# Analysis of ADSL traffic on an IP backbone link

Nadia Ben Azzouna and Fabrice Guillemin
France Telecom R&D
2, Avenue Pierre Marzin, 22300 Lannion, France

***Abstract*— Measurements from an Internet backbone link carrying TCP traffic towards different ADSL areas are analyzed in this paper. For traffic analysis, we adopt a flow based approach and the popular mice/elephants dichotomy. The originality of the experimental data reported in this paper, when compared with previous measurements from very high speed backbone links, is in that commercial traffic comprises a significant part generated by peer-to-peer applications. This kind of traffic exhibits some remarkable properties in terms of mice and elephants, which are described in this paper. It turns out that by adopting a suitable level of aggregation, the bit rate of mice can be described by means of a Gaussian process. The bit rate of elephants is smoother than that of mice and can also be well approximated by a Gaussian process.**

## I. Introduction

Characterization of Internet traffic has become over the past few years one of the major challenging issues in telecommunications networks. As a matter of fact, understanding the composition and the dynamics of Internet traffic is essential for network operators in order to offer quality of service and to supervise their networks. Since the celebrated paper by Leland *et al* [1] on the self-similar nature of Ethernet traffic in local area networks, a huge amount of work has been devoted to the characterization of Internet traffic. In particular, different hypotheses and assumptions have been explored to explain the reasons why and how Internet traffic should be self-similar (see for instance [2], [3]).

A common approach to describing traffic in a backbone network consists of observing the bit rate process evaluated over fixed length intervals, say a few hundreds of milliseconds. Long range dependence as well as self-similarity are two basic properties of the bit rate process, which have been observed through measurements in many different situations. Different characterizations of the fractal nature of traffic have been proposed in the literature (see for instance Norros [4] on the mono-fractal characterization of traffic and Levy-Véhel *et al* [5], Abry *et al* [6] on the multi-fractal properties of traffic). An exhaustive account to fractal characterization of Internet traffic can be found in the book by Park and Willinger [7].

Even though long range dependence and self similarity properties are very intriguing from a theoretical point of view, their significance in network design has recently been questioned in the paper by Cao and Ramanan [8], where it is shown that the overflow probability in a buffer fed with the superposition of a large number of flows satisfying some reasonable regularity assumptions can be well approximated by that obtained when the input process is Poisson. While the above result may not be directly applicable in an access network with limited transmission capacities, the assumption of a large number of flows is reasonable on a high speed backbone link and are in favor of using a simple $M/G/1$ queue for buffer dimensioning in a backbone network composed of gigarouters.

While self-similar models introduced so far in the literature aims at describing the global traffic on a link, it is now usual to distinguish short transfers (referred to as mice) and long transfers (referred to as elephants) [9]. This dichotomy was not totally clear up to a recent past (see for instance network measurements from the MCI backbone network [10]). Now, the discrimination between mice and elephants become more and more evident with the emergence of peer-to-peer (p2p) applications, which give rise to a large amount of traffic on a small number of TCP connections, as it will be shown in the following.

In this paper, we analyze TCP traffic on an Internet backbone link collecting data in direction to several ADSL areas. The primary goal of this paper is to draw attention to several salient features of ADSL traffic. In particular, we consider commercial traffic, which comprises a significant part of p2p traffic, giving rise to very large elephants.

The above observation leads us to analyze ADSL traffic by adopting a flow based approach and more precisely the mice/elephants dichotomy. The intuitive definition of a mouse is that such a flow comprises a small number of packets so that it does not leave or leaves slightly the slow start regime. Thus, a mouse is not very sensitive to the bandwidth sharing imposed by TCP. On the contrary, elephants are sufficiently large so that one may expect that elephants share the bandwidth of a bottleneck according to the flow control mechanism of TCP. As a consequence, mice and elephants have a totally different behavior from a modeling point of view.

The organization of this paper is as follows: Basic definitions are given in Section II. Mice traffic is analyzed in Section III, where p2p and non p2p mice are handled separately. Elephants traffic is described in Section IV. Finally, some concluding remarks are presented in Section V.

## II. Traffic analysis using mice and elephants

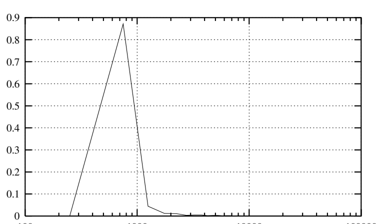
Throughout this paper, we consider a 1 Gbps link between the France Telecom IP backbone network and several ADSL areas. Traffic originated or in direction to these different ADSL areas is multiplexed on this single link. We observe TCP traffic from the IP backbone network towards the ADSL areas (downstream traffic). It is worth noting that traffic local to an ADSL area cannot be observed in the collected data.

0-7803-7975-6/03/$17.00 (C) 2003

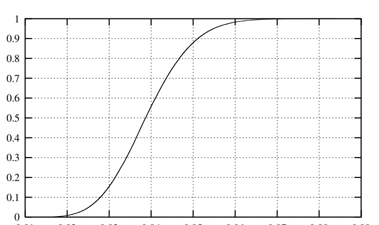
To analyze the traffic characteristics, we adopt the mice and elephants dichotomy. There is no commonly adopted definition for a mouse. A mouse is intuitively a data transfer, which does not leave or leaves slightly the slow start period. In fact, a mouse is a short data transfer, which has no time to adapt to network conditions according to the TCP control loop.

As a convention, we adopt in this paper the following definition: a mouse is a data transfer comprising a number of packets less than or equal to 20 packets; a flow is terminated if no packets of the flow have been observed for a time period of 5 seconds. Other definitions for mice are possible; for instance in the paper by Zhang *et al* [11], a small data transfer contains at most $10^4$ bytes. If the MTU is equal to 1500 bytes, $10^4$ bytes roughly correspond to 8 packets. The value of 20 packets is chosen because if we assume that the maximum congestion window size is 8Kbytes and if there is an ACK for each packet received by the destination, then about 15 packets are necessary to hit the maximum congestion window size in the slow start phase.

The timer of 5 seconds may appear at first glance very sharp. However, since we intend to describe the bit rate of mice, we have to consider the data transfer phase of a mouse. Long mice are mostly due to FIN segments, which arrive quite a long time after the last data segment. This introduces some bias in the evaluation of the duration of mice. To avoid this phenomenon, we use the 5 s timer to remove segments, which are too far away from data segments. The counterpart of this method is that single packet mice artificially appear.



(a) Flow size in bytes



(b) PDF of the contribution of mice to global traffic

Fig. 1. Flows on a backbone link.

The flow size distribution of downstream traffic on the 1 Gbps link is displayed in Figure 1(a). From this figure, it turns out that the majority of flows comprise less than 1000 bytes and actually correspond, as shown in the following, to mice. Even though these flows are the most numerous, they contribute a very small proportion of the total amount of traffic, as shown in Figure 1(b) representing the distribution of $X_t/Z_t$, where $\{X_t\}$ is the amount of traffic due to mice and $\{Z_t\}$ is the global bit rate process. Mice actually contribute about 6% of global traffic but represent more than 97% of the total number of flows.

Finally, on the link observed, more than 49% of traffic is due to p2p applications (Kazaa, Morpheus, Edonkey, Gnutella, etc.), as shown in Table I. In this table, only p2p traffic observable via port numbers is reported. The significant proportion of p2p traffic gives rise to remarkable phenomena, which are described in the next sections.

| Applications | | percentage |
|---|---|---|
| non p2p | http | 14.6 |
| | ftp | 2.1 |
| | nntp | 1.9 |
| | others | 31.8 |
| | total non p2p traffic | 50.4 |
| p2p | Edonkey | 37.5 |
| | Kazaa&Morpheus | 7.8 |
| | Napster | 3.8 |
| | Gnutella | 0.3 |
| | Total p2p traffic | 49.6 |

TABLE I

COMPOSITION OF ADSL TRAFFIC PER APPLICATION.

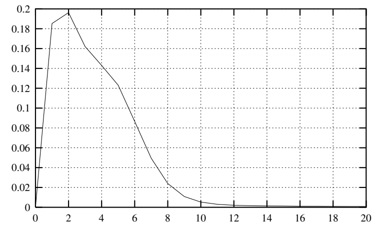
## III. CHARACTERISTICS OF MICE TRAFFIC

### A. Analysis of mice

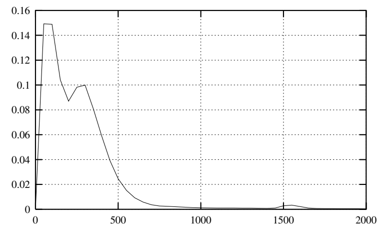
Figure 2 displays the distribution of the number of packets and bytes comprised in a mouse. It turns out that the majority of mice comprise less than 1000 bytes and as stated in the previous section, the majority of flows are indeed mice (see Figure 1(a)).

From Figure 2, we also observe that a large number of mice are composed only of one or two packets. Single packet mice are Reset segments, SYN segments, which are not really associated to a mouse because of transaction interruption or very long response times by servers, or FIN segments, which arrive far away from the last data segments and which appear as single packet mice because of the 5 s timer used to decide whether a mouse is terminated. Moreover, a large number of single packet mice are generated by p2p protocols.

Two packet mice are composed of SYN and FIN segments only. This is due to the fact that a large number of TCP connections (associated with HTTP transactions for instance) are opened and immediately closed or not used at all; this may be caused by too long response times by servers, which lead users to interrupt their transactions, or by the fact that certain implementations of HTTP systematically opens several TCP connections in parallel. Actually, only a small number of mice carry useful information (data segments). This phenomenon has

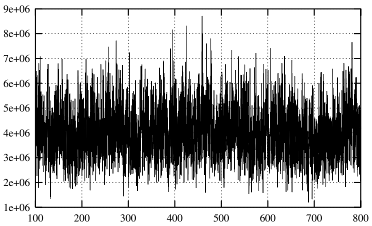



(a) Number of packets

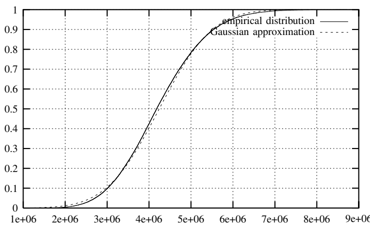



(b) Number of bytes

Fig. 2. Distribution of the number of packets and bytes comprised in a mouse.




(a) Bit rate




(b) Stationary distribution

Fig. 3. Instantaneous bit rate and stationary distribution of the bit rate process $\{X_t^1\}$ estimated over time intervals with length $\Delta = 100$ ms.

to be taken into account when characterizing the mice arrival process, as shown in the following.

When analyzing more carefully the generation process of mice, it turns out that mice generated by p2p protocols exhibit a behavior, which is quite different from that of other mice (regular mice related to usual applications using TCP, such HTTP, ftp, etc.). This is why we analyze the two types of mice separately. Note that since mice are not sensitive to TCP fairness, global mice traffic is the superposition of p2p and non p2p mice traffic; these two types of traffic do not really interact one with each other.

### B. Non p2p mice

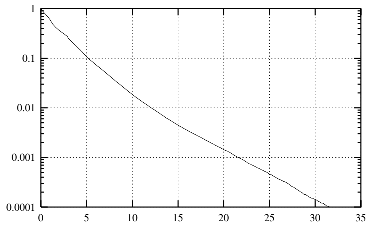
In this section, we analyze mice, which are not generated by p2p protocols, i.e., with port numbers different from 1214 (Kazaa), 4662 (Edonkey), 6346 (Gnutella) and other p2p protocol port numbers. The objective of this section is to describe the bit rate process of those mice.

The process $\{X_t^1\}$ representing the bit rate offered by mice evaluated over time intervals with length $\Delta = 100$ ms, is highly varying as displayed in Figure 3(a). The "instantaneous" bit rate has been observed over a time period of 4900 seconds between 1:27 pm to 2:51 pm; only a time interval of 700 seconds is displayed in Figure 3(a). The empirical distribution of $X_t^1$ is displayed in Figure 3(b). It turns out that this distribution is very close a Gaussian distribution. It can actually be shown (see [12] for details) that the process $\{X_t^1\}$ is indeed Gaussian.

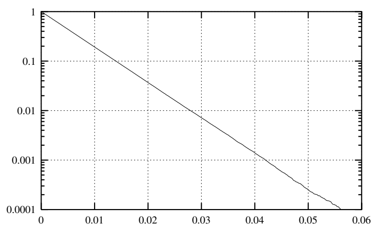
To explain the form of the curve of $\{X_t^1\}$, let us describe mouse traffic in more details. In a first step, we have observed arrivals of individual mice. The mouse inter-arrival time is remarkably exponential and the mouse duration is Weibull (see [12] for details). At first glance, one may conclude that mice arrive according to a Poisson process. However, when we compute the stationary distribution of the number of active mice at an arbitrary instant, we should obtain a Poisson distribution if the mouse arrival process were Poisson (namely, the stationary distribution of the number of customers in an $M/G/\infty$ queue). In particular, if this were true, the variance should be equal to the mean. However, experimental data show that this last property is not verified. This is sufficient to show that the mouse arrival process is not Poisson.

To overcome this problem, we note that, as mentioned above, mice are actually not independent. In fact, for a same destination IP address, a certain number of mice arrive near one to each other, forming what we call is the following a macro-mouse. We specifically define a macro-mouse as a set of mice, which have the same destination address and which arrive within a rather short time interval, say with a length of 1 second; moreover, a macro-mouse must comprise more than one packet. The inter-arrival time of macro-mice is displayed in Figure 4(b) and the distribution of their duration is displayed in Figure 4(a). Their inter-arrival time is exponential with mean $1/\lambda = 0.00562$. The probability distribution of the duration of a macro-mouse can be well approximated by a Weibullian distribution with scale parameter $\eta = 1.78$, skew parameter $\beta = 0.8$, and location parameter equal to 0. The mean number of mice in a macro-mouse is equal to 1.8.

When computing the stationary distribution of the number of macro-mice active at a given instant, we get a Poisson distribution. Moreover, the Arrival See Time Averages property




(a) Duration




(b) Inter-arrival time

Fig. 4. Characteristics of a mouse group.

is verified. Hence, we may reasonably conjecture that the macro-mouse arrival process is Poisson. Thus, in spite of the fact that the individual mouse arrival process is not Poisson, grouping mice in an adequate manner yields a Poisson process.

When we consider the bit rate created by macro-mice, we can adopt a fluid flow approach, i.e., by neglecting discrete packet arrivals, we assume that the bit rate of a macro-mouse is constant and equal to the number of bytes divided by the duration of the macro-mouse. We then get the fluid approximation of the bit rate of the macro-mouse. The key point is in that since the mean arrival rate $\lambda$ of macro-mice is high, the fluid bit rate of macro-mice converges in distribution to a Gaussian process, which auto-correlation function is perfectly known. This last property is crucially due to the fact that the macro-mouse arrival process is Poisson. It follows that the actual arrival bit rate can be approximated by a Gaussian process perturbed by a white noise, which is due to discrete packet arrivals. The autocorrelation function of the Gaussian process is given by

$$c(t) = \frac{\mathbb{E}[Y^2(\sigma - t)^+]}{\mathbb{E}[Y^2\sigma]}, \tag{1}$$

and $Y$ is the bit rate of the macro-mouse and $\sigma$ is duration.

Finally, it remains a fraction of single packet mice, which are not included in macro-mice. The bit rate created by these mice is very small (a few tens of Kbps). We admit without more details (see [12] for more information) that single packet mice is very small and is in fact a white noise. With the collected data, the mean and the variance of this white noise are equal to $m = 3668$ bit/s and $\sigma^2 = 368,195.71$ (bit/s)$^2$, respectively.

In conclusion, the bit rate created by macro-mice is basically a Gaussian process perturbed by a white noise, which is due to single packet mice and discrete packet arrivals. This can be verified by computing the empirical spectral function of the time series $\{X_t^1\}$ and the theoretical spectral function corresponding to the autocorrelation function defined by equation (1); see [12] for more details.

### *C. P2p mice*

We analyze in this section the traffic offered by p2p, i.e., mice with a port number corresponding to a p2p protocol. Figure 5 represents the number of active p2p mice. Moreover, it is possible to compute the empirical probability distribution of the p2p mouse inter-arrival time [12]. While the p2p mouse inter-arrival time is exponential, it clearly appears that the process counting the number of active p2p mice is composed of "bursts" as shown in Figure 5. Finally, the size of p2p mice is rather small (in general less than 8 packets).

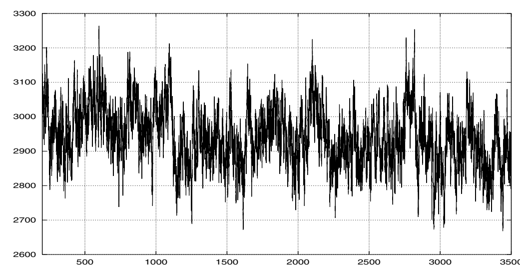



Fig. 5. Number of active p2p mice.

As in the previous section, we are led to group mice comprising more than one packet according to some criterion. At a first glance, we may group p2p mice according their source address. Intuitively, this criterion corresponds to the fact that a member of a p2p network seeking a content sends requests to different nodes. But this level of aggregation is not sufficient because the process counting the aggregated p2p mice on the basis their source address remains quite irregular. A second level of aggregation consists of grouping the aggregated p2p mice on the basis of their destination address. This is motivated by the fact that the different hosts of the p2p network respond to the request corresponding to an aggregated p2p mice.

This second level of aggregation gives rise to macro p2p mice, which are composed of p2p mice with the same address source and/or the same destination address and arriving in a time interval of 1 second. It turns out that the process counting these p2p macro mice is indeed Poisson. This is checked by considering the process counting the number of active p2p macro mice over time. The resulting process is identical to the occupation process of an $M/G/\infty$ queue.

From a theoretical point of view, p2p macro mice can be described as Poisson clouds. But, for characterizing their offered bit rate, we can consider as in the previous section the fluid bit rate of p2p macro mice. The fluid bit rate of a p2p macro mouse is simply the quanitity of bit contained in a p2p macro mouse divided by the duration of the mouse. The exact bit rate can then be roughly approximated by the fluid bit rate perturbed by a white noise; this latter white noise is due to discrete packet arrivals. Finally, as in the previous section, it remains a white noise due to single packet p2p mice.

The advantage of the above approach is in that the global fluid bit rate of p2p macro mice can be approximated by a Gaussian process, which autocorrelation function is given by equation (1), where as above $Y$ is the bit rate of a p2p macro mouse and $\sigma$ is its duration. As in the previous section, it turns out that the bit rate of p2p mice can be described by means of a Gaussian process perturbed by a white noise.

## IV. Characteristics of elephants

In this section, we investigate the bit rate created by elephants. Figure 6 displays the global bit rate due to elephants on the link. The first observation is that the bit rate of elephants is large and much smoother than the bit rate of mice. In fact, the bit rate of elephants is oscillating around a mean value.

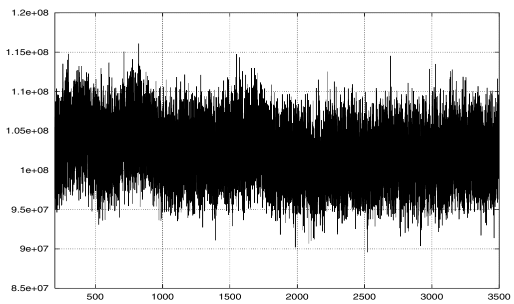



Fig. 6. Bit rate created by elephants.

While it is usually assumed that elephants share the transmission capacity of a bottleneck according to some fairness criterion (max-min or proportional fairness), we first have to draw attention to the fact that certain elephants are by nature with a very small bit rate. This is typically the case of elephants composed of ACKs generated by a terminal retrieving a large file. To eliminate those elephants, we have fixed a threshold for the mean value of the length of packets contained in an elephant. If the mean packet length is less than 80 bytes, this certainly means that the elephant is merely composed of ACK segments and its bit rate is small. Those elephants represent a small fraction of the global bit rate of elephants and are eliminated in the following. As a first approximation, these elephants generate a bit rate, which is a constant plus a white noise (due to discrete packet arrivals).

When examining the bit rate created by the remaining part of the elephants, we come up with the conclusion that the link is not congested (the global load is about 10 percent). Hence, elephants are bottlenecked somewhere else in the network and the transmission capacity of the observed link is not shared by the TCP control loop. Thus, the $M/G/1$ processor sharing queue cannot be directly used. In fact, in the case of ADSL traffic, bottlenecks are frequently located on the link between the broadband access server and the customer terminal. The $M/G/1$ processor sharing queue may possibly be used for such bottlenecks.

A more careful analysis of the dynamics of elephants shows that the transmission of packets in elephants is not constant. In fact, the transmission of data is composed of transmission phases where a large number of packets are transmitted, elapsed by periods, where only a few packets are transferred. Thus, the transmission of packets during elephants is not smooth but rather bursty. This phenomenon may be due to various factors. One possible cause is that a p2p server does not always serve the same TCP connection but may share its resources in a round robin manner between different clients. This may cause bursty transmission phases elapsed by less active transmission periods. Nevertheless, it can be shown that the bit rate created by elephants is still a Gaussian process.

## V. Conclusion

We have analyzed in this paper TCP traffic delivered by an IP backbone network to several ADSL areas. One salient feature is that a significant part of global traffic is due to p2p applications.

It is possible to decompose traffic into several components on the basis of the mice/elephants dichotomy. By analyzing each component separately and by adopting an adequate level of aggregation, it is possible to describe each component by means of a Gaussian process perturbed by a white noise. The next step is to investigate how to characterize global traffic by means of a few parameters only. Indeed, to monitor traffic and to estimate the quality of service perceived by users (essentially through the bit rates achieved by elephants), traffic traces shall be analyzed but recording complete traffic traces would require huge storage capacities and prohibitive off line processing.

## References


[1] W. Leland, M..Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Trans. Net.*, pp. 1–15, 1994.

[2] M. Crovella and A. Bestravos, "Self-similarity in world wide web. Evidence and possible causes," *IEEE/ACM Trans. on Networking*, pp. 835–846, December 1997.

[3] A. Feldmann, A. Gilbert, W. Willinger, and T. Kurtz, "The changing nature of network traffic: Scaling phenomena," in *Computer Communication Review*, vol. 28, 1998, pp. 5–19.

[4] I. Norros, "On the use of fractional Brownian motion in the theory of connectionless networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, August 1995.

[5] J. L. Véhel and R. Riedi, *Fractals in Engineering*. Springer, 1997, ch. Fractional Brownian motion and data traffic modeling: The other end of the spectrum.

[6] P. Abry, R. Baraniuk, P. Flandrin, R. Riedi, and D. Veitch, "The multiscale nature of network traffic," *IEEE Signal Processing Magazine*, 2002.

[7] K. Park and W. Willinger, Eds., *Self similar network traffic and performance evaluation*. Wiley Interscience, 2000.

[8] J. Cao and K. Ramanan, "A Poisson limit for buffer overflow probabilities," in *Proc. Infocom 2002*, New York, June 2002.

[9] V. Paxson and S. Floyd, "Wide area traffic: The failure of the Poisson assumption," *IEEE/ACM Trans. on Networking*, pp. 226–244, 1995.

[10] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurement from an Internet backbone," in *Proc. of Inet*, 1998.

[11] Z.L.Zhang, V. Ribeiro, S. Moon, and C. Diot, "Small time scaling behaviora of Internet backbone traffic: An empirical study," in *Proc. Infocom 2003*, San Francisco, April 2003.

[12] N. B. Azzouna, C. Fricker, and F. Guillemin, "Modeling ADSL traffic on an IP backbone link," February 2003, submitted for publication.