Passive Identification and Analysis of TCP Anomalies

Marco Mellia, Michela Meo, Luca Muscariello, Dario Rossi
Dipartimento di Elettronica, Politecnico di Torino, Italy

Abstract— **In this paper we focus on passive measurements of TCP traffic, main component of nowadays traffic. We propose a heuristic technique for the classification of the anomalies that may occur during the lifetime of a TCP flow, such as out-of-sequence and duplicate segments. Since TCP is a closed-loop protocol that infers network conditions by means of losses and reacts accordingly, the possibility of carefully distinguishing the causes of anomalies in TCP traffic is very appealing, since it may be instrumental to the deep understanding of TCP behavior in real environments and to protocol engineering as well. We apply the proposed heuristic to traffic traces collected at both networks edges and backbone links. By studying the statistical properties of TCP anomalies, we find that their aggregate exhibits Long Range Dependence phenomena, but that anomalies suffered by individual long-lived flows are on the contrary uncorrelated. Interestingly, no dependence to the actual link load is observed.**

I. Introduction

In the last ten years, the interest in data collection, measurement and analysis to characterize Internet traffic behavior increased steadily. Indeed, by acknowledging the failure of traditional modeling paradigms, the research community focused on the analysis of the traffic characteristics with the twofold objective of i) understanding the dynamics of traffic and its impact on the network elements and ii) finding simple, yet satisfactory, models for the design and planning of packet-switched data networks, like the Erlang teletraffic theory in telephone networks.

By focusing on passive traffic characterization, we face the task of measuring Internet traffic, which is particularly daunting for a number of reasons. First, traffic analysis is made very difficult by the correlations both in space and time, which is due to the closed-loop behavior of TCP, the TCP/IP client-server communication paradigm, and the fact that the highly variable quality provided to the end-user influences her/his behavior. Second, the complexity of the involved protocols, e.g. TCP itself, is such that a number of phenomena can be studied only if a deep knowledge of the protocol details is exploited. Finally, some of the traffic dynamics can be understood only if the forward and backward directions of flows are jointly analyzed – which is especially true for the detection of erratic flows behavior. Starting from [1], where a simple but efficient classification algorithm for out-of-sequence TCP segments is presented, this work aims at identifying and analyzing a larger subset of phenomena, including, e.g., network duplicates, unneeded retransmissions and flow control mechanisms triggered or suffered by TCP flows.

The proposed classification technique has been applied to a set of real traces collected at different measurement points. Results on both the network core and at the network edge show that the proposed classification allows to inspect a plethora of interesting phenomena: e.g., the limited impact of the daily load variation on the occurrence of anomalous events, the surprisingly large amount of network reordering, the correlation among different phenomena, to name a few.

II. Methodology

The methodology adopted in this paper furthers the approach followed in [1], in which the authors present a simple classification algorithm of out-of-sequence TCP packets and apply it to analyze the Sprint backbone traffic; the algorithm discriminates among out-of-sequence segments due to i) necessary or unnecessary packet retransmissions by the TCP sender, ii) network duplicates, or iii) network reordering. Similarly, we adopt a passive measurement technique and, building over the same idea, we complete the classification rule to include other event types. Besides, we not only distinguish among the other possible *kinds* of out-of-sequence or duplicate packets, but we also focus on the *cause* that actually triggered the segment retransmission by the sender.

As previously mentioned, we assume that both directions of a TCP flow are available: thus, both data segments and ACKs are analyzed; moreover, both the IP and TCP layers are exposed, so that the TCP sender status can be tracked. Figure 1-a sketches the evolution of a TCP flow: connection setup, data transfer, and tear-down phases are highlighted. It is worth pointing out that the measurement point (or sniffer) can be located anywhere in the path between the client and the server; furthermore, as shown in the figure, the sniffer can be close to the end host when measurements are taken at the network edge (as it happens, e.g., on campus LANs), whereas this bias vanishes when the sniffer is deployed on the core of the network (as it happens in the case of backbone traces). Since we rely on the observation of both data and ACK segments to track TCP flow evolution, a problem may arise when asymmetric routing forces data and ACK segment to follow two different paths: in such cases, we ignore the "half" flows.

A TCP flow starts when the first SYN from the client is observed, and ends after either the tear-down sequence (the FIN/ACK or RST messages), or when no segment is observed for an amount of time larger then a given threshold[1].

[1]The tear-down sequence of flows under analysis is possibly *never* observed: to avoid memory starvation we use a 15-minutes timer, which is sufficiently large [2] to avoid discarding on-going flows.

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

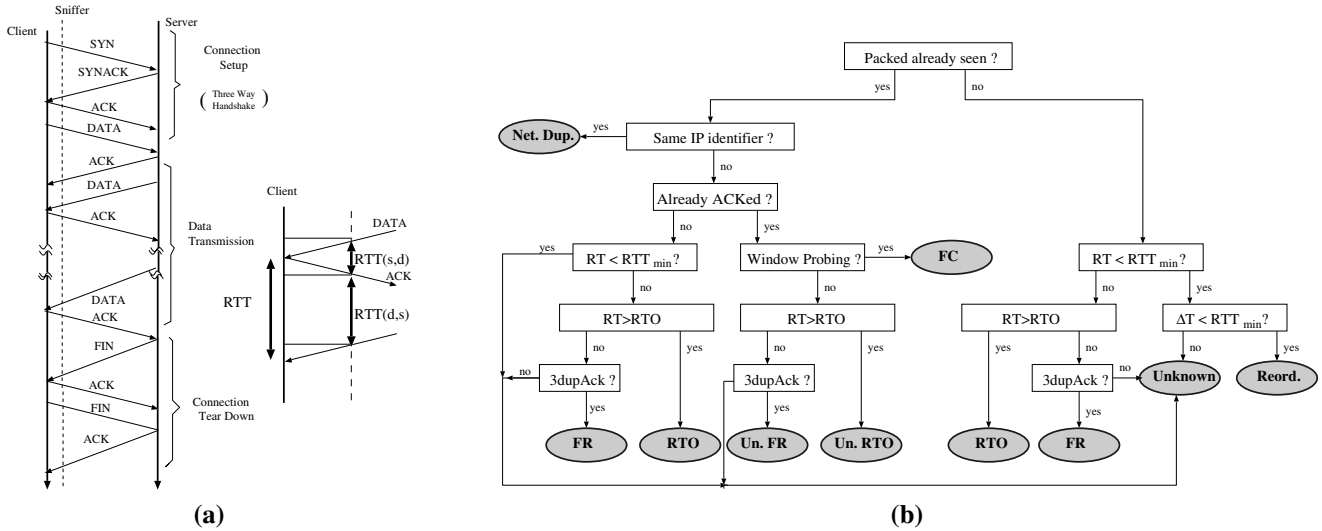

Fig. 1. Evolution of a TCP flow and the *RTT* estimation process **(a)** and anomalous events classification flow-chart **(b)**.

By tracking both flow directions, the sniffer correlates the sequence number of TCP data segments to the ACK number of backward received acknowledgments and classifies the segments as:

- *In-sequence*: if the sender initial sequence number of the current segment corresponds to the expected one;
- *Duplicate*: if the data carried by the segment have already been observed before;
- *Out-of-sequence*: if the sender initial sequence number is not the expected one, and the data carried by the segment have never been observed before.

The last two entries are symptomatic of some *anomaly* occurred during the data transfer: to further discriminate such anomalous events, we devise a fine-grained heuristic classification and implement it in Tstat [3], which we then use to analyze the collected data. The decision process followed to classify anomalous events makes use of the following variables:

- $RTT_{\min}$: Minimum *RTT* since the flow started;
- *RT*: Recovery Time is the time elapsed between the time the current anomalous segment has been observed and the time the segment with the *largest* sequence number has been received;
- ΔT: Inverted-packet gap is the difference between the observation time of the current anomalous segment and of the previously received segment;
- *RTO*: sender Retransmission Timer value, as in [4];
- $DupAck$: number of duplicate ACKs observed on the reverse path.

A. Heuristic Classification of TCP Anomalies

Following the flow diagram of Figure 1-b, we describe the classification heuristic. Given an anomalous segment, the process initially checks if the segment has already been seen by comparing its payload sequence number with those carried by segments observed so far: thus, the current segment can be classified as either duplicate or out-of-sequence. In the first case, following the left branch of the decision process, the IP identifier field of the current packet is compared with the same field of the original packet: in case of equality, then the anomalous packet is classified as **Network Duplicate** (Net. Dup.). Network duplicates may stem from malfunctioning apparatuses, mis-configured networks, or, finally, by unnecessary retransmissions at the link layer. Differently from [1], we classify as network duplicate all packets with the same IP identifier *regardless* of ΔT: indeed, there is no reason to exclude that a network duplicate may be observed at any time, and there is no relation between the *RTT* and the time a network can produce some duplicate packets.

When the IP identifiers are different, the TCP sender may have performed a retransmission. If all the bytes carried by the segment have already been acknowledged, then the segment has successfully reached the receiver, and this is an unneeded retransmission, which has to be discriminated further. Indeed, the *window probing* TCP flow control mechanism performs "false" retransmissions: this is done to force the immediate transmission of an ACK so as to probe if the receiver window *RWND*, which was announced to be zero on a previous ACK, is now larger than zero. Therefore we classify as retransmission due to **Flow Control** (FC) those segments for which: i) the sequence number is equal to the expected sequence number decreased by one, ii) the segment payload size is of zero length, and iii) the last announced *RWND* in the reverse ACK flow was equal to zero. This is a new possible cause of unneeded retransmissions which was previously neglected in [1].

Otherwise, the unnecessary retransmission could have been triggered because either a Retransmission Timer (*RTO*) has fired, or the Fast Retransmit mechanism has been triggered. We identify three situations: i) if the recovery time is larger than

the retransmission timer ($RT > RTO$), the segment is classified as an **Unneeded Retransmission by *RTO*** (Un. *RTO*); ii) if 3 duplicate ACKs have been observed, the segment is classified as an **Unneeded Retransmission by Fast Retransmit** (Un. FR); iii) otherwise, if none of the previous conditions holds, we do not know how to classify this segment and we are forced to label it as **Unknown** (Unk.). Unneeded retransmissions may be due to a misbehaving source, a wrong estimation of the *RTO* at the sender side, or, finally, to an ACK loss on the reverse path; however, distinguishing among these causes is impossible by means of passive measurements.

Let us now consider the case of segments that have already been seen but have not been ACKed yet: this is possibly the case of a retransmission following a packet loss. Indeed, given the recovery mechanism adopted by TCP, a retransmission can occur only after at least a *RTT*, since duplicate ACKs have to traverse the reverse path and trigger the Fast Retransmit mechanism. When the recovery time is smaller than $RTT_{\min}$, then the anomalous segment can only be classified as **Unknown**[2]; otherwise, it is possible to distinguish between **Retransmission by Fast Retransmit** (FR) and **Retransmission by *RTO*** (*RTO*) adopting the same criteria previously used for unneeded retransmissions. Retransmissions of already observed segments may be due to data segments loss on the path from the measurement point to the receiver, and to ACK segments delayed or lost before the measurement point.

Finally, let us consider the right branch of the decision process, which refers to out-of-sequence anomalous segments. Out-of-sequence can be caused either by the retransmission of lost segments, or network reordering. Since retransmissions can only occur if the recovery time *RT* is larger than $RTT_{\min}$, by double checking the number of observed duplicate ACKs and by comparing the recovery time with the estimated *RTO*, we can distinguish retransmissions triggered by *RTO* and FR or we can classify the segment as an unknown anomaly. On the contrary, if *RT* is smaller than $RTT_{\min}$, then a **Network Reordering** (Reord) is identified if the inverted-packet gap ΔT is smaller than $RTT_{\min}$. Network reordering can be due to either load balancing on parallel paths, or to route changes, or to parallel switching architectures which do not ensure in-sequence delivery of packets [5].

B. $RTT_{\min}$ and RTO Estimation

The algorithm described so far needs to set some thresholds from the packet trace itself based on some parameter estimation whose values may not be very accurate, or even valid, when classifying the anomalous event. Indeed, all the measurements related to the *RTT* estimation are particularly critical,. *RTT* measurement is updated during the flow evolution according to the moving average estimator standardized in [4]: given a new measurement m of the *RTT*, we update the estimate of the average *RTT* by mean of a low pass filter

[2]In [1] authors use the *average RTT*; however, being each *RTT*possibly different than the average *RTT*, and in particular *smaller*, we believe that the use of the average *RTT* forces a uselessly larger amount of unknown.

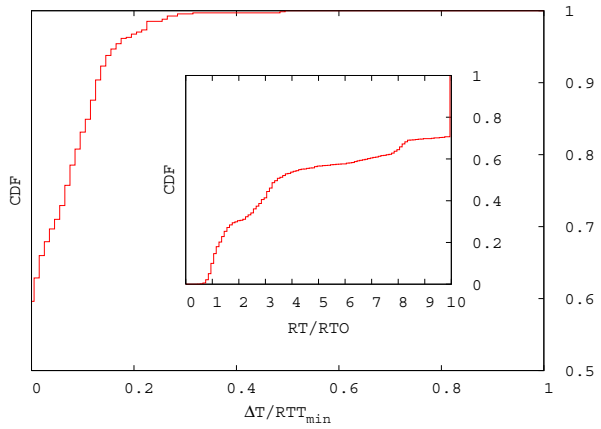

Fig. 2. CDFs of the ratio between the inverted packet gap ΔT over the estimated $RTT_{\min}$ (outset) and of the ratio between the recovery time *RT* over the actual *RTO* estimation (inset).

$E[RTT] = (1-\alpha)E[RTT] + \alpha\ m$ where $\alpha \in (0,1)$ is equal to $1/8$.

Since the measurement point is co-located neither at the transmitter nor at the receiver, the measure of *RTT* is not directly available. Therefore, to get an estimate of the *RTT* values, we build over the original proposal of [1]. In particular, denote by $RTT(s,d)$ the *Half path RTT sample*, which represents the delay at the measurement point between an observed data segment flowing from source s to destination d and the corresponding ACK on the reverse path, and denote by $RTT(d,s)$ the delay between the ACK and the following segment, as shown in Figure 1-a.

From the measurement of $RTT(s,d)$ and $RTT(d,s)$ it is possible to derive an estimate of the total round trip time as $RTT = RTT(s,d) + RTT(d,s)$; given the linearity of the expectation operator $E[\cdot]$, the estimation of the average round trip time $E[RTT] = E[RTT(s,d)] + E[RTT(d,s)]$ is unbiased. Furthermore, the *RTT* standard deviation, std(*RTT*), can be estimated following the same approach, given that in our scenario $RTT(s,d)$ and $RTT(d,s)$ are independent measurements. Finally, given $E[RTT]$ and std(*RTT*), it is possible to estimate the sender retransmission timer as in [4], and the minimum *RTT*:

$$\begin{aligned} RTO &= \max(1, E[RTT] + 4\text{std}(RTT)) \\ RTT_{\min} &= \min(RTT(s,d)) + \min(RTT(d,s)) \end{aligned}$$

In general, since $RTT_{\min} \leq \min(RTT)$, the latter equation gives a conservative estimate of the real minimum *RTT*, leading to a conservative algorithm that increases the number of anomalies classified as unknown rather than risking some mis-classifications.

In the following, a set of measurements are presented to inspect the impact of $RTT_{\min}$ and *RTO* estimation. The former is involved in the classification of the network reordering anomalies that may occur when identifying two out-of-sequence segments separated by a time gap smaller

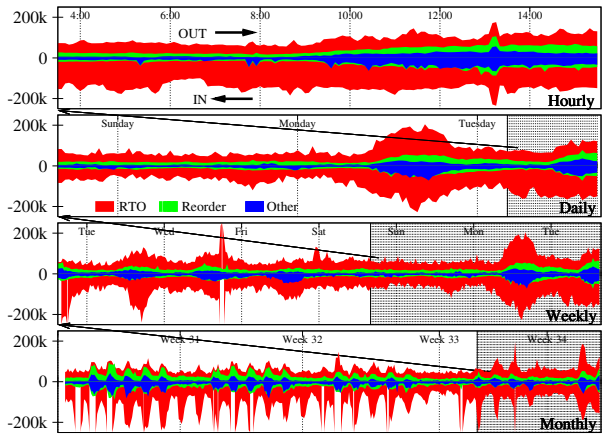

Fig. 3. Amount of incoming (bottom y-axis) and outgoing (top y-axis) anomalies at different timescales.

Fig. 4. Anomalies percentage normalized over the total traffic (top) and anomalies breakdown (bottom).

than $RTT_{\min}$. The outer plot of Figure 2 reports the typical[3] Cumulative Distribution Function (CDF) of the ratio between the inverted packet gap ΔT and the value of the $RTT_{\min}$, when considering anomalous events classified as network reordering only. It is easy to gather that ΔT is much smaller than the $RTT_{\min}$, which suggests that i) the initial choice of $RTT_{\min}$ is appropriate, and that ii) the conservative estimation of $RTT_{\min}$ does not affect the classification.

The inset of Figure 2 reports the CDF of the ratio between the actual Recovery Time *RT* and the corresponding estimation of the *RTO* relative to retransmissions by *RTO* events only. The CDF confirms that $RT > RTO$ holds, which is a clear indication that the estimation of the *RTO* is not critical. Moreover, it can be noted that about 50% of the cases have a recovery time which is more than 5 times larger than the estimated *RTO*. This apparently counterintuitive result is due to the *RTO* back-off mechanism implemented by TCP (but not considered by the heuristic), which doubles the *RTO* value at every retransmission of the same segment. Not considering the back-off mechanism during the classification leads to a robust and conservative approach.

[3]Since the result is similar across all the analyzed traces, we report the results referring to a single dataset, namely GARR'05 traces.

III. Measurement Results

In this section we present results obtained through traffic traces collected from different networks. The first one is a packet-level trace gathered from the Abilene Internet backbone, publicly available on the NLANR Web site [6]. The trace, which is 4 hours long, has been collected on June 1st, 2004 at the OC192c Packet-over-SONET link from Internet2's Indianapolis node toward Kansas City. The second measurement point is located on the GARR backbone network [7], the nation-wide ISP for research and educational centers, that we permanently monitor using Tstat. The monitored link is a OC48 Packet-over-SONET from Milano-1 to Milano-2 nodes, and statistics reported in the following refer to the month of August 2005. The third measurement point is located at the sole egress router of Politecnico campus LAN, which behaves thus as an Internet stub. It is a OC4 AAL5 ATM link from Politecnico to the GARR POP in Torino. Traces show different *routing symmetry*: while the Politecnico and GARR traces reflect almost 100% of traffic symmetry, only 46% of Abilene traffic is symmetric; this is of particular relevance, since Tstat relies on the observation of both data and ACK segments to track TCP flow evolution.

In the following, we will *arbitrarily* use "In" and "Out" tags to discriminate between the traffic directions of GARR and Abilene traces: since the measurement point is located in the backbone, neither an inner nor an outer network regions actually exists.

A. Statistical Analysis of Backbone Traffic

Figure 3 depicts the time evolution of the volume of anomalous segments classified by the proposed heuristic technique applied to the GARR traffic; in order to make plots readable, only the main causes of anomalies, i.e., *RTO* and network reordering, are explicitly reported, whereas the other kinds of anomalies are aggregated. Measurements are evaluated with different granularities over different timescales: each point corresponds to a 5 minute window in the hourly plot, a 30 minute window for both daily and weekly plots and a 2 hour interval in the monthly plot. Both link directions are displayed in a single plot: positive values refer to the "Out" direction, while negative to "In". The plot of the entire month clearly shows a night-and-day trend, which is mainly driven by the link load. The trend is less evident at finer time scales, allowing us to identify stationary periods during which advanced statistical characterization can be performed. The same monthly dataset is used in Figure 4, which reports in top plot the volume of anomalies normalized over the total link traffic, and in bottom plot the anomaly breakdown, i.e., the amount of anomalies of each class normalized over the total number of anomalous events. Labels report, for each kind of anomaly, the average occurrence over the whole period obtained aggregating both traffic directions. The amount of *RTO* accounts for the main part of anomalous events, being almost 70%. This is not surprising, since *RTO* events correspond to the most likely reaction of TCP to segment losses: indeed, Fast Retransmit is not commonly triggered, since only long TCP flows, which

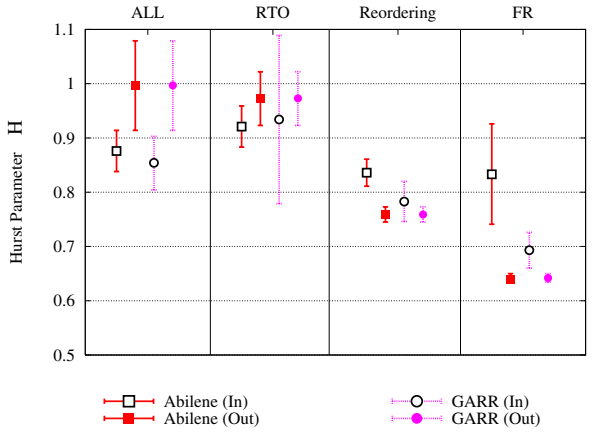

Fig. 5. Hurst parameter estimate for different anomalies and traces.

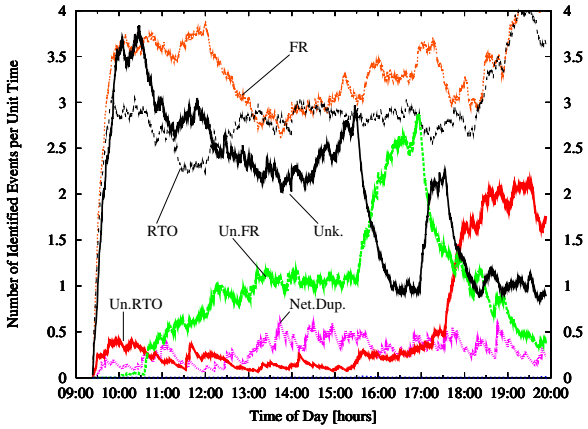

Fig. 6. Time plot of anomalous events for the selected elephant flow.

represent a small portion of the total traffic, can actually use this feature [9]. Packet reordering is also present in a significant amount. The average total amount of anomalous segments is about 5% considering Out traffic direction, and about 8% considering In direction, with peaks ranging up to 20%.

Besides the precise partitioning of such events, it is interesting to notice that the periodical trend exhibited by the *absolute* amount of anomalies observed early in the monthly plot of Figure 3, almost completely vanishes when considering the *normalized* amount shown in Figure 4. Thus, while the total amount depends on the traffic load, the percentage of anomalies seems quite independent on the load. This is an interesting finding that clashes with the usual assumption that the probability of anomalies and segment losses increases with load. Very similar results were also obtained for the Abilene trace, confirming that not only the percentage of anomalies over the total amount of traffic, but also the anomaly breakdown remain steadily the same even when the amount of anomalies over the link load exhibits large fluctuations. An intuition of the possible cause of such counterintuitive result relates it to the greedy nature of TCP, which pushes the instantaneous offered load to 1, therefore producing packet losses. Indeed, even if the average offered load is much smaller during off-peak periods, congestion may arise on bottleneck links (i.e., possibly at the network access) when there are active TCP flows.

Another interesting statistics we can look into is the correlation structure of the time series counting the number of anomalies per time unit. Though this could be presented in many different ways, we decided to use the Hurst parameter H as a compact index to identify the presence of Long Range Dependence (LRD) in the series. The estimation of H has been carried over whenever it has been possible to identify stationary time windows. In particular, this holds for the two backbone links when considering periods of a few hours, while this was not verified on campus LAN traces, due to the much smaller number of events. The estimate of the Hurst parameter H, measured through the wavelet based estimator [10], is depicted in Figure 5 for different kinds of anomalies, measurement points and traffic directions. Roughly speaking, $H = 0.5$ hints to uncorrelated processes, while increasing values of $H \in [0.5, 1]$ are symptomatic of increasing correlation in the process, with values $H \geq 0.7$ usually considered typical of LRD. In all cases, H is considerably higher than 0.5, meaning that these series are LRD. Moreover, H for the Reordering and FR series is smaller than for the *RTO* series, for all traces and traffic directions.

B. Analysis of a Single Elephant Connection

Other interesting observations can be drawn by focusing on a specific network path, where it can be assumed that network setup is homogeneous: in particular, we consider the longest TCP flows that has been measured on our campus LAN. The longest "elephant" that we ever observed corresponds to a large file download from a host in Canada which lasted about ten hours. Figure 6 plots a set of time series regarding the flow anomalies, where each series corresponds to the occurrence of events of one of the anomalies identified by the proposed heuristic technique. First of all, we point out that, while each time series is non-stationary, the aggregated series is stationary. Then, observe that the largest number of anomalies refers to *RTO* and FR, and this holds for the whole flow lifetime. It is worth noticing that typically, while the number of *RTO* increases, the number of FR decreases (and vice-versa): high numbers of *RTO* probably correspond to high levels of congestion, driving the TCP congestion control to shrink the sender congestion window, so that the chance of receiving three dup-ACKs is small. Similar behaviors were observed for other very long elephants not shown here for the sake of brevity.

No network reordering has been identified during the whole flow lifetime. In our experiments we observed that some flows do not suffer from network reordering at all, while others present several network reordering events: this behavior suggests that network reordering is path dependent, and a similar reasoning applies to network duplicate events.

Some of the anomalies could not be classified other than Unknown. Investigating further the reasons of such misiden-

tification, we suppose that either the sender was triggering FR with a number of Duplicate ACKs smaller than 3, or that the *RTO* value estimated by the sender was smaller than the *RTO* estimation at the measurement point. Notice again that, as we already mentioned, our classification strictly follows the TCP standards, avoiding mis-classifications, at the price of the increase of unidentified events.

We now focus on the fluctuation of the rate and the statistical properties of the counting process of i) the received packets per time unit and ii) the anomalous events per time unit, when the time unit is 100 m long. We analyze the scaling behavior of the data through the Multi Resolution Analysis (MRA) [11] with the code provided in [12]; according to MRA, both these processes are stationary: moreover, from our measurements, stationarity usually holds for long flows in general. Figure 7 depicts the log-scale diagram for both the considered time series to estimate the scaling exponents [10]. The diagram reports the estimate of the correlation of the considered process: an horizontal line is symptomatic of an uncorrelated process, while linear slopes for large scales have to be read as signs of high correlation. Therefore, from Figure 7 we observe that the anomalous events that TCP handles are *not* correlated. On the contrary, as already observed on traffic aggregates [10], we found the traffic generated by the elephant flow is *highly* correlated for the whole flow lifetime. The slope of this curve leads to an estimation of the parameter H about 0.74;

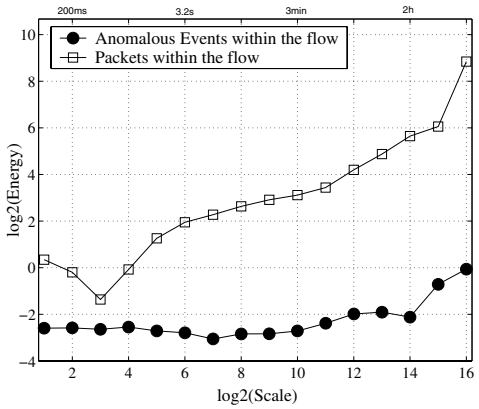

Fig. 7. Energy plots for the time series of the number of segments and the number of out of orders within the selected elephant flow.

IV. Conclusions and Discussion

In this paper, we have defined, identified and studied TCP anomalies, i.e., segments received either out-of-sequence or duplicate. We have extended and refined the heuristic originally proposed in [1], and applied it to real traffic traces collected at both network edges and core links. By studying the statistical properties of these phenomena, we can state few guidelines to understand the stochastic properties of TCP traffic. First, we observed that, though the absolute amount of anomalies highly depends on the link load, both its relative amount over the total traffic and the anomaly breakdown are almost independent from the current load. This interesting finding clashes with the usual assumption that the probability of anomalies and segment losses increases with load.

Second, we have shown that aggregated anomalies at any Internet link are correlated, as clearly shown by the estimation of the Hurst parameter H. Nevertheless, when considering individual flows, we observed that the anomalies arrival process is on the contrary uncorrelated, even though the general packet arrival process generated by the same flow is known to be highly correlated, as we experimentally verified. The correlation exhibited by the aggregated anomalies is intuitively tied to the duration of their causes (i.e., congestion periods, path reordering, etc.) which may last for large timescales. When considering an individual flow, the sub-sampling process induced by packet arrivals (i.e., the process observed by flow packets crossing a particular node/link/path), and the congestion control mechanisms adopted by TCP tend to uncorrelate the process of anomalous segments observed by the flow itself. Indeed, since the traffic composition comprises a large number of very short flows and a small number of long-lived flows, only a very small fraction of anomalous segments observed over a link is due to a single flow. Thus, we can conclude that, even though there is high correlation in the aggregate process, its impact on single flows is marginal.

V. Acknowledgments

This work was partly funded by the European Community "EuroNGI" Network of Excellence, and partly by the Italian MIUR PRIN project "MIMOSA". We would like to thank the GARR for allowing us to monitor some of their backbone links, the CESIT for their patience in supporting the monitoring at our campus LAN, and finally the people from NLANR for providing the research community their invaluable traces.

References

[1] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, D. Towsley, "Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone",*IEEE INFOCOM'03*, San Francisco, CA, Mar. 2003.

[2] G.Iannaccone, C.Diot, I.Graham, and N.McKeown, "Monitoring Very High Speed Links," *ACM Internet Measurement Workshop (IMW'01)*, San Francisco, CA, Nov. 2001.

[3] M. Mellia, R. Lo Cigno and F. Neri, "Measuring IP and TCP Behavior on Edge Nodes with Tstat," *Computer Networks*, Vol. 47, No. 1, pp. 1-21, Jan. 2005.

[4] V. Paxson an, M. Allman, "Computing TCP's Retransmission Timer," *IETF RFC 2988*, Nov. 2000.

[5] J.C.R. Bennett, C.C. Partridge and N. Shectman, "Packet Reordering is not Pathological Network Behavior," *IEEE/ACM Transactions on Networking*, Vol. 7, N. 6, pp.789-798, Dec. 1999.

[6] AbileneIII packet trace, `http://pma.nlanr.net/Special/ipls3.html`

[7] GARR Network, `http://www.noc.garr.it/mrtg/RT.TO1.garr.net`

[8] Tstat Web Page, `http://tstat.tlc.polito.it`

[9] M. Mellia, M. Meo and C. Casetti, "TCP Smart Framing: a Segmentation Algorithm to Reduce TCP latency," *IEEE/ACM Transactions on Networking*, Vol. 13, No. 2, pp. 316-329, Apr. 2005.

[10] P. Abry and D. Veitch. "Wavelet Analysis of Long-Range Dependent Traffic," *IEEE Transactions on Information Theory*, Vol. 44, No. 1, pp:2-15, Jan. 1998.

[11] D. Veitch and P. Abry, "A Statistical Test for the Time Constancy of Scaling Exponents," *IEEE Transactions on Signal Processing*, Vol. 49 No. 1, pp. 2325-2334, Oct. 2001.

[12] Darryl Veitch Home Page, Wavelet Estimation Tools, `http://www.cubinlab.ee.mu.oz.au/\~darryl`