On the Properties of TCP Flow Arrival Process

Dario Rossi, Luca Muscariello, Marco Mellia
Dipartimento di Elettronica – Politecnico di Torino
Corso Duca degli Abruzzi, 24 – I-10129 Torino, Italy
{rossi,muscariello,mellia}@mail.tlc.polito.it

***Abstract*— We study the TCP flow arrival process, starting from the aggregated measurement at the TCP flow level taken from our campus network. In particular, we analyze the statistical properties of the TCP flow arrival process. We define different traffic aggregates by splitting the original trace, such that i) each of them is constituted by all the TCP flows belonging to the same traffic relation, i.e., with the same source/destination IP addresses and ii) each traffic aggregate has, bytewise, the same amount of traffic. To induce a divisions of TCP-elephants and TCP-mice into different traffic aggregates, the used algorithm packs the largest traffic relations in the first traffic aggregates, so that subsequently generated aggregates are constituted by an increasing number of smaller traffic relations. The Long Range Dependency (LRD) characteristics are presented, showing as possible causes of the LRD of TCP flow arrival process i) the heavy tailed distribution of the number of flows in a traffic aggregate, and ii) the presence of TCP-elephants within them.**

I. Introduction

Since the pioneering work of Danzig [1], [2], and Paxons [3], [4] the interest in data collection, measurement and analysis to characterize either the network or the users behavior increased steadily, also because it was clear from the very beginning that "measuring" the Internet was not an easy job. The lack of simple, yet satisfactory model like the traditional Erlang teletraffic theory for the circuit-switched networks, still pose this research field as a central topic of the current research community. Moreover, the well known Long Range Dependency (LRD) behavior shown by the Internet traffic makes traffic measuring and modeling even more interesting. Indeed, after the two seminal papers [4], [5], in which authors showed that traffic traces captured on both LANs and WANs exhibit LRD properties, many works focused on studying the behavior of data traffic in packet networks. This, with the intent of both trying to find a physical explanation of the properties displayed by the traffic, and to find accurate stochastic processes that can be used for the traffic description in analytical models.

Considering the design of the Internet, it is possible to devise three different layers at which study Internet traffic: Application, Transport and Network layer, to which user sessions, TCP or UDP flows, and IP packets respectively correspond. In this paper, we only concentrate our attention to the *flow level*, and to the TCP flow level in particular, given that the majority of the traffic is today transported using the TCP protocol. The motivation behind this choice is that while it was shown (e.g., [4], [6]) that arrival processes of both packets and flows exhibit LRD properties, a lot of researchers concentrated their attention to the packet level, while the flow level traffic characteristics are relatively less studied. Moreover, even if the packet level is of great interest to support router design, e.g., for buffer dimensioning, the study of the TCP flow level is becoming more and more important, since the flow arrival process is of direct role in the dimensioning processes of web servers proxies and performance of flow aware algorithms; Going back at the packet level, the prevailing justification to the LRD presence at this layer is supposed to be the heavy tailed distribution of files size [7]: the presence of long-lived flows, called in the literature "elephants", induces correlation to the packet level traffic, even if the majority of the traffic is build by short-lived flows, or "mice". The question we try to answer in this paper is whether the presence of mice and elephants has an influence to the LRD characteristics at the *flow level* as well. To face this topic, we collected several days of live traffic from our campus network at the Politecnico di Torino, which consist of more than 7000 hosts, the majority of which are clients. Instead of considering the packet level trace, we performed a live collection of data directly at the TCP flow level, using `Tstat` [8], a tool able to keep track of single TCP flows by looking at both the data and acknowledgment segments. The flow level trace was then post-processed by `DiaNa` [9], a novel tool which allowed to easily derive several simple as well as very complex measurement indexes in a very efficient way. Both tools are under development and made available to the research community as open source.

To gauge the impact of elephants and mice on the TCP flow arrival process, we follow an approach similar to [10], [11], that creates a number of artificial scenarios, deriving each of them from the original trace into a number of sub-traces. We then study the statistical properties of different sub-traces, showing that the LRD tends to vanish on traffic aggregates composed mostly of TCP-mice.

II. Problem Definition

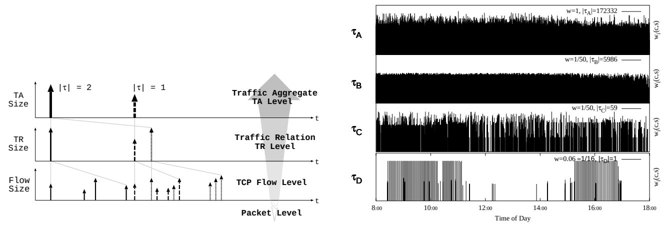
A. Preliminary Definitions

When performing trace analysis, it is possible to focus the attention on different aggregation levels. To mimic the splitting/aggregation process that data experience following different paths in the network, we define four level of aggregation, sketched in Fig. 1-a: IP packets, TCP/UDP flows, Traffic Relations (TR), and Traffic Aggregates (TA). Being interested into the Flow arrival process, we will neglect the packet level, and also the UDP traffic, because of its connectionless nature, and because it consists of a small portion of the current Internet traffic. Let us define:

TCP Flow: A single TCP connection[1] is constituted by several packets exchanged between the same client c (i.e., the host that

This work was supported by the Italian Ministry for University and Scientific Research under the project TANGO.

[1] In this paper we use the term "flow" and "connection" interchangeably.



(a) **(b)**

Fig. 1. **(a)** Aggregation Level from IP to TA Level and **(b)** Flow Size and Arrival Times for Different TAs

performed the *active* open) and same server s (i.e., the host that performed the *passive* open), having besides the same (source, destination) TCP port pair. We will consider only TCP flows whose three-way-handshake was successful, in which case the flow arrival time correspond to the first client SYN segment observed. $F_i(c, s)$ denotes the bytewise size of the i-th flow, which considers the amount of bytes flowing from the server s toward the client c, i.e., the usually most relevant part of data.

Traffic Relation: A Traffic Relation (TR) $\mathcal{T}(c, s)$ aggregates all TCP flows having c as client and s as server; $T_R(c, s) = \Sigma_{i=1}^{|\mathcal{T}(c,s)|} F_i(c, s)$ denotes its size, expressed in bytes. The intuition behind this aggregation criterion is that all the packets within TCP flows belonging to the same TR (usually) follow the same path(s) in the network, yielding then the same statistical properties on links along the path(s).

Traffic Aggregate: We define a higher level of aggregation, which we call Traffic Aggregate (TA). The J-th TA is stated by τ_J and its bytewise size is $T_A(J) = \Sigma_{(c,s)\in\{\tau_J\}} T_R(c, s)$; TAs will be artificially created such that, at a given aggregation level, they all have (almost) the same size. In the following, rather than using the bytewise size of flows, TRs and TAs, we will refer to their *weight*, i.e., their size normalized over the total amount of bytes observed in the whole trace $W = \Sigma_i \Sigma_{(c,s)} F_i(c, s)$; thus $\hat{w}_i(c, s) = F_i(c, s)/W$, $\hat{w}_{cs} = T_R(c, s)/W$, $\hat{w}_J = T_A(J)/W$ will be the flow, TR, TA weight respectively.

B. Input Data

The analysis was conduced over different traces collected in several days over our Institution ISP link during October 2002. Our campus network is built upon a large 100 Mbps Ethernet LAN, connected to the Internet by a single router, whose WAN link has a capacity of 28 Mbps[2]. We used `Tstat` to perform the live analysis of the incoming and outgoing packets sniffed on the router WAN link, splitting the flow-level trace into several 24 hours blocks, each of which has been separately analyzed. We eliminated the non-stationary time-interval (i.e., night/day effect) from each block and considered then the busy-period from 8:00 to 18:00. It is worth to point out that, even though *backbone* traffic is stationary over a few hours only, our *edge* traffic can be considered stationary over a longer window. Besides, we considered only the flows originated by clients internal to our LAN, since they represent the vast majority of hosts.

[2]The data-link level is based on an AAL-5 ATM virtual circuit at 34 Mbps (OC1).

TABLE I
TRACE SUMMARY

Internal Clients	2,380	Flows Number	$2.19 \cdot 10^6$
External Servers	35,988	Packets Number	$71.76 \cdot 10^6$
Traffi c Relations	172,574	Total Trace Size	79.9 GB

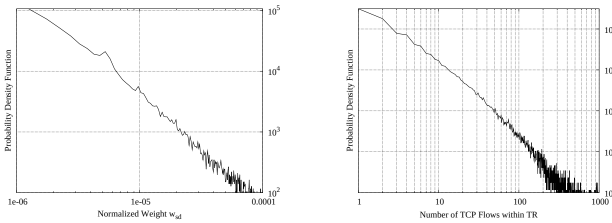
Given that the qualitative results observed on several traces do not change, in this paper we present results derived from a single trace, whose properties are briefly reported in Tab. I. In the considered mixture of traffic, TCP protocol represents 94% of the total packets, which allows us to neglect the influence of the other protocols; considering the application services, TCP connections are mainly constituted by HTTP flows, representing 86% of the total services and more than half of the totally exchanged bytes. Notice that Peer-to-peer traffic is blocked by a firewall. Considering the different traffic aggregation levels previously defined, Fig. 1-b shows examples of the flow arrival time sequence. Each vertical line represents a single TCP flow, which started at the corresponding time instant of the x-axis, and whose weight $\hat{w}_i(c, s)$ is reported on the y-axis. The upper plot shows trace τ_A, whose $\hat{w}_A = 1$, and represents the largest possible TA, built considering all connections among all the possible source-destination pairs.

Trace sub-portions τ_B and τ_C, while being constituted by a rather different number of TRs ($|\tau_B| = 5986$ and $|\tau_C| = 59$), have indeed the same weight $\hat{w}_B = \hat{w}_C = 1/50$. Observing Fig. 1-b, it can be gathered that τ_B aggregates a larger number of flows than τ_C; furthermore, weight of τ_B flows is smaller (i.e., TCP flows tend to be “mice”), while τ_C is build by a much smaller number of heavier (i.e., “elephants”) TCP flows. This intuition will be confirmed by the data analysis presented in Sec. III. Finally, TCP flows shown in τ_D constitute a unique traffic relation; this TR is built by a small number of TCP flows, whose weight is very large, so that they amount to 1/16 of the total traffic.

To give the reader more details on the statistical properties of the different TRs, Fig. 2-a shows the cumulative distribution of $\hat{w}_{cs}$ for all the TRs (using a log/log plot). It can be noticed that the distribution can be approximated by a heavy tailed distribution. Given that we are interested into the TCP flow arrival process, the distribution of the TCP-flows number per TR is more interesting. It is shown in Fig. 2-b using a log/log plot, which exhibits a clearly heavy tailed distribution. It is tempting to conjecture that this could be one of the possible causes of the LRD properties in the TCP-flow arrival process: this is analogous with respect to the heavy-tailed pdf of the flow-size, which induces LRD properties to the packet level, i.e., the also known $M/G/\infty$ effect, and service time distribution with infinite variance, e.g., Pareto distributed service times.

C. Properties of the Aggregation Criterion

The aggregation criterion has been designed in order to satisfy some properties that help the analysis and interpretation of results. We consider TR aggregation a natural choice, since it preserves the characteristics of packet within TCP flows following the same network path, having therefore similar properties. But several TRs can share the same link along their path, thus forming an higher level of aggregate, i.e, a TA. Obviously, different criterion can be defined when forming



(a) **(b)**

Fig. 2. **(a)** Traffic Relation Size and **(b)** Flow Number Distributions

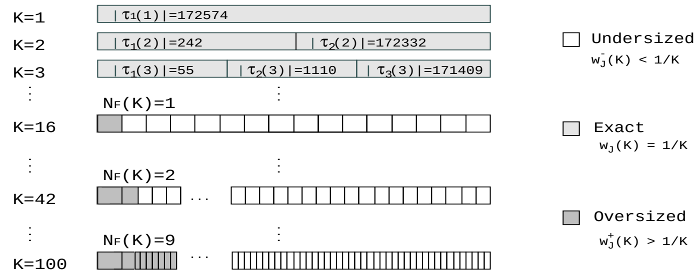


Fig. 3. Trace Partitioning: Algorithm Behavior

TA; among those, we impose that the original trace is split into K different TA, such that each TA has, bytewise, the same amount of traffic, i.e., the K-th portion of the total traffic. Being possible to find more than one solution to the previous problem, the splitting algorithm we implemented packs the largest TRs in the first TAs; besides, in virtue of the bytewise traffic constraint, subsequently generated aggregates are constituted by an increasing number of smaller TRs. To formalize the problem, and to introduce the notation that will be used also to present the results, let us define:

- Class K: the number of TA in which we split the trace;
- Slot J: a specific TA of class K, namely $\tau_J(K)$, $J \in [1, K]$;
- Weight $\hat{w}_J(K)$: the weight of slot J of class K;
- Target Weight $\hat{w}(K) = 1/K$: the ideal portion of the traffic that should be present in each TA at class K.

Fig. 3 sketches the splitting procedure. When considering class $K = 1$ we have a single TA of weight $\hat{w}(K) = 1$, derived by aggregating all TRs. It corresponds to the original trace. Considering $K = 2$, we have two TAs, namely $\tau_1(2)$ and $\tau_2(2)$; the former is build by TRs (only 242 in the considered trace), which account to $\hat{w}(K) = 1/2$ of relative traffic, while the latter contains all the remaining TRs. This procedure can be repeated for increasing values of K, until the weight of a single traffic relation becomes larger than the target weight. Being impossible to split a single TR into smaller aggregates, we are forced to consider TAs having a weight $\hat{w}^+_J(K) > \hat{w}(K)$. The weight $\hat{w}(K)$ has therefore to be interpreted as an *ideal* target, in the sense that it is possible that one (or more) TRs will have a weight $\hat{w}_J(K) > \hat{w}(K)$, as the number of slots grows. In such cases, there will be a number of *fixed* slots (stated by $N(K)$), i.e., TAs constituted by a single TR, of weight $\hat{w}^+_J(K) > w(K)$; the remaining weight will be distributed over the $K - N(K)$ non-fixed slots; therefore the definition of $\hat{w}_J(K)$ is:

$$\hat{w}_J(K) = \begin{cases} \hat{w}^+_J(K) > \dfrac{1}{K}, & 0<J\leq N(K) \\ \dfrac{1-\sum_{i=1}^{N(K)} \hat{w}^+_i(K)}{K-N(K)} < \dfrac{1}{K}, & N(K)<J\leq K \end{cases}$$

In the dataset considered in this paper, for example, the TR τ_D shown in Fig. 1 is the largest of the whole trace, having $\hat{w}_D = 0.062 > 1/16$. Therefore, from class $K = 16$ on, the slot J=1 will be always occupied by this aggregate, i.e., $\tau_1(K) = \tau_D, \forall K \geq 16$, as evidenced in Fig. 3.

D. Trace Partitioning Model and Algorithm

More formally, the problem can be conduced to a well known optimization problem $P||C_{max}$ of job scheduling over identical parallel machines [12], which is known to be strongly NP-hard. Traffic relations TR are the jobs that have to be scheduled on a fixed number K of machines (i.e., TA) minimizing the maximum completion time (i.e., the TA weight). The previously introduced ideal target $\hat{w}_K = 1/K$ is the optimum solution in the case of *preemptive* scheduling. Since we preserve the TR identities, preemption is not allowed; however, it is straightforward that minimizing the maximum deviation of the completion time from $w(K)$ is equivalent to the objective function that minimize the maximum completion time. We define $\underline{R}$ as the $1 \times |\tau_1(1)|$ vector of the jobs length (i.e., TR weights $\hat{w}_i(c, s)$) and $\underline{A}$ as the $1 \times K$ vector of the machine completion times (i.e., TA weights $\hat{w}_J$). Denoting the mapping matrix as $\underline{M}$ (i.e., $\underline{M}_{ij} = 1$ means that the i-th job is assigned to the j-th machine), and stating with $\underline{M}_i$ its i-th column, we have:

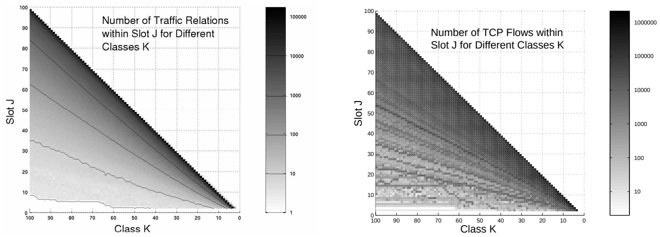
$$\begin{aligned} & min\ p \\ & p \geq z_i, \quad \forall i \in [1, K] \subset \mathbb{N} \\ \text{s.t.} & \begin{cases} \sum_j \underline{M}_{ij} = 1, & \forall i \\ z_i \geq \underline{M}_i \cdot \underline{R} - A_i, & \forall i \\ z_i \geq A_i - \underline{M}_i \cdot \underline{R}, & \forall i \end{cases} \end{aligned}$$

The greedy adopted solution, which has the advantage over, e.g., an LPT[12] solution of the clustering properties earlier discussed, implies the preliminary bytewise sorting of the traffic relations, and three simple rules:

- allow a machine load to exceed $1/K$ if the machine has no previously scheduled job;
- keep scheduling the biggest unscheduled job into the same machine while the load is still below $1/K$;
- remove the scheduled job from the unscheduled jobs list as soon as job has been scheduled.

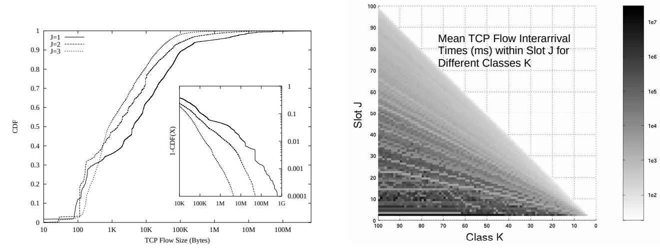
III. Results

To help the presentation and discussion of the measurement results, we first propose a visual representation of the dataset in which data is plotted as a function of the class K and slot J indexes, using besides different gray-scale intensities to represent the measured quantity we are interested in. As a first example, Fig. 4-a depicts the number $|\tau_J(K)|$ of the traffic relations mapped into each traffic aggregate $\tau_J(K)$. Looking at the plot and choosing a particular class K, every point of the vertical line represents therefore the number of TRs within



(a) **(b)**

Fig. 4. **(a)** Number of Traffic Relations $|\tau_J(K)|$ and **(b)** Number of TCP Flows within each Traffic Aggregate



(a) **(b)**

Fig. 5. **(a)** TCP Flows Size Distribution of TAs (Class $K = 3$) and **(b)** Interarrival Time Mean.

each of the K possible TA. Given the partitioning algorithm used, it is straight-forward to understand the reason why the larger J is, the larger is the number of TRs within the same TA, as the gray gradient clearly shows. Contour lines are shown for $|\tau_J(K)| \in \{1, 10, 100, 1000, 10000\}$ as reference values; it can be gathered that the bottom white-colored zone of the plot is constituted by *fixed* slots (i.e., $|\tau| = 1$), whereas the highest slot ($J = K$) has always $|\tau_K(K)| > 100,000$, $\forall K$.

A. Traffic Aggregate Bytewise Properties

The number of TRs within TAs spread smoothly; let us see how this is reflected on the number of TCP flows withing each TA $\Sigma_{(c,s)\in\tau_J(K)}|\mathcal{T}(c, s)|$, shown in Fig. 4-b. Quite surprisingly, we observe that also the number of TCP connections within TA shows an almost similar spreading behavior: the larger number of TRs within a TA, the larger number of TCP flows within the same TA. Indeed, the smoothed upper part of the plot (i.e., roughly, slots $J > K/2$) is represented by TAs with a number of flows larger than 5,000; instead, TAs composed by few TRs contain a much smaller number of TCP connections. Indeed, it must be pointed out that there are exception to this trend, as shown by the "darker" diagonal lines (e.g., those ending in slot $J = 25$ or $J = 54$, considering class $K = 100$). Probably, within these TAs there is one (or possibly more) TR which is built by a large number of short TCP flows. Coupling this result with the bytewise constraint imposed on TAs within the same class, we can state that the bottom region is constituted by a small number of long flows, while top region is constituted by a huge number of short flows.

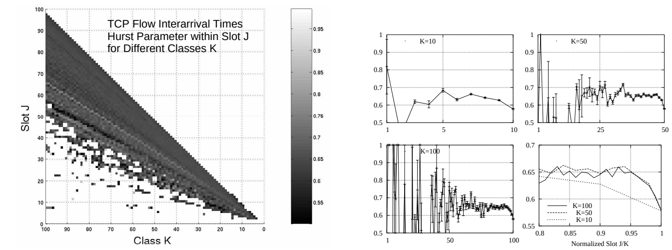
While this might be surprising at first, the intuition behind this clustering is that the largest TRs are built by heavier TCP-connections than the smaller TRs, i.e., TCP-elephants play a big role also in defining the TR weight. Therefore the splitting algorithm, by packing together larger TRs, tends also to pack together TCP-elephants. This approach is also useful in practice if we consider the fact that the "Pareto effect" is visible at different network layers and characterizes heavily the properties of the traffic. This induces to study separately traffic made of TA-elephants and TA-mice and the described procedure, based on traffic volume, is also able to do so. However, the gained result do not exclude the presence of TCP-mice in the bottom TAs, nor the presence of TCP-elephants in the top TAs; therefore, to further ensure that the clustering property of TCP-elephants in the first slots holds, we investigated how the TCP flow size distributions change in the different aggregates. Fig. 5-a plot the empirical flow size distribution for each TAs, considering class $K = 3$. As expected, i) TCP-elephants are evidently concentrated in lower slots, as shown by the heavier tail of the distribution; ii) the TCP-elephants presence decreases as the slot increases, i.e., when moving toward higher slots. The complementary cumulative distribution function $P\{X > x\}$, shown in the inset, confirms this trend. Therefore the adopted aggregation criterion induces a division of TCP-elephants and TCP-mice into different TAs. In the following, we will use for convenience the terms TA-mice and TA-elephants to indicate TA constituted (mostly) by TCP-mice and TCP-elephants flows respectively.

B. Inspecting TCP Interarrival Time Properties within TAs

The result shown in previous section has clearly important consequences when studying the TCP flow arrival process of different aggregates.

Let us first consider the mean interarrival time of TCP flows within each TA, shown in Fig. 5-b. Intuitively, TA-mice have a large number of both TR and TCP flows, and therefore TCP flow mean interarrival time is fairly small (less than 100ms). This is no longer true for TA-elephants, since a smaller number of flows has to carry the same amount of data over the same temporal window. Therefore, the mean interarrival time is much larger (up to hours). We recognize that a possible problem might arise, affecting the statistical quality of the results: the high interarrival time may be due to non-stationarity in the TA-elephants traffic, where TCP flows may be separated by long silence gap. This effect becomes more visible for large values of K. We will try to underline whether the presented results are affected by this problem in the remaining part of the analysis. Let us now consider the Hurst parameter $h(J, K)$ measured considering the interarrival time of TCP flows within TAs. For each TA, we performed the calculation of $h(J, K)$ using the wavelet-based approach developed in [13] and usually referred to as the *AV* estimator, which has emerged as one of the best estimators. The results are shown in Fig. 6-a, which shows that the Hurst parameter tends to decrease for increasing slot, i.e., the TA-mice show $h(J, K)$ smaller than TA-elephants. Whenever either the confidence interval is too large or the series is not stationary, the corresponding $h(J, K)$ are not reported in the plot. Still, the increase in the Hurst parameter is visible for TA-elephants.

To better show this property, Fig. 6-b presents detailed plots of $h(J, K)$ for $K \in \{10, 50, 100\}$. It can be observed that the Hurst parameter always tends to decrease when considering the TA-mice slots, while it becomes unreliable for TA with



(a) **(b)**

Fig. 6. **(a)** Interarrival Time Hurst Parameter of TCP Flows within all TAs and **(b)** for Selected Classes $K \in \{10, 50, 100\}$

few TRs, as testified by the larger confidence intervals. This is particularly visible when considering the $K = 100$ class. In the bottom-right plot, finally, we report a detail of the decaying feature of the $h(J, K)$ value for large J. In the x-axis, the J/K value is used, so that to allow a direct comparison among the three different classes. Notice that the last class, the one composed by many, small TRs which are aggregation of small TCP flows, always exhibits the same Hurst parameter. Therefore, the most important observation, trustable due to good confidence interval, is that we are authorized to say that TA-mice behavior is driven by TCP-mice. Similarly, TA-elephants are driven by TCP-elephants. Moreover, the interarrival process dynamic in TA-elephants and TA-mice are completely different in nature because light TRs tend to contain a relatively small amount of data carried over many small TCP flows, which do not clearly exhibit LRD properties. On the contrary, TCP-elephants seem to introduce a more clear LRD effects in the interarrival time of flows within TA-elephants.

A possible justification of this effect might reside in the different behavior the users have when generating connections: indeed, when considering that TCP-mice are typically of Web browsing, the correlation generated by Web sessions tends to vanish when a large number of (small) TRs are aggregated together. On the other side, the TCP-elephants, which are rare but not negligible, seem to be generated with a higher degree of correlation so that i) TRs are larger, ii) when aggregating them, the number of users is still small (see, for an example, the τ_D aggregate in Fig. 1). As a consequence, the effect on the TCP flow arrival time is similar to a ON-OFF source behavior, whose ON period is heavy-tailed and vaguely equal to the file download period, which turns out to follow an heavy tailed distribution; besides, we do not care about the OFF period. This could be one of the possible causes of LRD properties at the flow level. Finally, TAs tends to aggregate several TRs, separating short term correlated connections (TA-mice) from low-rate ON-OFF connections with infinite variance ON time (TA-elephants). This clearly recall to the well know phenomena described in [7]; that is, the superposition of ON-OFF sources (as well as "packet trains") exhibiting the Noah effect (infinite variance in the ON or OFF period) produces the so called "Joseph effect": the resulting aggregated process exhibits self-similarity and in particular LRD properties. Besides, consider again the heavy-tailed distribution of the TCP flow number within TRs, shown earlier in Fig. 2. If we further consider TRs as a superset of web sessions, we gather the same result stated in [14], that is, the $M/G/\infty$ effect with infinite variance service time distribution.

IV. Conclusions

In this paper we have studied the TCP flow arrival process, starting from the aggregated measurement taken from our campus network; specifically, we performed a live collection of data directly at the TCP flow level, neglecting therefore the underlaying IP-packet level. Two layered high-level traffic entities were defined: i) traffic relations, which are constituted by all TCP flows with the same destination, and ii) traffic aggregates of traffic relations. The algorithm used to create traffic aggregates split the original trace into sub-traces mainly made of either TCP-elephants or TCP-mice.

This permitted to gain some interesting insights on the TCP flow arrival process. First, we have observed, as already known, that long range dependence in the TCP flow arrival process can be caused from the fact that the number of flows within a traffic aggregate is heavy-tailed. In addition, the traffic aggregate properties allowed us to see that TCP-elephant aggregates behave like ON-OFF sources characterized by an heavy-tailed activity period, which cause LRD as well. Besides, we were able to observe that LRD vanishes for TCP-mice aggregates: this strongly suggests that even ON-OFF behavior is responsible of the LRD at TCP level.

References

[1] R. Caceres, P. Danzig, S. Jamin, D. Mitzel, *Characteristics of Wide-Area TCP/IP Conversations*, ACM SIGCOMM, Sep.1991, pp. 101–112.

[2] P. Danzig, S. Jamin, R. Caceres, D. Mitzel, D. Estrin, *An Empirical Workload Model for Driving Wide-Area TCP/IP Network Simulations*, Internetworking: Research and Experience, Vol.3, No.1, pp.1–26, 1992

[3] V. Paxons, *Empirically Derived Analytic Models of Wide-Area TCP Connections*, IEEE/ACM Tran. on Net., Vol.2, pp.316–336, Aug. 1994

[4] V. Paxson and S. Floyd, *Wide-area Traffi c: The Failure of Poisson Modeling*, IEEE/ACM Tran. on Net., Vol.3, No.3, pp.226–244, Jun. 1995

[5] W.E. Leland, M.S. Taqqu, W. Willinger, V. Wilson, *On the Self-Similar Nature of Ethernet Traffi c (Extended version)*, IEEE/ACM Tran. on Net., Vol.2, No.1, pp.1–15, Jan. 1994

[6] A. Feldmann, *Characteristics of TCP Connection Arrivals*, Park and Willinger (editors) *Self-Similar Network Traffi c and Performance Evaluation*, Wiley-Interscience, 2000

[7] W. Willinger, M.S. Taqqu, R. Sherman, D.V. Wilson, *Self-Similarity through High Variability: Statistical Analysis of Ethernet LAN Traffi c at the Source Level*, IEEE/ACM Tran. on Net., Vol.5, No.1, pp.71–86, Jan. 1997

[8] M. Mellia, A. Carpani, R. Lo Cigno, *Measuring IP and TCP behavior on Edge Nodes*, IEEE Globecom, Taipei (TW), Nov 2002 Vol. 3 pp. 2563–2567

[9] `DiaNa` home page `http://www.tlc-networks.polito.it/diana`

[10] A. Erramilli, O. Narayan and W. Willinger, *Experimental Queueing Analysis with Long-Range Dependent Packet Traffi c*, IEEE/ACM Tran. on Net., Vol.4, No.2, pp.209–223, 1996

[11] N. Hohn, D. Veitch and P. Abry, , *Does Fractal Scaling at the IP Level depend on TCP Flow Arrival Processes?*, 2nd Internet Measurement Workshop, Marseille, Nov. 2002

[12] S.C. Graves, A.H.G. Rinnooy Kan and P.H. Zipkin, *Logistic of Production and Inventory*, Nemhauser and Rinnoy Kan (editors), *Handbooks in Operation Research and Management Science*, Vol.4, North-Holland, 1993.

[13] P. Abry, P. Flandrin, M.S. Taqqu and D.Veitch, *Self-similarity and Long-Range Dependence Through the Wavelet Lens*, In *Long Range Dependence: Theory and Applications*, Doukhan, Oppenheim, 2000

[14] A. Feldmann, A. Gilbert, W. Willinger and T. Kurtz, *The Changing Nature of Network Traffi c: Scaling Phenomena*, Computer Communication Review 28, No.2, April 1998 Vol.28 pp.5–29