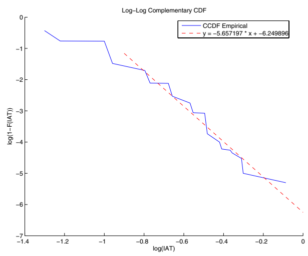
(c) PSI (d) PSO

Figure 4. Comparisons between analytical and empirical PDFs

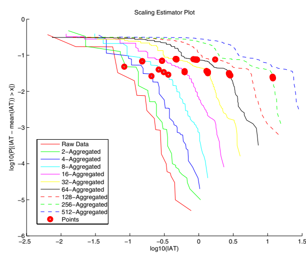
Scaling Method, that is not applicable because we have few points in the upper tail. Therefore the IAT distribution does not exhibit a heavy tail. As for IDT, the results are similar and the conclusion is the same. As regards correlation analysis, Table 2 indicates for IAT and IDT a negative $ACF(1)$ near -0.15. We have obtained close results for the 4 and 8 players scenarios. As regards auto-correlation plots of inter-packet times, we found different values from those regarding client-server games reported in [11], where $ACF(1)$ ranges from 0.3 to 0.8. Finally, in Figure 7 the auto-correlation plots from lag 1 to lag 100 for all modeled variables are shown. We note that for IAT and IDT the auto-correlation decays very slowly without reaching zero. This behaviour indicates a not summable auto-correlation function and thus the presence of Long Range Dependence. On the contrary the auto-correlation functions of packet sizes rapidly decrease to zero.

In Figure 8, the wavelet spectrum, with the estimation of the Hurst parameter, of IAT, IDT, and their corresponding packet rates are shown. It is clear that inter-packet time and packet rate are directly connected. Indeed, we evaluated the presence of self-similarity in both, and we found that the estimated Hurst parameters for the examined sequences of inter-packet times were close to the ones estimated for their respective packet rates. Moreover, we found non significative estimates of the Hurst parameter, slightly above 0.5, for inbound traffic, whereas both outbound packet rate and IDT present an estimated Hurst parameter above 0.7. This confirms a burstier nature of outbound traffic, which is also preserved over different time scales.

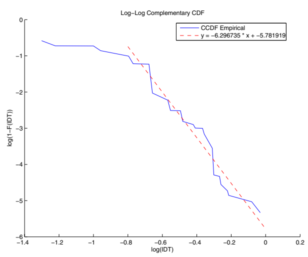
From the results related to the four studied variables, observed in different scenarios, we can conclude that Starcraft generates very small UDP packets with strong periodicity. Also, the uplink traffic produced by a single player has an higher bursty nature when compared to the downlink. Such properties can have a significant negative impact on routers found in current networks, which are not designed for this type of traffic, being more tuned against bulk data transfers using large TCP segments. Router designers indeed often make packet size assumptions, expecting average sizes around 400 bytes. As stated in [8] the explosion of multiplayer online games could result in a significant shift in packet size which could make the route lookup function in routers the bottleneck versus the link speed, leading to possible packet-loss and increased packet-delay in routers not designed to efficiently handle small packets.



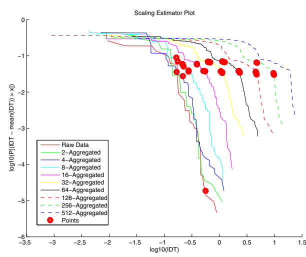
(a) Log-Log Complementary CDF Plot



(b) Scaling Method Plot

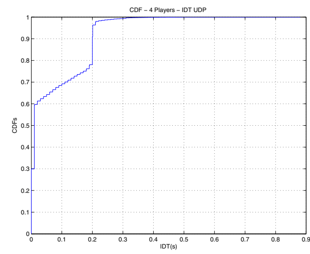


(c) Log-Log Complementary CDF Plot

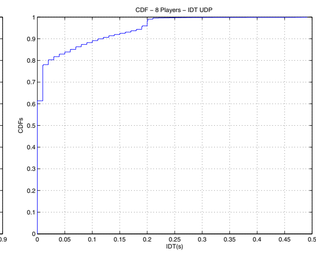


(d) Scaling Method Plot

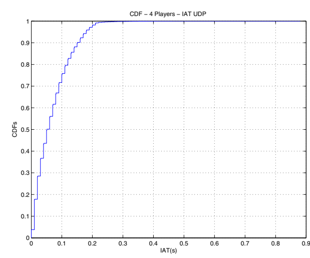
Figure 5. Tail Analysis for IAT (a-b) and IDT (c-d)



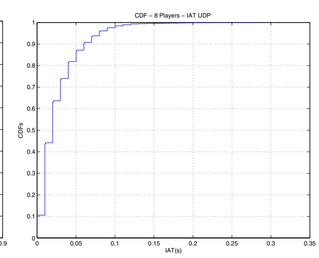
(a) IDT 4-players scenario



(b) IDT 8-players scenario



(c) IAT 4-players scenario



(d) IAT 8-players scenario

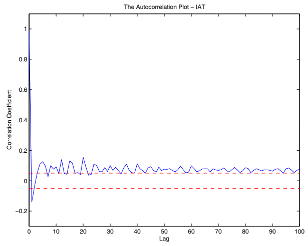
Figure 6. CDF of IDT and IAT for different numbers of players

6. Comparison with other Starcraft models

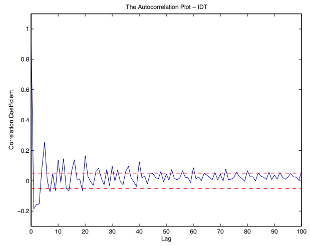
Because there are two previous works in which some characteristics of network traffic generated by Starcraft have been studied, in this Section we briefly highlight the differences with the present work. In [27] a study based on collected data in a LAN environment from a commercial Australian *Internet cafè* is presented. The communication protocol among players was IPX, not UDP/IP. Several scenarios with a different number of players were analyzed and the distributions of packet sizes and of time distance between subsequent packets have been studied, reporting the corresponding empirical CDFs. Measurements were made with a resolution of 10*ms* and packets were not divided into inbound and outbound, but the traffic generated by a single player was studied as a whole.

In [26] the WAN traffic traces that we analyzed in our study were used for the first time. Also in this work it was made no distinction between inbound and outbound packets. In the same work the traffic generated by a client-server game was studied to be compared to the one produced by Starcraft, which instead is based on a Peer-to-Peer paradigm. Traffic was studied in terms of bandwidth and, about packet analysis, inter-packet times and full Ethernet frame sizes have been analyzed showing the corresponding CDFs and comparing the results obtained for scenarios with a different number of players. Both works are basically in accordance with our findings when they state that packet sizes are small and that inter-packet times become smaller when the number of players increase, but [26] analyzes a very different scenario: a local area network running IPX, while in [26] packet sizes reported by the authors are not consistent with the traces. Indeed we found considerably smaller packet sizes then them.

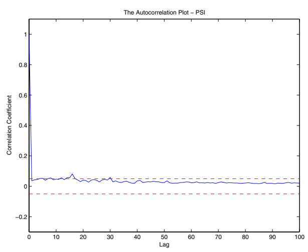
As regards our modeling approach, by splitting the traffic generated by a single player into inbound and outbound we separately characterized in detail traffic in both link directions, highlighting differences and similarities, and also allowed us to make considerations linked to the Peer-to-Peer communication structure used by the game. We reported the empirical distributions also in terms of PDFs, not only CDFs, and found analytical models approximating them through the λ^2 discrepancy measure. We studied the tail behavior of the IAT and IDT distributions and explicitly evaluated the goodness of fit for the tails. Finally we reported the auto-correlation function at lag 1 and the auto-correlation plots for all variables, and, by investigating the presence of self-similarity in inter-packet times and packet rates, we studied traffic characteristics at different time scales.



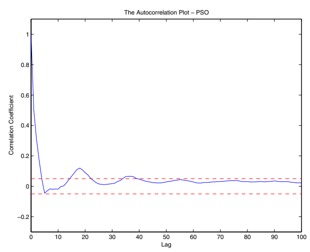
(a) IAT



(b) IDT

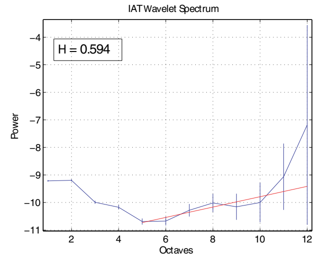


(c) PSI

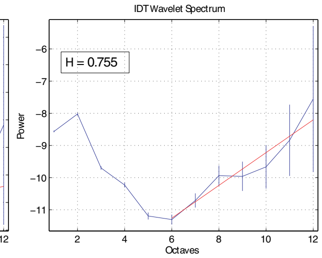


(d) PSO

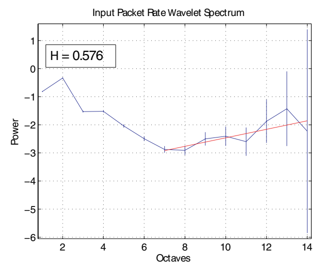
Figure 7. Autocorrelation plots



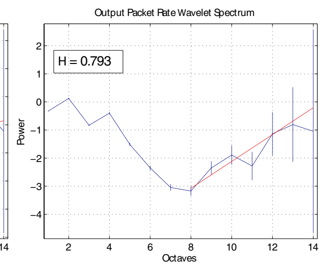
(a) IAT



(b) IDT



(c) Packet Rate In



(d) Packet Rate Out

Figure 8. Wavelet estimation of the Hurst parameter

7. Conclusions

We performed a statistical analysis of the traffic generated by a popular Real Time Strategy game, Starcraft, based on a Peer-to-Peer communication architecture. The analysis has been performed at "packet-level", that is, studying distributions of UDP payload size and time distance between subsequent packets, dividing traffic generated by a single player into inbound and outbound traffic. We found that Starcraft generates very small packets with high periodicity and that outbound traffic is more bursty than inbound. This has an impact on the routing infrastructure, which is tuned for a different kind of traffic, more bulky and with large TCP segments. Analysis also showed that packet size distribution does not change when the number of players increases, while inter-packet times tend to become smaller. We studied the autocorrelation function of packet-level variables, finding that inter-packet times tend to be correlated on a long range, and we investigated the presence of self-similarity in the traffic. We also developed analytical models of the observed traffic that could be easily applied in traffic simulation and emulation. Indeed a future work will probably be to include such models in D-ITG [29], a traffic generator developed at our university department. Of course the models reported in this work cannot be considered definitive before analyzing other traffic traces. Finally, the results of our work can be used to design networks that support traffic generated by Peer-to-Peer multi-player network games more efficiently. As regards future works, we plan to apply the same statistical methodology to study network traffic as it is generated by other Peer-to-Peer games and applications. Indeed, we are currently working on the analysis of the traffic generated by "Age of Mythology", another network game with a Peer-to-Peer communication model, and preliminary results seem to confirm some of the results shown in this work.

Acknowledgment

This work has been partially supported by the Italian Ministry for Education, University and Research (MIUR) in the framework of the WEB-MINDS FIRB Project. The authors would like to thank Giuseppe Galeone for his relevant contribution to this work and Alessio Botta, Donato Emma, and Salvatore Guadagno for their support and their valuable hints.

References

[1] S. McCreary and k. Claffy, "Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange". 13th ITC Specialist Seminar on Measurement and Modeling of IP Traffic, September 2000, pp. 1-11.

[2] D.S. Jackson, "Video Games: A room Full of Doom". Time Magazine (US Edition), vol. 153, n. 20 May 24, 1999.

[3] Peter Lincroft, "The Internet Sucks: Or, What I Learned Coding X-Wing vs. Tie Fighter". Gamasutra (Online Game Developer Magazine), September 1999. http://www.gamasutra.com/features/19990903/lincroft_01.htm
[4] Bettner P., Terrano M. "1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond". Game Developers Conference 2001. http://www.gdconf.com/archives/proceedings/2001/prog_papers.html
[5] Ng Yu-Sheng, "Designing Fast-Action Games for the Internet". Gamasutra (Online Game Developer Magazine), May 1997, http://www.gamasutra.com/features/19970905/ng_01.htm
[6] L. Gautier and C. Diot, "MiMaze, a Multiuser Game on the Internet". INRIA research report reference: RR-3248, September 1997.
[7] J. Farber, "Network game traffic modelling". 1st ACM workshop on Network and System Support for games, April 2002.
[8] W. Feng, F. Chang, W. Feng, J. Walpole, "Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server". SIGCOMM Internet Measurement Workshop, November 2002.
[9] F. Chang, Wu-chang Feng, "Modeling Player Session Times of On-line Games". ACM NetGames '03, pp. 23-26.
[10] T. Lang, G.J. Armitage, P. Branch, H. Choo, "A Synthetic Traffic Model for Half-Life". Australian Telecommunications Networks & Applications Conference 2003, Melbourne (ATNAC 2003), Australia, December 2003.
[11] J. Lakkakorpi, A. Heiner, J. Ruutu, "Measurement and Characterization of Internet Gaming Traffic". Helsinki University of Technology, Networking Laboratory, Espoo, Finland, February 2002.
[12] C. Heyaime-Duvergé, V. K. Prabhu, "Modeling Action and Strategy Internet-Games Traffic". IEEE VTC 2002.
[13] T. Lang, G.J. Armitage, "A Ns2 model for the Xbox System Link game Halo". Australian Telecommunications Networks & Applications Conference 2003 (ATNAC 2003), Melbourne, Australia, December 2003.
[14] John C. McEachen, II, "A self-similarity traffic analysis of an internet-based multiplayer online game", ACM SIGCOMM 2004 workshops on NetGames '04. p. 170.
[15] R. A. Bangun, E. Dutkiewicz, "Modelling Multi-Player Games Traffic". The International Conference on Information Technology: Coding and Computing (ITCC'00).
[16] http://www.theesa.com/pressroom.html
[17] http://nile.wpi.edu/downloads/#net-game
[18] V. Paxson, "Empirically-Derived Analytical Models of Wide-Area TCP Connections". IEEE/ACM Transactions on Networking, Vol. 2, No. 4, pp. 316-336, Aug. 1994.
[19] M. Borella, "Source Models of Network Game Traffic". Computer Communications, vol. 23, no. 4, pp. 403-410, February 2000.
[20] S. Pederson and M. Johnson, "Estimating Model Discrepancy". Technometrics, 32(3), August, 1990, pp. 305-314.
[21] V. Paxson, "Measurements and Analysis of End-To-End Internet Dynamics". Ph.D. Thesis, UC Berkeley, Jun. 1997.
[22] A. Mukherjee, "On the Dynamics and Significance of Low-Frequency Components of Internet Load". Internetworking: Research and Experience, Vol. 5, pp. 163-204, 1994.
[23] M. Crovella, M.S. Taqqu, "Estimating the Heavy Tail Index from Scaling Properties". In Methodology and Computing in Applied Probability, Vol 1 No. 1 (1999).
[24] M. Crovella, "Network Traffic Modeling". Tutorial at SIGCOMM 2004.
[25] T. Karagiannis, M. Molle, M, Faloutsos, "Long Range Dependence Ten Years Of Internet Traffic Modeling". IEEE Internet Computing, September-October 2004.
[26] Mark Claypool, David LaPoint, and Josh Winslow, "Network Analysis of Counter-strike and Starcraft". 22nd IEEE International Performance, Computing, and Communications Conference (IPCCC), Phoenix, Arizona, USA, April 2003.
[27] R.A. Bangun, E. Dutkiewicz, G.J. Anido, "An Analysis of Multi-Player Network Games Traffic". 9th ACM international conference on Multimedia, Ottawa, Canada, 2001.
[28] http://www.battle.net/starcraft-universe.shtml
[29] http://www.grid.unina.it/software/ITG
[30] K. Tutschku, "A Measurement-based Traffic Profile of the eDonkey Filesharing Service". 5th Passive and Active Measurement Workshop (PAM2004), Antibes Juan-les-Pins, France. April 19-20., April 2004
[31] N.B. Azzouna, F. Guillemin, "Impact of peer-to-peer applications on wide area network traffic: an experimental approach". Globecom'2004, Dallas, TX, November 29- December 3, 2004
[32] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, M. Faloutsos, "File-sharing in the Internet: A characterization of P2P traffic in the backbone" Technical report. (November 2003)
[33] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic". IEEE/ACM Transactions on Networking, Vol. 2, No. 1, pp. 1- 15, Jan. 1994.
[34] M. E. Crovella, A. Bestavros, "Self-Similarity in the World-Wide Web: Evidence and Possible Causes". IEEE/ACM Transactions on Networking, Vol. 5, No. 6, pp. 835-846, Dec. 1997.
[35] A. Erramilli, O. Narayan, and W. Willinger. "Experimental queueing analysis with long-range dependent packet traffic". IEEE/ACM Transactions on Networking 4:209–223, Apr. 1996.
[36] P. Abry and D. Veitch, "Wavelet Analysis of Long-Range-Dependent Traffic". IEEE Trans. Inform. Theory, vol. 4, no. 1, pp. 2–15, 1998.