SELF-SIMILAR TRAFFIC IN A COMMERCIAL VIDEO-ON-DEMAND SYSTEM

David Rincón, Glòria Ferrer, Xavier Hernández
Dept. of Applied Mathematics and Telematics, Technical University of Catalunya (UPC)
Mòdul C3, Campus Nord UPC, Jordi Girona 1-3, 08034 Barcelona, Spain
e-mail: drincon@mat.upc.es

ABSTRACT[1]

Several studies have shown that traffic carried by wide area networks presents self-similar (fractal) behaviour, which can severely impact system performance. This paper analyses the statistical properties of traffic generated by a commercial video-on-demand (VoD) system. Some parameters are measured and analysed, being the most important the packet arrival distribution and its correlation, and the degree of self-similarity found in the traces. Two tests with different client profiles (representing typical load scenarios for VoD services) have been performed. From the analysis of the traces we conclude that a high degree of self-similarity can appear in the server local area network, and that its intensity is directly related to the client profile, being the most typical scenario also the one with higher self-similarity. Our conclusion is that the impact of self-similarity on video distribution systems over the Internet is important and must be considered.

1. INTRODUCTION

The Internet is a highly complex system capable of carrying a great amount of traffic, generated by "classical" services (such as FTP, Telnet, web, e-mail, etc) and "new" applications. Some examples of these innovative applications are electronic commerce, collaborative tools for tele-education and tele-medicine, video on demand, and videoconference. This kind of applications usually includes transmission of audio and video streams (both live and/or previously stored). Among the possible generators of real time multimedia traffic on the net we can find on-line shops (promotion), webcams, Internet telephony applications (NetMeeting, CUSeeMe), or network radios (with the help of free servers from Real Networks and Microsoft). And this is only the beginning: audio and video streams are expected to be the most important traffic sources on the Internet in short time.

Multimedia streaming transmissions are known to be great resource consumers (in a network, resources are mainly link bandwidth, nodes' computation time and buffer size). Therefore, introduction of new services must be planned carefully, from both the hardware and network points of view. Aggregation of several video streams may overload links, break servers down, or cause congestion at routers.

In order to plan the deployment of high-scale audio-visual streaming services, traffic models are essential. Good, accurate descriptions of the behaviour of the connections, in terms of bandwidth and burstiness, are needed by traffic engineers to properly dimension the service. Recent studies show that Poissonian or Markovian models, which are the classical ways of characterising data sources, do not correctly describe network traffic. Better, more accurate models are based on **long-range dependence** (LRD) and **self-similarity** properties, which are related to fractals. Self-similarity and LRD properties have been found in almost every Internet service, and are especially intense in video streaming. It has been shown that variable bit-rate coding can contribute in a high degree to the self-similar characteristics of a video stream, and that aggregation of many streams results in an increment of LRD.

This paper focuses on the traffic generated by a commercial video-on-demand system, very similar to those mentioned before. To our knowledge, no study about the aggregation of low-bandwidth video sources has been performed until today (work has been done on high-quality MPEG and MJPEG streams). We

[1] This work has been supported by Spanish government under contract TIC1998-0495-C02-01

want to determine if self-similar characteristics are present in such systems.

The rest of the paper is organised as follows. First, a brief introduction to traffic modelling is presented, followed by the essential points to understand long-range dependence and self-similarity. Then some details about self-similar analysis and parameter estimation will be given, finishing the mathematical background. A description of the test scenario will follow, with analysis of the data obtained in the measurements, comparing them to what we intuitively expected to happen. The paper ends with some conclusions and open research lines for the future.

2. TRAFFIC MODELLING AND ITS IMPORTANCE FOR MULTIMEDIA STREAMING

Internet, as any other packet-switching network, relies on the concept of statistical multiplexing. This technique is based on the aggregation of several data streams from different users, which can be served with much less resources than those needed if all users were considered one-by-one. In order to optimise resource allocation and take advantage of statistical multiplexing, network engineers need mathematical expressions that describe accurately the statistic behaviour of users. These expressions are called **traffic models** [7] and have been used for a long time, since the early studies of Erlang for telephone traffic. Queue analysis applied to buffer dimensioning on IP routers is another example of the importance of having accurate traffic models. With the coming of multimedia applications, the need for good models has increased. For example, an accurate traffic model that describes precisely the temporal evolution of bandwidth required for an IP telephony application would be very helpful for network designers.

Multimedia streams have an important impact on network performance, due to their higher quality of service (QoS) requirements in terms of bandwidth, absolute delay and delay variation (also known as *jitter*). Video streaming not only demands a high bandwidth[2], but also is very restrictive on the *jitter*. These requirements are even more critic in the surroundings of the video source, since the server local network becomes a bottleneck that can be overloaded if a great number of users try to download the streams simultaneously (as will happen if we want to offer a massive video distribution service). An incorrect dimensioning of the server machine and network facilities may result in a poor service: large delays, high packet loss, and an unacceptable overall quality. What is more: if we underestimate the resource consumption for the streaming service, the rest of our traffic can suffer severe congestion. Therefore, traffic modelling is an essential issue in network design and performance.

Traffic modelling of data networks has usually assumed some characteristics that make the analysis easier: client statistical independence, infinite number of sources, exponential inter-arrival time, etc [1]. That is why Poisson and Markov models are so widely used. These models provide closed, elegant mathematical expressions for aggregated bandwidth, queue delay and other important parameters. However, recent studies [2,3,7,12] have shown that reality is not so easy: Markov- and Poisson-derived models work properly if applied to telephone calls, but not so well for data traffic.

3. LONG-RANGE DEPENDENCE AND SELF-SIMILARITY

Leland et al. [3] showed that the use of models based on Poisson lead to an underestimation of traffic burstiness on Ethernet local area networks, and proposed a new family of models, with fractal properties. This was one of the firsts of a large series of papers dedicated to self-similar processes and its suitability for network traffic modelling. Further studies proved that although Poisson is still valid for modelling the arrival of user sessions (i.e. TELNET connections, FTP control connections [2]), for both local-area and wide area network traffic the distribution of interarrival time and packet rate clearly differs from the classic models [3]. Analyses of real traffic traces showed that auto-correlation of both temporal series (interarrival time and packet rate) do not decay to zero as fast as Poisson and Markov-based models. This means that there is a high relationship between the samples of the process even at high time lags. This property is known as "long-range dependence", and can be mathematically defined in terms of the summability of the auto-correlation function.

Definition 1. *A process is considered long-range dependent if its auto-correlation r(k) is <u>not sumable</u>.*

$$\sum_{k=-\infty}^{\infty} r(k) \to \infty \qquad (1)$$

On the other hand, classic models are considered as short-range dependent (SRD) stochastic processes, with a sumable auto-correlation function.

$$\sum_{k=-\infty}^{\infty} r(k) < \infty \qquad (2)$$

2 from 30-50 Kbit/s for low-quality home transmissions, to several Mbit/s for high-quality videoconference.

Long-range dependence is tightly related with burstiness. LRD traffic sources tend to generate large bursts of information followed by periods of silence, with a highly variable duration (some models assume infinite variance).

The generation of LRD network traffic models has been an intense research field during last years. The most used are called self-similar models [2-4]. These models are able to detect LRD, and can be characterised with a single parameter: the Hurst parameter[3], H. Self-similarity and LRD have often been mentioned in the literature as equivalent terms, but they are not. Self-similar processes are not the only models capable of capturing LRD behaviour [2,7], and as we will see not all self-similar processes (in the strictest definition) present LRD.

3.1. Self-similar processes: definition

There are a number of different, not equivalent, definitions of self-similarity. The standard and more generic one states that a continuous-time process $Y=\{Y(t), t\in T\}$ is *self-similar* (with self-similarity parameter H) if it satisfies the condition

$$Y(t) \stackrel{d}{=} a^{-H} Y(at), \quad \forall t \in T, \quad \forall a > 0, \quad 0 \le H < 1 \quad (3)$$

where the "d-equality" means "both terms have the same finite-dimensional distributions" [4]. The idea behind the definition is that a self-similar process "appears" to be the same (in statistical terms) at any temporal scale. It is the same if we study it at microseconds or hours; it will always look similar. The value of H for a self-similar process varies between 0.5 (no self-similarity at all) and 1.0 (highest self-similarity).

The concept of self-similarity in (3) and LRD are not equivalent. A process Y satisfying (3) can never be stationary, thus cannot present LRD (auto-correlation function can only be calculated for stationary processes). If stationarity of increments of the process is assumed, then the process can present LRD. Self-similarity defined as (3) refers to the behaviour of the finite dimension distributions in different time scales, while LRD is related to the behaviour of auto-correlation functions in stationary processes.

3.2. Exact and asymptotic self-similarity

The next vision of self-similarity, more appropriate in the context of standard time series theory, involves a stationary sequence $X=\{X(i), i \ge 1\}$. Let $X^{(m)}(k)$,

$$X^{(m)}(k) = 1/m\left[\sum_{i=(k-1)m+1}^{km} X(i)\right] \qquad \text{with } k=1,2,... \qquad (4)$$

be the aggregated sequence corresponding to the level of aggregation m, obtained by dividing the original series $X(i)$ into non-overlapping blocks of size m and averaging over each block. The index, k, identifies the block.

If $X(i)$ is the increment process of a self-similar process $Y(i)$ defined as (3), (i.e. $X(i)=Y(i+1)-Y(i)$), then if for all integers m,

$$X(i) \stackrel{d}{=} m^{1-H} X^{(m)}(i) \qquad (5)$$

Definition 2. *A stationary sequence $X(i)$, $i\ge 1$ is exactly self-similar if satisfies (5).*

Definition 3. *A stationary sequence $X(i)$, $i\ge 1$ is called asymptotically self-similar if (5) holds as $m\to\infty$.*

Similarly, we call a covariance-stationary sequence $X(i)$, with $i\ge 1$, *exactly second-order self-similar* or *asymptotically second-order self-similar* if $m^{1-H} X^{(m)}$ presents the same variance and auto-correlation as X, for all m, or as $m\to\infty$ [3-6]. Although these definitions are not in general equivalents is interesting to note that when the distribution of the series is gaussian, the definitions of self-similarity and second-order self-similarity are equivalents.

3.3. Properties of self-similar processes

For a process $X(i)$ *exactly or asymptotically second-order self-similar*:

- $Var(X^{(m)}) = \sigma^2 m^{-\beta}$, $m \ge 1$, implying LRD in their aggregated process, with $H = 1-\beta/2$.

- $r^{(m)}(k) = r(k)$ for $k \ge 0$ (exactly) or, $r^{(m)}(k) \to r(k)$, as $k\to\infty$ (asymptotically), preserving the correlation structure for different time scales.

Summarising, a process $X(i)$ is *exactly or asymptotically second-order self-similar* if the aggregated processes $X^{(m)}$ are equal to $X(i)$ or asymptotically indistinguishable, at least considering the auto-correlation function.

We have said that self-similarity on (3) and LRD are not equivalent, but these new definitions (especially second order self-similarity, restricted to the behaviour of auto-correlation function) we can use either LRD or self-similarity indistinctly when

[3] Named after H.E. Hurst, who first applied self-similar models to hydrology.

analysing our real traffic data traces. We only have to be sure that series under study follow gaussian distributions (a usual property).

4. ESTIMATION OF SELF-SIMILARITY

As we have seen before, the fact that self-similar processes are described with a single parameter, (Hurst parameter H), is one of their most interesting characteristics. There are many H estimators [4,8], most of which assume second-order self-similarity on the process. We can distinguish two classes of estimators, the ones based in the periodogram and maximum likelihood-type estimators; and the graphical estimators, that exploit LRD properties. The first ones has been shown to have desirable statistical properties, and can also provide confidence intervals for the self-similar parameter H, but require a known probability density function (PDF) [8]. Graphical estimators calculate H by fitting a simple least-squares line through different kind of plots. This makes them inadequate for a more refined data analysis (e.g. confidence intervals), but allow us to work with unknown processes with a difficult to express or unknown PDF.

For the study presented in this paper two different estimators were used (after a literature scan), in order to ensure correctness of the analysis.

4.1. Scaling of moments

Suggested in [4], this method is based on the behaviour of the absolute moments $\mu^{(m)}(q)$ of the aggregated time series.

$$\mu^{(m)}(q) := E\left|X^{(m)}\right|^{q} = E\left|\frac{1}{m}\sum_{i=1}^{m} X(i)\right|^{q} \qquad (6)$$

If X is self-similar, $\mu^{(m)}(q)$ is proportional to $m^{\beta(q)}$; that is, $\log \mu^{(m)}(q)$ is linear in $\log m$. For a fixed q:

$$\log \mu^{(m)}(q) = \beta(q) \quad \log m + C(q). \qquad (7)$$

The exponent $\beta(q)$ is linear with q and H. Therefore, regression on the logarithm of the sample absolute moments will derive a estimation of H through β.

4.2. Residuals of regression

This method involves several steps. It is based in the sample variance of the residuals obtained by fitting a least-squares line to the partial sums within every block the series is broken. It has been demonstrated that for large aggregation level, the resulting number is proportional to m^{2H} for Fractional Gaussian Noise [8]. Thus if the result is plotted on a log-log plot versus m, we should get a straight line with a slope of $2H$, and H will be easily estimated.

4.3. Validation

We have validated our own implementations of the algorithms against synthetic traces (generated with a known H) and real data (comparing results of other works).

5. TESTBED DESCRIPTION

The analysis presented in this paper has been performed with the help of Microsoft Windows Media Load Simulator. Windows Media (WM) is a suite of tools designed to provide audio and video on demand over best-effort IP networks [11]. Among the members of the family, we can find WM Encoder, WM Server, and WM Load Simulator. The audio and video streams are stored in files previously coded, or generated in real time, by the Encoder. In both cases the Server is the responsible of managing clients' requests, generating RTP-style streams, and monitoring the transmission.

WM Load Simulator (LS) emulates the behaviour of several clients accessing a WM Server for playing or browsing video streams. The tool was developed for giving to server's administrators an estimation of the capabilities of their machines: CPU, I/O and storage system characteristics. LS stresses the machines to the point of breakdown, in order to determine the maximum sustainable load. This information allows the administrators to dimension appropriately their hardware for the video-on-demand service. From the traffic engineering point of view, the Load Simulator is interesting because each simulated client opens a connection to a Windows Media Server, which serves the requests with real traffic. This allows us to capture and analyse an aggregated stream of real traffic and study its statistical properties (self-similar behaviour).

The profile of clients' activity against the server is configurable by Load Simulator's user. For example, some clients play the movies from the beginning to the end, while others seek forward and backward for specific scenes within the film. The client may simply open the file, decide that is not interesting, and close it. WM Load Simulator allows the definition of profiles that contain different quantities of each client class. For our traffic study, we have considered the following different categories of clients:

- *Play*: users who play entire streams from the beginning of the film to the end. The traffic generated in this case will consist in constant bit rate bursts of approximately the same size, which can be periodic (since they open another file as soon as the previous is finished). Intuitively, we can foresee that this kind of behaviour will not show self-similar properties ($H \approx 0.5$).

- *Browse*: these simulate "browsing users". They play for random lengths of time, seek, stop, pause, and sometimes close the files. Therefore, they will generate a very bursty traffic that will appear as an aggregation of runs of random duration, adding self-similarity to the traces (with a high H, close to 1.0).
- *Seek*. These ones attempt to seek forward and backward in a stream. The traffic generated in this case will also be bursty, but not as much as "browse clients", showing less self-similar properties (a lower Hurst parameter, between 0.5 and 1.0).

Figure 1 shows the test scenario. Two Windows NT computers were used as WM Server and Load Simulator, while a third one (Linux powered) captured the packet stream with `tcpdump`. The traces were later studied with our self-similar analysers.

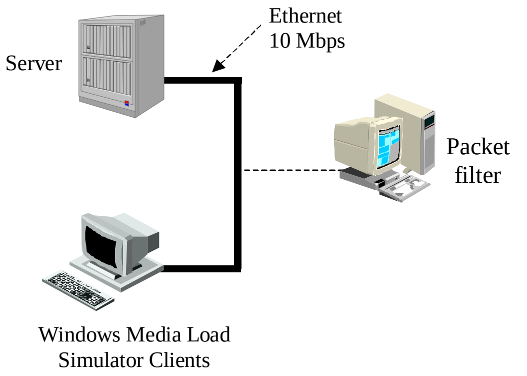

Figure 1. Testbed description.

Our main aim was to measure the influence of client profile in the self-similar properties of the aggregated traffic. In our analysis, we tried to simulate traces that could represent extremes of self-similar behaviour: H≈0.5 and H≈1.0. Table 1 shows trace profiles.

Clients' class	Trace 1	Trace 2
Play	30	60
Browse	10	3
Seek	10	3

Table 1. Client activity profiles for the traces.

We expect to have very different statistical and self-similar properties for both traces. In trace 1, although the difference between *Browse*/*Seek* clients and Players is not so high (20%-20%-60%), the traffic generated by this profile is very bursty and it was chosen for study because of its probable self-similar properties. On the other hand, trace 2 offers a scenario where almost all the clients belong to the *Play* class. That is why the aggregated traffic shows a higher regularity, which goes against self-similarity. Table 2 shows general data from the traces.

	Trace 1	Trace 2
Start time	12:44h	15:25h
Stop time	14:29h	16:25h
Total time	1h 45min	1h 0min
Total data exchanged	1.5 Gbytes	934 Mbytes

Table 2. Summary of data from the traces.

Any interested reader may have noticed that an essential piece of information is missing: the distribution of the command generation process for browse and seek clients. Actually, this is the source of burstiness and its (probable) associated self-similarity. The authors have searched for it in Microsoft, unsuccessfully.

6. RESULTS

As usual, our first study of the traces is just a visual inspection. Figures 2 and 3 depict the temporal evolution of the number of packets per second generated for both traces.

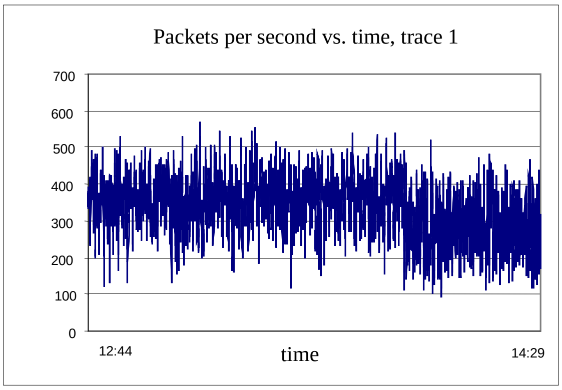

Figure 2. Trace 1.

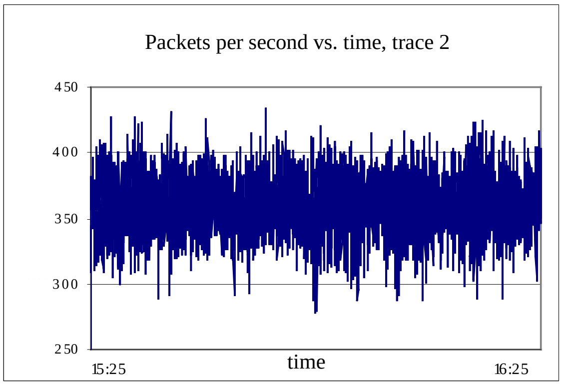

Figure 3. Trace 2.

First, notice the similar average packet generation rate (approximately 360 packets/second), which validates our election of profiles. This is not a trivial detail. Traces with different means should not be compared. That is why the total number of clients in both profiles is different, since *non-play* clients generate more traffic than *play* ones. The main difference between both plots is the higher burstiness[4] in trace 1. The second one has a more compact and stable shape, with much less variance, as expected.

6.1. Visual proof of self-similarity

The best way of determining if a trace is a good candidate for self-similarity is the "visual proof", that consists on the comparison of the data at different time scales [3].

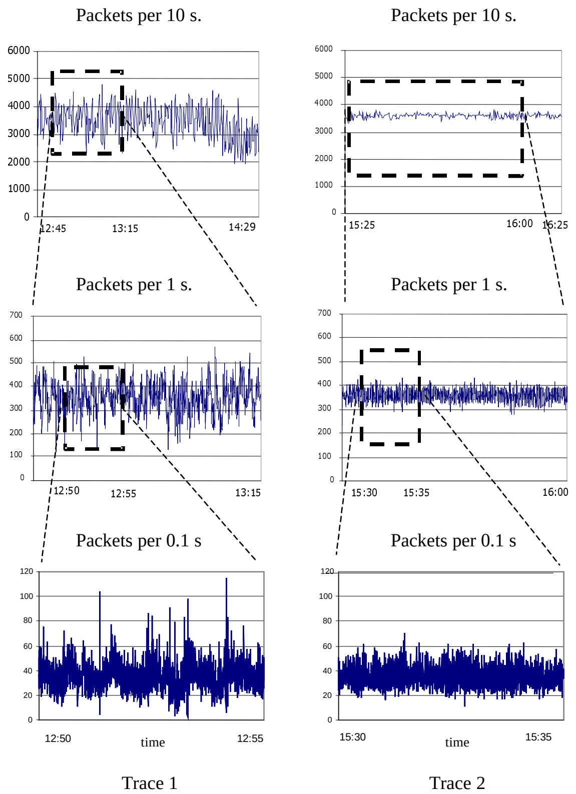

Figure 4. Visual proof of self-similarity.

As can be seen from figure 4, trace 1 appears similar at any scale, while trace 2 tends to smooth as the aggregation factor grows. Therefore, we can conclude that trace 1 will probably have a high H parameter, while H for trace 2 will be approximately 0.5 (no self-similar at all).

[4] Note the different scales in figures 2 and 3. Trace 2 is "narrower" than trace 1.

6.2. Analysis of packet arrival rate distribution

Now we will examine the empirical probability distribution of packet arrival rate. It is important in order to validate our Hurst parameter measurements. In a previous section, we saw that some self-similarity analysers only perform well if the traces follow the normal (gaussian) distribution.

Figures 5 and 6 show histograms for trace 1 and 2. As we can see, both traces tend to a gaussian distribution. Actually, the fit appears to be good except for some pronounced deviation, especially intense in the first histogram. Although these deviations, the traces were assumed as compliant with the requirements of the analysers.

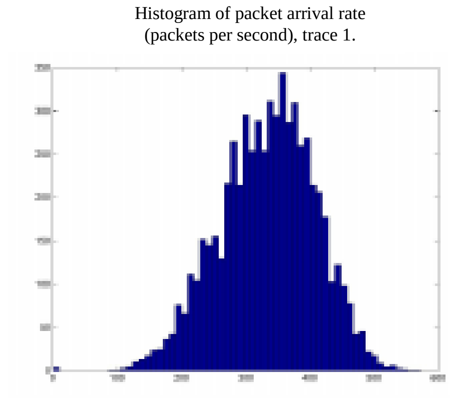

Figure 5. Histogram, trace 1.

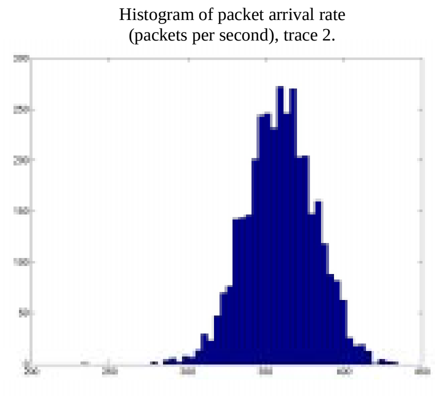

Figure 6. Histogram, trace 2.

The main difference between both plots is that the second histogram has a narrower curve (less variance) due to its client activity profile. As we have seen in figure 3, samples from Trace 2 tend to the mean value without too much variability. This behaviour is translated into the histogram as a much narrower curve around the mean, which confirms our first assumption about the higher burstiness of the first trace.

6.3. Analysis of packet arrival rate correlation

Next, we studied the correlation structure of packet arrival rate for both traces. Assuming that the number of packets per second X is a random process, with sample mean $\overline{X}$ and sample variance S^2, then the auto-correlation function can be estimated for all lag k as in (8) [12].

Figure 8 shows the auto-correlation *r(k)* of the packet arrival plotted as a function of the lag *k*, for both traces.

$$r(k) = \frac{\sum_{i=1}^{N-k}(X_i - \overline{X})(X_{i+k} - \overline{X})}{(N-k)S^2} \qquad (8)$$

The figure shows how auto-correlation structure for trace 2 decays faster than the other, which even for high *k* values it remains over zero. Trace 2 tends to zero very quickly, while trace 1 decays very slowly and, in fact, does not converge to zero This behaviour suggests that trace 1 is long-range dependent and trace 2 is short-range dependent. However, we also have to consider that these traces are real ones so they probably might have mixed traffic with SRD and LRD properties simultaneously, although LRD is dominant in the case of trace 1.

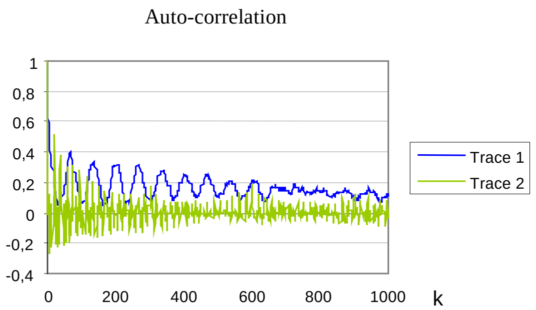

Figure 7. Auto-correlation function for packet arrival time series.

Another interesting effect is the periodicity of r(k) for both traces. Actually, we expected to have some periodic behaviour because of the simulation characteristics, due to the cyclical behaviour of *play* clients (open, play, close, wait, and begin again). The periodicity effect is more accentuated (and with a higher frequency) in the second trace because its profile contains twice the quantity of *play* clients.

6.4. Analysis of self-similarity

Now that we know the distribution and the auto-correlation structure of the traces, and having detected long-range dependence in the first one, it is time to quantify self-similarity. Table 3 includes the statistical results obtained from our analysers.

	Trace 1	**Trace 2**
H (Moments)	0,898	0,462
H (Residuals)	0,908	0.481

Table 3: Estimation of Hurst parameter.

The data obtained are coherent with our intuitive previsions, and with the previous analysis of long range dependence. Both analysis methods agree on the degree of self-similarity included in the traces. For trace 1, H≈0.9 (which means a high grade of self-similarity), while trace 2 is close to 0.5 (meaning no self-similarity at all).

7. CONCLUSIONS AND FUTURE WORK

The purpose of this paper has been to establish a relationship between the behaviour of the clients of a video-on-demand commercial system, and the self-similarity of the aggregated traffic created by their requests. Several studies have been carried in the field of single variable-bit-rate video sources characterisation, but to our knowledge nobody studied the aggregation of constant-bit-rate video streams, taking into account the behaviour of the customers. Our test included a commercial video server widely used, fact that gives generality to our study.

After analysing statistical distribution, auto-correlation, and self-similarity, we have been able to establish a relation between the client activity profile and the statistical properties of the flow. Our main conclusion is that the seeking and browsing possibilities of a real video-on-demand system can be very powerful sources of self-similarity. This effect appears in the server's connection, affecting the design of the video service provider's network, and should be considered when deploying a VoD service. Our study has been limited (by hardware availability) to about 60 clients, but in real systems (such as pay per view video, tele-education, or electronic commerce) several hundred clients or even thousands[5] are expected to aggregate at the server connection. It is known that aggregation reinforces self-similarity. Therefore, real situations will probably be burstier than results presented here (with a higher H parameter).

Being a preliminary study, some further work could be done. Our first priority is to find out the distribution of command generation from seek and browse clients. This information has not been published by Microsoft, and it is very important in

[5] Microsoft release notes for Load Simulator mentions 800-1000 users per server. Servers can join in clusters for load balancing.

order to establish the source of the LRD and self-similarity properties we have detected in our study.

The traffic profile of an authenticated system (pay per view, for example) is an unexplored issue. Our intuition tells us that the amount of traffic generated during the authentication process could be negligible if compared with the complete transmission of a movie, but what will happen if the seeking clients must also be authenticated? In this case the authentication traffic could be as important as the browse information, strengthening self-similarity (burstiness would increase).

Other open paths for study is the influence of network protocols (ARQ mechanisms, TCP error and flow control, congestion avoidance) in the aggregated traffic, as well as link layer medium access mechanisms (MAC) interaction with traffic at higher layers. It has been shown [13] that protocols may be a very important traffic shaping factor, and self-similarity could be a side effect of protocols interaction with the data sources.

Finally, it would be interesting to extend this study to audio sources and analyse aggregation of Internet telephony and radio clients (for example, NetMeeting, RealPlayer and WinAmp), and its impact on network performance. Each application has a different buffering and streaming strategy, which can add or extract self-similarity to the background traffic. Another interesting application is Napster[6], a MP3 file sharing system that is creating a huge amount of traffic on the Internet. This application is also important because, different to asymmetrical audio and video distribution systems, it allows a bidirectional and symmetrical data flow. We expect similar results like the ones we have presented in this paper, maybe with even more intense LRD and self-similarity.

ACKNOWLEDGEMENTS

The authors want to thank Rafael Vidal for his help during the test, and Sebastià Sallent for his advice.

REFERENCES

[1] V. Frost and B. Melamed, “Traffic modeling for telecommunication networks”, *IEEE Commun. Magazine,*. vol. 33, pp. 70-80, March 1994

[2] V. Paxson and S. Floyd, “Wide Area Traffic: The Failure of Poisson Modeling”, *IEEE/ACM Trans. on Networking,* vol. 3, nº 3, June 1995.

[3] W. E. Leland, M. S. Taqqu, W. Willinger and D.V. Wilson, “On the Self-Similar Nature of Ethernet Traffic (Extended Version)”, *IEEE/ACM Tranactions. on Networking,* vol. 2, pp. 1-15, February 1994.

[4] V. Teverosky, "Detection and estimation of long-range dependence". PhD Dissertation, 1997.

[5] H. F. Zhang, Y. T. Shui and O. Yang, “Estimation of Hurst Parameter by Variance-Time Plots”, *0-7803-3905-3/97 IEEE.*

[6] S. Mólnar, A. Vidács and A. Nilsson, “Bottlenecks on the Way Towards Fractal Characterisation of Network Traffic: Estimation and Interpretation of the Hurst Parameter”.

[7] A. Adas, “Traffic Models in Broadband Networks”, IEEE Communications Magazine, Vol. 35, No. 7, July 1997.

[8] M. S. Taqqu, V. Teverosky and W. Willinger, “Estimators for Long-range Dependence: an Empirical Study”, *Fractals*, vol. 3, pp. 785-788, 1995.

[9] V. Paxson, Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic. Computer Comm. Review, vol. 27 N. 5, October 1997, pp. 5-18.

[10] The Internet Traffic Archive: http://ita.ee.lbl.gov/index.html.

[11] Microsoft Windows Media Technologies, http://www.microsoft.com/windows/windowsmedia

[12] M. T.Lucas, D.E.Wrege, B.J.Dempsey i A C.Weaver, "Statistical Characterization of Wide-Area Self-Similar Network Traffic", University of Virginia Technical Report CS-96-21, October 1996.

[13] J. M. Peha, "Retransmissions mechanisms and self-similar traffic", Proceedings of teh Communication Networks and Distributed, vol. 1, pp. 47-52, 1997.

[6] http://www.napster.com