Persistent Gbps Link Monitoring with Tstat

Dario Rossi
Politecnico di Torino – Dipartimento di Elettronica
dario.rossi@polito.it

Marco Mellia
Politecnico di Torino – Dipartimento di Elettronica
marco.mellia@polito.it

I. Introduction

Traffic measurement represents an indispensable and valuable tool for the analysis of nowadays telecommunication networks, as testified by the interest exhibited by several research groups [1], [2], [3]. Moreover, it is desirable for traffic measurement and analysis to be both *continuous* and *persistent*, since only these joint requirements allow to track important changes of the traffic patterns. On the other hand, transmission links bandwidth keep improving, at a seemingly inexorable rate: therefore, the analysis of the traffic is becoming more complex than ever.

Generally speaking, traffic characterization can be performed by means of either active (i.e., by sending and receiving probe traffic) or passive (i.e., by collecting packets flowing through networks links) measurements. The layered structure of the TCP/IP protocol suite requires the analysis of traffic at least at the IP (network), TCP/UDP (transport), and possibly Application (session) layers. Since the majority of Internet traffic is carried by TCP flows, the analysis of the traffic at the transport-layer is of particular interest, which poses a great deal of additional complexity compared to pure packet layer analysis: indeed, each layer-4 measurement usually requires to build and maintain the flow status for the whole flow lifetime.

This work focuses on the description and the benchmarking of `Tstat` [4], [5], as an example of *open source* and *passive* analysis tools able to provide advanced transport-layer statistics, featuring besides scalable and ever-lasting monitoring capabilities. Particularly, our aim is to assess what kind of links, and under which load, can be continuously and persistently monitored without compromising the complexity of the traffic analysis that has to be performed. By running several benchmarking tests using different real traffic traces, we will show that off-the-shelf hardware can easily be used to perform real-time transport-layer analysis of Gbps traffic.

II. Tstat

Despite we describe and benchmark a single software tool, we argue that the performance insights gathered will be valid to a more general extent: indeed, `Tstat` performs typical operations (such as flow status tracking, measurement computation, and statistical data generation) that *every* layer-4 measurement tool must perform. Started as evolution of `TCPtrace` [6], the tool analyzes either off-line packet traces (supporting several storage formats) as well as real-time traffic (using either common hardware and standard software libraries [7] or more sophisticated ad-hoc hardware [8]).

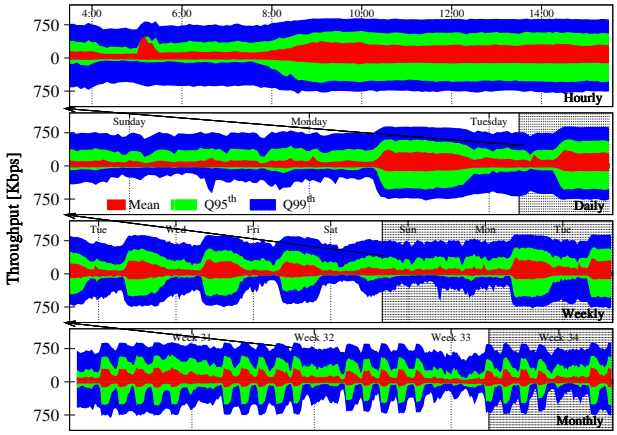

Fig. 1. RRD Output: Time-Varying Throughput Distribution

The software provides several measurements at the network layer, but its main focus is transport layer protocols analysis. Assuming that both forward and backward stream of packets are observed, `Tstat` can correlate them to infer advanced measurement indexes: indeed, rebuilding each TCP flow status allows:

- the derivation of *novel statistics*, such as the congestion window size, out-of-sequence and duplicated segments classification, whose complete list and description is available in [4],
- which are collected distinguishing both between *clients* and *servers* (i.e., hosts that actively open a connection and hosts that reply to the connection request),
- and also identifying *internal* and *external* hosts (i.e., with respect to an arbitrary partition of the network at the measurement point).

As output, `Tstat` builds histogram of measured indexes, dumping the collected distribution periodically, rather than dumping each single measured datum. The data produced by the on-line statistical analysis is ready to be visualized as either time plots or aggregated plots over different time spans. A complete transport layer log, which is useful for post-processing purposes, tracks all analyzed layer-4 flows by including all performance indexes for later post-processing. Finally, `Tstat` has been integrated with `RRDtool` [9]: the whole measurement indexes can now be stored as a Round Robin Database (RRD). Since RRD has fixed size, this allows ever-lasting live-capture without compromising the statistical relevance of the data, as it can be gathered by considering Figure 1, which shows the time evolution of the TCP-flow

throughput distribution on a backbone link that is persistently monitored. In order to make plots readable, only the average, 95^{th} and 99^{th} percentiles are explicitly reported; moreover, both link directions are displayed in a single plot, using either positive or negative values for a given direction. Measurements are evaluated with different granularities over different timescales: each point corresponds to a 5 minute window in the hourly plot (at the top), a 30 minute window for both daily and weekly plots and a 2 hour interval in the monthly plot (at the bottom). 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 to identify stationary periods during which `Tstat` allows to perform advanced statistical characterization.

III. Peformance Evaluation

This section, reporting an excerpt of `Tstat` benchmarking, shows that a rich and persisten transport-layer analysis is feasible with common hardware, even when the traffic rate exceeds the Gbps threshold. All experiments have been performed on a PC, running a GNU+Linux operating system with kernel version 2.4.29, sporting two Intel Xeon CPUs clocked at 2.40GHz with 512 KB cache, equipped with 3 GB of RAM memory and several Seagate 7200-rpm hard-disks. Though the aim is to provide insights on the performance of *on-line* traffic analysis, we consider *off-line* analysis of packet-level traces only, with the twofold intent of i) being able to run *batch* of tests on the same input as well as to ii) stress the tool to its limits: indeed, in our experience, the incoming backbone traffic rate has always been much *lower* than the pace of `Tstat` processing capabilities.

We present results obtained through two real backbone traffic traces collected from different networks, focusing either on the pure-processing performance (i.e., no output is actually produced but all the layer-4 computations are performed) or investigating what kind of link/traffic can be persistently monitored (i.e., only the RRD output is produced and no transport-layer log is available). The first packet-level trace, denoted in the following as ABILENE, has been gathered June the 1st, 2004 from the Abilene Internet backbone [10], on the OC192c Packet-over-SONET link between Internet2's Indianapolis and Kansas City nodes. Peculiar of this trace is the large presence of very high-speed transfers of long files, which generates very short spikes of intense load above 2.5 Gbps. Other quite anomalous traffic patters (e.g., large port scanning presence), are present as well; these are due to the experimental traffic ABILENE carries, and both form a *stress* scenario when considering layer-4 processing. The second packet-level trace has been gathered April the 15th, 2005 from GARR backbone network [11], on the OC48 Packet-over-SONET link (that we permanently monitor: results shown early in Figure 1 refer to this measurement point) between Milano-1 and Milano-2 nodes. GARR is the nation-wide ISP for research and educational centers, and is thus representative of *typical* today traffic in which business, home and research traffic share the same infrastructure.

Backbone Trace	Output Type	π [Kpps]	Elapsed [h:m:s]	CPU Time [h:m:s]	Speed-up
GARR	Null	252.35	0:18:33	0:16:16	8.06
	RRD	233.86	0:20:01	0:16:32	7.93
ABILENE	Null	255.79	2:00:42	1:26:30	1.49
	RRD	255.01	2:01:04	1:26:37	1.49

TABLE I
FEASIBILITY ANALYSIS OF PERSISTENT BACKBONE LINK MONITORING

Both GARR and ABILENE traces correspond to about two-hours long period and include 80 Bytes of packet headers only; GARR traces occupy 46 GB of storage whereas ABILENE amount to 36 GB of compressed data (141 GB uncompressed. Moreover, traces show different *routing symmetry*: while the GARR trace reflects 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 correctly track TCP flow evolution. Finally, the average load is quite different for the two traces: on average, GARR traffic rate is about 36 Kpps or 177 Mbps. ABILENE traffic rate is more than 6 times higher, resulting in an average rate slightly above 225 Kpps or 1.12 Gbps.

Table I reports some deeper performance figures relative to both ABILENE and GARR traces: total elapsed time, pure CPU time, packet processing rate π (totally processed packets over the total elapsed time) and processing *speed-up*, i.e., the ratio of the trace time-length over the CPU time required to complete the analysis. Results confirm that `Tstat` painlessly processes GARR trace: we point out that only 18 minutes were needed to process more than two-hour worth of traffic. Even in the ABILENE case, though the gain margin is scarcer, `Tstat` processing is faster than real-time traffic arrival. Recall that the ABILENE trace is compressed: thus, a significant portion of the CPU has been devoted to the input *decompression*. Indeed, more than one fourth of the CPU processing is wasted: this possibly entails that, roughly, the performance are *underestimated* of about 25%, which suggests a possible speed-up 2. Notice that the use of dedicated capture cards (mandatory to monitor OC192 links) will further lessen the CPU from IO processing, thus allowing for even more traffic to be processed, which finally confirms that a statistical rich and persistent Gbps link monitoring is feasible even with common hardware.

References

[1] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose and D. Towsley. "Inferring TCP Connection Characteristics Through Passive Measurements," *IEEE INFOCOM'04*, Hong Kong, March 2004
[2] L. Li, M. Thottan, B. Yao and S. Paul, "Distributed Network Monitoring with Bounded Link Utilization in IP Networks," *IEEE INFOCOM'03*,
[3] M. Jain, C. Dovrolis, "End-to-end Estimation of the Available Bandwidth Variation Range," *ACM SIGMETRICS'05*, Canada, Jun. 2005.
[4] M. Mellia and D. Rossi, `Tstat`, `http://tstat.tlc.polito.it`
[5] M. Mellia, R. Lo Cigno and F. Neri, "Measuring IP and TCP behavior on edge nodes with Tstat", *Computer Networks*, Jan. 2005.
[6] S. Ostermann, `TCPtrace`, `http://www.tcptrace.org`
[7] S. McCanne, V. Jacobson, `pcap`, `http://www.tcpdump.org`
[8] Endace, `http://www.endace.com`
[9] T. Oetiker, `RRDtools`, `http://people.ee.ethz.ch/~oetiker/webtools/rrdtool`
[10] AbileneIII, `http://pma.nlanr.net/Special/ipls3.html`
[11] GARR, `http://www.noc.garr.it/mrtg/RT.TO1.garr.net`