On the combined effect of self-similarity and flow control in quality of service for transactional Internet services.

Javier Aracil, Daniel Morató and Mikel Izal
Dpto. Automática y Computación
Universidad Pública de Navarra
Campus Arrosadía s/n
31006 Pamplona, Spain
Tel: +34 48 16 97 33
Fax : +34 48 16 92 81
email: {javier.aracil, daniel.morato, mikel.izal}@upna.es

Abstract
In this paper we show that the combined effect of heavy-tailedness and flow control leads to considerable transaction delays. Neither heavy-tailedness nor flow control separately imply a significant degradation in quality of service. We consider transactional Internet services such as WWW and relate user perceived quality of service to total transaction delay instead of packet or cell instantaneous delay [2]. We evaluate transaction delay by simulation of an IP over ATM link in which a large number of users are multiplexed and we compare to M/G/1 analysis. Our traffic model assumes heavy-tailed features in file sizes and a constant rate for packet interarrival times within transactions. We show that an in increase in bandwidth assignment, i.e. a decrease in link utilization factor, does not translate into a significant performance improvement. However, an increase in window size proves more effective.

Keywords
Internet service provisioning, self-similarity, TCP

1 INTRODUCTION AND PROBLEM STATEMENT

Nowadays, we are witnessing a huge demand of Internet services like the World Wide Web. Internet traffic self-similarity poses new challenges regarding bandwidth allocation, billing and pricing for Internet services. Traffic burstiness is preserved at any time scale, in contrast to short-range dependent models such as the Poisson process. Queueing analysis with self-similar input is an active research area since network dimensioning for Internet services has became a very important issue. However, performance metrics are obtained at the cell or packet level : buffer overflow probability and delay estimates under self-similar input [4, 7, 9, 10]. Buffer overflow probability and delay at the packet or cell level may not be an adequate QOS metric for service provisioning. Little literature exists on QOS metrics that relate Internet user satisfaction and network parameters such as end-to-end delay and bandwidth. David Clark addresses this issue in [2], arguing that user satisfaction grows with transaction throughput. Namely, a large instantaneous bit rate is useless unless such bit rate is mantained during the whole transaction. Since it is possible to know the file sizes before the transaction takes place bandwidth allocation can be done beforehand. If we consider transaction duration as the valid QOS metric a detailed analysis at the transaction level is needed.

Transactional services (FTP-data and WWW) represent the most important part of Internet traffic [8, 1, 3]. Pareto distributions prove accurate to model file size and transmission duration for FTP-data and WWW [8, 1, 3]. Inactivity periods of a single user turn out to be heavy-tailed as well [11]. This approach leads to an on-off model with heavy-tailed distribution to model individual users. The multiplex of a large number of users shows exponential behavior in the transaction interarrival times [1, 6]. The transaction arrival process in the busy hours can be modeled approximately as a Poisson process. Nabe et al. [6] show that Poisson arriving heavy-tailed bursts constitute an accurate traffic model for busy hours of WWW service.

Tsybakov and Georganas show in [10] that Poisson arriving heavy-tailed batches with constant cell rate within the batch lead to an asymptotically second order self-similar process. If we consider Tsybakov and Georganas model, a transaction level analysis of a multiplex of a large number of users in a single virtual circuit can be undertaken using the well-known M/G/1 or M/G/1/PS model [6]. Furthermore, the M/G/1 model provides a simple framework to explain how self-similarity affects user perceived QOS. Since file sizes are heavy-tailed the service time squared coefficient of variation (namely, variance normalized by the squared mean [5, page 187]) is large and degrades performance. Other factors such as TCP flow control also make such squared coefficient of variation increase since transaction duration increases due to source active waiting periods.

In this paper we show that the joint effect of a simple window flow control mechanism and heavy-tailed file sizes causes a significant performance drop, even in a small roundtrip delay environment. However, neither the former nor the latter separately degrade QOS in a so significant manner. Our methodology consists of simulations using a single virtual circuit model and comparison to M/G/1 analysis. We evaluate network contribution to QOS perceived in contrast to other factors such as heavy-tailed file sizes. Our findings suggest that network parameters such as window sizes may be tuned to provide a better QOS.

The rest of this paper is structured in three parts: section 2 explains our traffic model and simulation setup, section 3 presents the results and discussion and section 4 presents the conclusions that can be drawn from this study.

2 USER TRAFFIC MODEL AND SIMULATION SETUP

Our simulation setup is shown in figure 1. We consider a large population of users whose traffic is being multiplexed over the same link with a unique queue. That is the common situation for Internet Service Providers (ISPs) and corporation and academic networks: the edge router is configured with a unique constant bandwidth VP/VC to the ATM cloud and a unique inbound queue to the users. We will assumme that both queues have infinite capacity. IP packets are segmented into ATM cells but there is no cell interleaving from different IP packets.

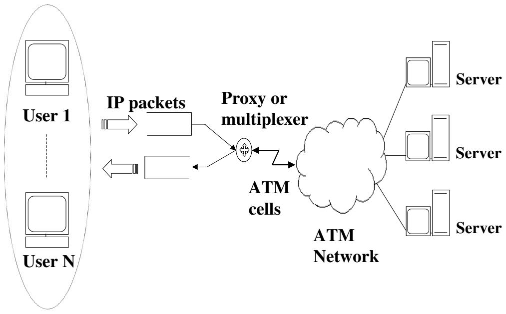

Figure 1 Simulation setup

We evaluate the input queue to the router from the ATM cloud because Internet traffic is highly asymmetric. We assumme that file transfer queries (GET commands) are issued from the users population. In response to such queries the bulk traffic stream comes in the inbound direction. Transaction duration and size (bytes) are both heavy-tailed [1, 3, 6].

Our sliding window flow control mechanism resembles TCP behavior: Each ACK packet acknowledges all transmitted packets whose sequence numbers are smaller than or equal to the sequence number announced by the ACK packet. The traffic source stops transmission whenever the negotiated window is full of unacknowledged packets. Such behavior is typical of transport layer protocols such as TCP. TCP also incorporates a congestion control algorithm known by Slow Start. At any given transmission time the minimum between the flow control window and the congestion control window is selected by the TCP protocol agent as the actual transmission window. The value of the flow control window is negotiated at the connection establishment phase (SYN PDUs). Note that the negotiated flow control window determines the maximum size of the transmission window. Even if the congestion control window eventually allows for a larger transmission window size the negotiated flow control window imposes a limit. We will show later that small window sizes imply performance penalties. Slow start makes transmission window decrease if congestion is detected.

We will assume no packet loss in the transmission link, i. e. no retransmissions. We aim at showing the influence in network performance of a simple, yet explanatory, sliding window flow control algorithm in presence of heavy-tailedness in file sizes. Our simple model represents a best-case model in comparison to TCP since for the latter the transmission window can take values smaller than the negotiated flow control window due to the congestion control algorithm, as explained before. Note that unacknowledged packets may suffer considerable queueing delays in high load situations. If the flow control window is full of unacknowledged packets such queueing delays are particularly harmful because the source gets stopped until new acknowledgments are received. Note that even though flow control is performed on an end-to-end basis a unique queue is shared by all sources. We observe two different contributions to transaction duration: queueing delay, that depends on the utilization factor and service time squared coefficient of variation (M/G/1) and flow control delay, which increases transaction delay each time an ACK is needed from the destination in order to alleviate the flow control window of unacknowledged packets.

We will assume a roundtrip delay of 0.01 s., which is a reasonable empirical value for TCP connections within a statewide network. Larger RTDs, such as the ones for overseas connections would make transaction duration increase.

Once the simulation scenario is defined we are faced with the selection of a traffic model that accurately portrays user behavior. Several studies show that an on-off model with heavy-tailed on-off periods is accurate to model a single user behavior for Internet bulk data transfer services (i.e. WWW and FTP-data) [11]. The heavy-tailed nature of the on period is mainly due to the Internet file sizes distribution while off periods are related to user think time. File sizes variance and mean depend on the media: text, images, sound or video files [6]. Considering the multiplex of different types of files in a real trace a mean value in the range of 50 KB can be adopted for WWW services [3].

Fluid-flow on-off models assumme that the time to transfer a file equals file size divided by link capacity, namely no time gaps between packet transmissions. However we do not use a fluid-flow model for activity periods since transmission time is larger than file size divided by link capacity. The operating systems and network interface cards impose limitations in the packet interarrival times so that a larger service time for a transaction is observed. In order to evaluate interpacket spacing, we perform a number of WWW transactions between a client and server stations in two different situations: dedicated LAN and departmental network. The client WWW cache is set to zero bytes so that we always enforce transmission.

We perform two different measurement experiments: the first one in a dedicated Ethernet of a SUN workstation and a PC, the second one is taken with both client and server in the departmental network in the busy hour. We perform a total of 600 transactions with file sizes ranging from 10 KB to 3 MB with a 100 KB step size. Our departmental network is not isolated by a router to the campus backbone so that we receive the traffic multiplex of approximately 900 hosts. The results show packet interarrival times in the vicinity of 1.5 ms ($P(interarrivaltime < 1.5ms) = 0.85$) so that a significant deviation from a fluid-flow behavior is not observed. A 1500 bytes (Ethernet MTU) packet transmission time is 1.2 ms for a 10 Mbps Ethernet. However, the cumulative effect for large file transmissions can be significant. In order to have a better picture of packet-level transmission we plot in figure 2 the measured transaction duration and the same transaction duration assuming a fluid-flow model and a constant rate packet transmission (1500 bytes) with interarrival times equal to 1.5 ms and 5 ms. Note that significant deviations can be observed specially with large file sizes. Therefore, we adopt the discrete model (constant packet rate within the bursts) in contrast to the fluid-flow model.

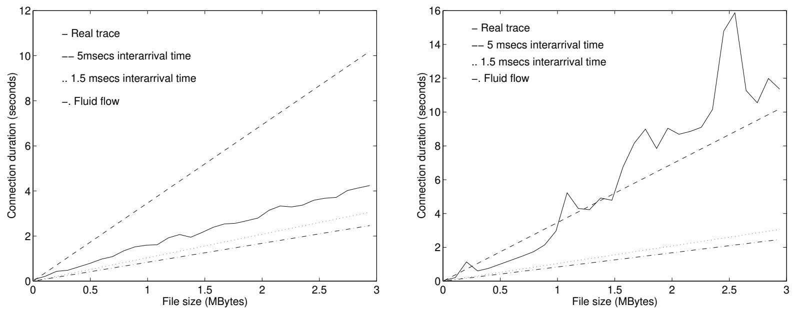

Figure 2 Transaction duration: real trace, fluid-flow model and constant packet rate model (1.5 ms and 5 ms). Right: dedicated network, left: departmental Ethernet

As far as the transaction arrival process is concerned, we assume that the multiplex of a large number number of independent arrivals converges to a

Poissonian model. However, such connection arrival process has little influence in our analysis.

As stated before, Tsybakov and Georganas show in [10] that a Poisson batch arrivals process with heavy-tailed batches and constant cell rate is an asymptotically second order self-similar process. Therefore, we are considering a self-similar model in our simulation setup.

One important factor in our analysis is the flow control window size value (Kbytes). We collect a traffic trace to determine what is the typical window size value in the TCP flow control algorithm for transactional Internet services such as FTP-data and WWW.

The trace is obtained from Public University of Navarra campus network, that consists of a high-speed backbone (FDDI) and approximately 30 departmental Ethernets. The estimated number of hosts connected is 900. The analyzed trace comprises 244,568 FTP-data and WWW connections recorded during 12 hours. Interestingly, the probability mass function shows two outcomes that dominate the sample: 8760 bytes and 31744 bytes with a probability of 0.61 and 0.33 respectively. Thus, we consider window sizes of 8 KB, 16 KB and 32 KB for our simulation experiments.

Our conclusions about the traffic model can be summarized as follows. Since we consider a large population of users we choose a Poisson transaction arrivals model in which file sizes are heavy-tailed. Furthermore, we consider that file sizes follow the Pareto law since such distribution models accurately transaction sizes (bytes) for FTP-data and WWW services [1, 3]:

$$f_X(x) = \alpha k^{\alpha} x^{-\alpha-1} \tag{1}$$

where k represents the minimum batch size, which we adjust to a value of 1000 bytes. The parameter α relates to the batch size heavy-tailedness and, ultimately, to the service time variability. A value of α in the range $1 < \alpha < 2$ would produce self-similarity features in the packet counting process [10]. We truncate the file size distribution in equation 1 to a maximum value of 10 MB. The probability of such maximum file size in a WWW or FTP transaction is around 10^{-7} [1, 3]. Such truncation permits the calculation of the variance and coefficient of variation. On the other hand, the truncated distribution resembles accurately the file sizes distribution in the Internet. Crovella et al. report a value of α for WWW transactions approximately around 1.1 [3]. In previous studies we report a value of $\alpha = 1.28$ for FTP-data transfers, considering a sample size of four days worth of IP traffic from the UC Berkeley campus network (439 Mpackets, 69 Gbytes) [1]. Finally, we assumme Ethernet rates (10 Mbps) for the capacity assigned in the inbound queue and constant size packets of 1500 bytes (MTU size).

3 RESULTS AND DISCUSSION

In this section we present simulation results and discussion. We perform three different simulation experiments with a finite horizon simulation (24 hours simulated time):

- Experiment one: we analyze the effect of **flow control with fixed size batch**. A fixed size batch is the best case regarding service time (M/D/1) so that it should provide the best performance figures as far as transaction delay is concerned. Our aim is to show that the influence of flow control in QOS is not significant if file sizes are deterministic.
- Experiment two: we replace the fixed-size batch by a **heavy-tailed batch with no flow control**. We show that the effect of heavy-tailed file sizes is not so significant for QOS if flow control is not activated.
- Experiment three: we assume **heavy-tailed file sizes and 8 KB, 16 KB and 32 KB flow control window sizes**. We show that the effect of flow control in presence of infinite variance of file sizes is dramatic. Our findings show that an increase in window size translates into a very significant performance improvement.

3.1 Fixed size file sizes and flow control

Figure 3 shows transaction delay for fixed size (50 KB) files and different values for the flow control window size (4, 16 and 32 KB). Furthermore the M/D/1 results are also shown for comparison purposes.

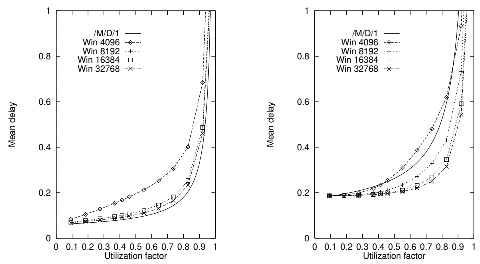

Figure 3 Fixed-size file sizes. Right: 5 ms packet interarrival times, left: 1.5 ms packet interarrival times

Let ρ be the utilization factor, B the file size in bits and C the link capacity in bps. Let λ be the joint arrival rate of the multiplex of users and let the utilization factor be $\rho = \lambda B/C$.

It is important to note that the time to transfer a file is not equal to file size divided by link capacity. Such time can be computed as the total time to transfer a file taking into account real packet interarrival time (see figure 2). Let t_p be the packet interarrival time (1.5 ms or 5 ms in figure 3) and M the MTU size in bytes (1500 bytes) then the time to transfer a file $\overline{x}$ equals $\overline{x} = (B/M) * t_p$. The average transaction duration, considering a fluid-flow model in which service time equals time to transfer a file ($\overline{x}$) is given by:

$$T = \overline{x}\left(1 + \frac{\rho}{2(1-\rho)}\right) \tag{2}$$

Interestingly, curves differ considering 1.5 and 5 ms packet interarrival times. As far as the latter is concerned some statistical multiplexing gain can be observed at the packet level in comparison to the fluid-flow model. Such gain is neglectable at 1.5 ms packet interarrival time since this time is too close to the packet transmission time (1.2 ms). User perceived quality of service can be estimated with the simple M/D/1 model at the expense of an estimation error in the range of 10^{-1} seconds.

Furthermore, we observe that no significant differences in comparison to the M/D/1 case are observed if packet interarrival time is close to packet transmission time (1.5 ms in comparison to 1.2 ms.) and window sizes are large. Intuitively, the performance drop due to flow control depends on the ratio (file size)/(window size). A combination of large file sizes and small flow control windows makes the probability of source active waiting increase. The worst ratio shown in figure 3 is around 10 (4 KB window size and 50 KB file size). Same results can be observed if window size and file size are increased mantaining the same ratio.

3.2 Heavy-tailed file sizes and no flow control

In this section we evaluate the effect of heavy-tailed file sizes with no flow control. Figure 4 shows the average transaction duration with a heavy-tailed file size ($\alpha = 1.05, 1.2, 1.6$) and no flow control. Probability of transaction delay beyond 10 seconds and measured utilization factor versus the transaction arrival rate are also presented. The results match our intuition: the smaller the value of α the larger the squared coefficient of variation and, therefore, smaller values of α imply performance degradation.

The results shown in figure 4 are rather striking: the performance drop caused by file sizes variability is not so significant as far as QOS perceived by user is concerned. Values of α equal to 1.05, 1.2 and 1.6 give approximately the same performance figures (compare with figures 5 and 6). In the next subsection we evaluate the effect of flow control in a heavy-tailed file sizes scenario.

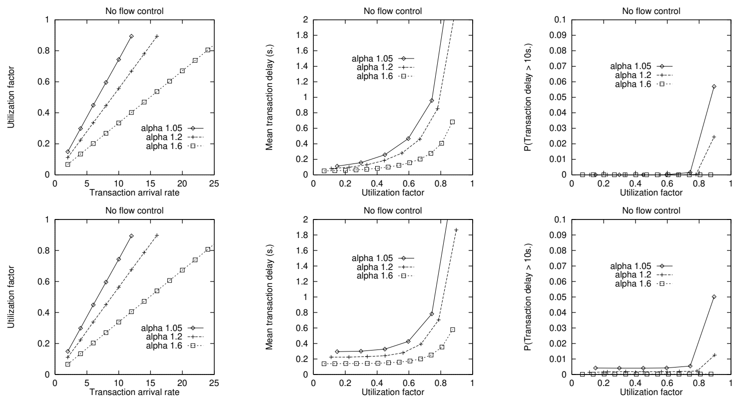

Figure 4 Utilization factor (left), mean transaction duration (center) and probability of transaction duration beyond 10 s. (right). Heavy-tailed file sizes and no flow control. Top: 1.5 ms packet interarrival times, bottom: 5 ms packet interarrival times

3.3 Heavy-tailed file sizes and flow control

Figures 5 and 6 (1.5 and 5ms respectively) show the performance degradation suffered by introducing flow control in a heavy-tailed file sizes scenario. Utilization factor, mean delay and probability of transaction delay beyond a 10 seconds threshold are shown. Note that the typical value reported for α is in the range $1.0 \leq \alpha \leq 1.2$ with an estimate of $\alpha = 1.1$ for the most popular service in the Internet: the WWW [1, 3]. Utilization factors in the range of 0.4 give unacceptable QOS to users for window sizes of 8 KB. Recall from section 2 that a window size of 8KB dominates in our sample with a probability of 61%. The queue seems to be saturated as far as user QOS perception is concerned even for small utilization factors. However, note that the performance degradation is due to the combined effect of flow control and heavy-tailed file sizes. Neither the latter nor the former would produce such performance drop separately as seen in previous sections.

The mean file size for a value of $\alpha = 1.2$ is 50 KB, that gives a ratio of (file size)/(window size) equal to 6.2 for an 8 KB window size and equal to 3.1 for a 16 KB window size. Even in the last case we observe a significant degradation compared to the deterministic file size scenario with a ratio (file size)/(window size) in the vicinity of 10. The heavy-tailed features of file sizes increase the probability of large files present in queue. The performance penalty implications are twofold: first, large files make the service time squared coefficient of variation increase, secondly, since the ratio (file size)/(window size) is larger the traffic source performs active waiting more often.

Considering a 5 ms packet interarrival time (figure 6) we observe a similar behavior. Note that the measured performance is slightly worse. Larger packet

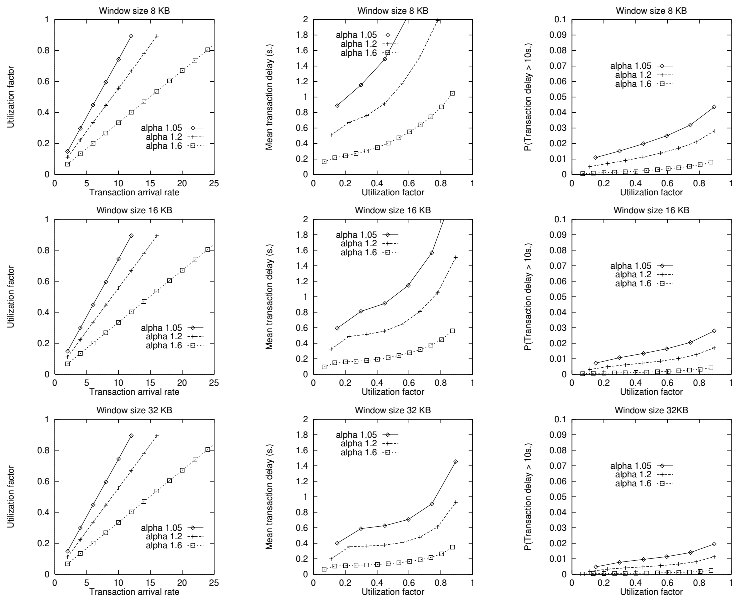

Figure 5 Utilization factor (left), mean transaction duration (center) and probability of transaction duration beyond 10 s. (right). Heavy-tailed file sizes and flow control (top 8 KB window size, center 16 KB window size, bottom 32KB window size, 1.5 ms packet interarrival time).

interarrival times produce an increase in statistical multiplexing gain at the packet level for small values of utilization factor. However, they also make time to transfer a file increase and thus the probability of active waiting for larger values of utilization factor.

4 CONCLUSIONS

Implications for self-similarity. An important conclusion of this paper is that the high variability of file sizes, which generates self-similarity, does not degrade QOS significantly in presence of infinite buffers and large flow control window sizes. In the finite buffer scenario, the joint effect of high variability in file sizes and window flow control supposes a significant performance drop since retransmissions are more likely to occur due to buffer overflow. Such retransmissions make transaction duration grow larger and window size decrease due to the congestion control algorithm.

Implications for billing and pricing TCP services. If we assume that user perceived QOS is determined by transaction duration, an increase in network bandwidth is not translated directly into user satisfaction. Such increase

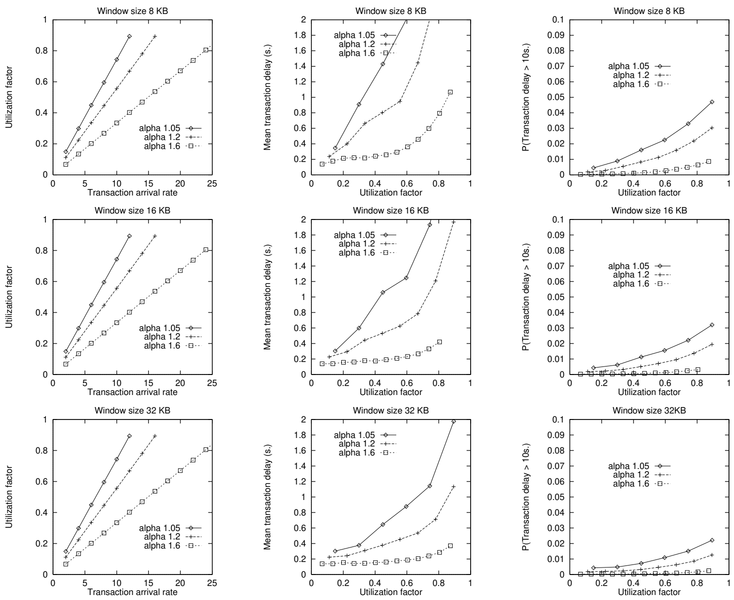

Figure 6 Utilization factor (left), mean transaction duration (center) and probability of transaction duration beyond 10 s. (right). Heavy-tailed file sizes and flow control (top 8 KB window size, center 16 KB window size, bottom 32KB window size, 5 ms packet interarrival time)

would make utilization factor decrease, but note from figures 5 and 6 that an increase in window size would significantly contribute to a better QOS at a lower cost. Therefore, billing for bandwidth may not be an adequate scheme as far as user perceived QOS.

Finally, the use of window flow control protocols with relatively small window size may not be justified in an Internet in which workstations and PCs have a growing capacity of CPU and I/O. New mechanisms for flow control and selective retransmission have to be investigated in order to meet the increasing demand for quality of service in the Internet.

REFERENCES

[1] J. Aracil, R. Edell, and P. Varaiya. An empirical Internet traffic study. In *35th Annual Allerton Conference on Communications, Control and Computing*, Urbana-Champaign, Illinois, October 1997.

[2] David D. Clark. Adding service discrimination to the internet. Technical report, MIT LCS, September 1992.

[3] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traf-

fic: Evidence and possible causes. In *ACM SIGMETRICS Annual Conference on Measurement and Modeling of Computer Systems*, May 1996.
[4] A. Erramili, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, April 1996.
[5] L. Kleinrock. *Queueing Systems*, volume 1. John Wiley and Sons, 1975.
[6] M. Nabe, M. Murata, and H. Miyahara. Analysis and modeling of World Wide Web traffic for capacity dimensioning for Internet access lines. In *SPIE Video, Voice and Data Communications Conference*, Dallas, TX, November 1997.
[7] M. Parulekar and A. Makowski. Tail probabilities of a multiplexer with self-similar traffic. In *IEEE INFOCOM '96*, volume 3, pages 1452–1459, 1996.
[8] V. Paxson and S. Floyd. Wide area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 4(2):226–244, April 1996.
[9] B. K. Ryu and S. B. Lowen. Point process approaches to the modeling and analysis of self-similar traffic - part I: Model construction. In *IEEE INFOCOM '96*, volume 3, pages 1468–1475, March 1996.
[10] B. Tsybakov and N. D. Georganas. On self-similar traffic in ATM queues: Definitions, overflow probability bound and cell delay distribution. *IEEE/ACM Transactions on Networking*, 5(3):397–409, June 1997.
[11] W. Willinger, M. S. Tacqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. In *ACM SIGCOMM 95*, pages 100–113, Cambridge, MA, 1995.

5 BIOGRAPHY

Javier Aracil received the Ph. D. in Telecommunications Engineering from Technical University of Madrid in 1995. In 1996 he was a Fulbright scholar and Postdoctoral researcher of the Dpt. of Electrical Engineering and Computer Sciences of University of California, Berkeley. Nowadays, he is an associate professor of Public University of Navarra, Spain.

Daniel Morató received the MSc in Telecommunications Engineering in 1997. He is a research and teaching assistant of Public University of Navarra.

Mikel Izal received the MSc in Telecommunications Engineering in 1997. He is a research and teaching assistant of Public University of Navarra.