Estimating Server Load Based on its Correlation with TCP SYN Response Time

Luis de Pedro, Marta Martínez Redondo, Cristina Mancha, Jorge E. López de Vergara
High-Performance Computing and Networking research group,
Departamento de Tecnología Electrónica y de las Comunicaciones,
Escuela Politécnica Superior, Universidad Autónoma de Madrid
Francisco Tomás y Valiente, 11, 28049 Madrid
{luis.depedro, jorge.lopez_vergara}@uam.es,

***Abstract*—In this paper, we show that server load correlates with TCP SYN response time (SRT), that is, the time from SYN to SYN+ACK segments at the server side. For this, it applies the innovative approach of modeling the SRT with an appropriate α-stable heavy tail distribution. We also show that α-stable distribution parameters are related to the server load the connection is attempted to. This approach provides a non-intrusive way to estimate the server load, and it can be useful to identify if a problem in a distributed application is caused by the end system, or to distribute the load among servers. Finally, based on obtained results, we propose a method that estimates server load based on SRT.**

***Index Terms*—non-intrusive measurement, α-stable distribution, traffic analysis.**

I. Introduction

Network traffic has been widely monitored and analyzed to study the network behavior, with many research papers [1], [2], as well as several patents about this topic [3], [4]. Nevertheless, it can also be valuable to study end systems behavior. For instance, several zero-window announcements in a session show that an application is receiving data at a higher rate that the one it can process. Such a type of information can be very useful to identify the root cause of problems [5] in distributed systems, which are usually difficult to fix, partly because network and system departments typically report to a different manager, and none of them wants to take the blame of the miss-configuration.

In this paper, we focus on how the load of a server can be estimated based on the analysis of network traffic. In this way, it is possible to distinguish if a problem is caused by the network or by the server, therefore reducing the time to identify its root causes, or, at least, the so-called mean time to innocence (MTTI) [6].

Moreover, this analysis can also be useful to fix other problems that have appeared in distributed systems in recent years. For instance, cloud computation is being incrementally used by all kind of organizations, and load balancing intended to use resources as efficiently as possible is an active research topic [7]. Server Load is the input to a number of proposed scheduling algorithms [8], [9]. Load measuring without interfering significantly with applications in a server can be difficult, partly because the measurement also affects the load of the server, and partly due to previously referred organizational management.

ISBN 978-3-903176-28-7 © 2020 IFIP

The way in which we propose to estimate the server load with network traffic is based on TCP SYN Response Time (SRT). This is, the time it takes the server to answer a connection request [10], as we will see in following sections.

The contributions of this paper are manifold:

1) We present a novel method to estimate server load using a network-traffic-based, non-intrusive approach by analyzing SRT distribution characteristics.
2) We show that SRT varies along the day with the workload of the server, and its values are statistically distributed following a heavy-tailed distribution.
3) We demonstrate that this distribution can be accurately modeled by using an α-stable distribution. In contrast, other typical distributions, such as Gaussian or log-normal, do not fit well.
4) Finally, we also analyze the relation between α-stable distribution parameters and the server load.

The rest of the paper is organized as follows. First, we present a description of SRT and α-stable distributions, and how they can be used to model SRT. Then, we discuss the correlation of the statistical parameters to the server load. Experimental data, taken from real server traffic, is presented and discussed as well. Based on these results, we propose a method to estimate the server load. Finally, we conclude the paper remarking its key ideas and future research lines.

II. TCP SYN Response Time

A. SRT measurement

TCP protocol requires a connection set-up before transmitting or receiving any useful application-layer data. The complete connection set-up involves three segments: SYN, from client to server, SYN+ACK, from server to client, and ACK, once again from client to server. TCP protocol does not consider the connection established until all of them have been successfully received. This 3-way handshake, shown in Fig. 1, can be measured by using traffic probes [11] placed at a vantage point.

Any of the phases delay can be used as an estimation for Round Trip Time (RTT) [12], which has been used to estimate

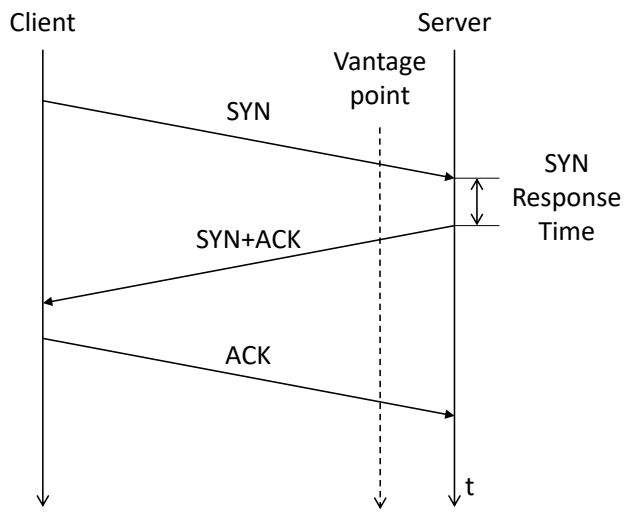

Fig. 1. SYN Response Time in the TCP handshake.

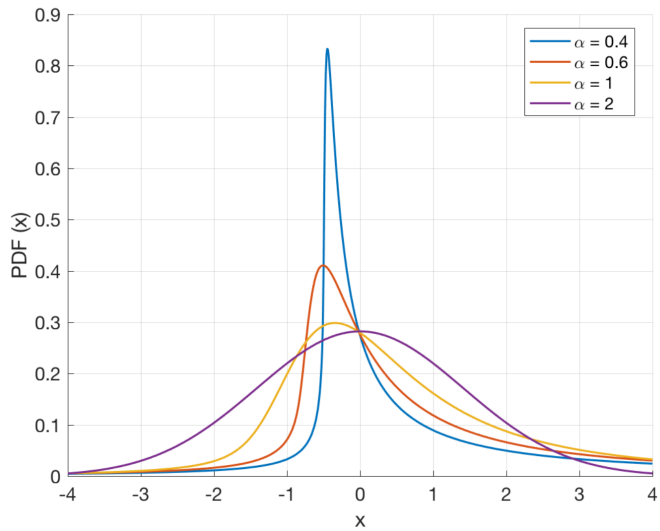

Fig. 2. α-stable distribution for β = 0.7, γ = 1.0 and δ = 0.0

network infrastructure issues [13]. However, it should be noted that, in contrast to these previous works, we are not going to measure neither the RTT nor the network status. What we present here is an innovative approach that uses SRT as an estimator of the server load. In this case, it is important to place the vantage point close to the server, so network influence on measured SRT is minimized.

Note also that other approaches, based on the number of SYN segments received by the server, do not provide an accurate measure of the server load, as the load depends both on this arrival rate and also on the service rate, which is usually unknown, and not necessarily related to the transmitted data.

B. α-stable distributions

To model SRT statistics, due to the existence of heavy tails in the measured values, we use an α-stable distribution. Note that heavy tail distributions first and second moments are not necessarily defined, so mean and variance may be useless in this context. This distribution has been used to model several traffic parameters such as bandwidth consumption in bits per second or packets per second, or even RTT, with better results than Gaussian or log-normal distributions [13], [14]. As far as we know, this distribution has not been used before to model SRT, although it is a natural extension of previous research.

From a mathematical point of view, α-stable distributions can be considered as an extension of Gaussian and, therefore, there is an equivalent to the central limit theorem for them. Actually, the sum of a number of α-stable distributions is also an α-stable distribution [15]. The main inconvenient when using this type of distributions is that there are no closed expressions available for the probability density function (PDF) as there are for Gaussian distributions. The equivalent is the expression for $E\{\exp(itX)\}$ in Eq. (1).

$$\begin{cases} \exp\{-\gamma^\alpha |t|^\alpha \big[1+ \\ \quad +i\beta \mathrm{tg}\big(\frac{\pi\alpha}{2}\big)\mathrm{sgn}(t)(|\gamma t|^{1-\alpha}-1)\big] + i\delta t\} \text{ if } \alpha \neq 1 \\ \exp\{-\gamma|t|\big[1+i\frac{2\beta}{\pi}\mathrm{sgn}(t)\log(\gamma|t|)\big] + i\delta t\} \text{ if } \alpha = 1 \end{cases} \tag{1}$$

The α-stable parameters in Eq. (1) are not directly related to mean or variance and can be transformed in several ways [15]. α is the stability index or characteristic exponent, β is the skewness, γ is the scale parameter and δ is the location parameter. Parameter α is related to the burstiness or kurtosis of the distribution. Fig. 2 shows the influence of α parameter in the distribution shape. Higher values of α indicate distributions closer to Gaussian (which is the case when $\alpha = 2$).

C. SRT statistical model

To check whether α-stable can model SRT we used real traffic from a *Fortune 500* company, obtained by a probe capturing traffic from a mirror port of a switch that is close to the server infrastructure in the data center, receiving connection requests from branch offices during business hours. The obtained SRT time series in the peak hour, when the branch offices start to work, is shown in Fig. 3. Note the high excursion of the SRT values, which indicates an asymmetric long-tailed distribution.

The α-stable fitting to the SRT data requires to choose a time window long enough to capture the statistical features of the traffic, but also short enough to consider the traffic stationary. Between one minute and one hour, we have not found noticeable differences in the fitting accuracy and we decided to use a 10-min. window to compare different distribution fittings. Other options are available, as mentioned, so we test several alternatives. Fig. 4 shows that α-stable distribution is by far the best choice to model SRT.

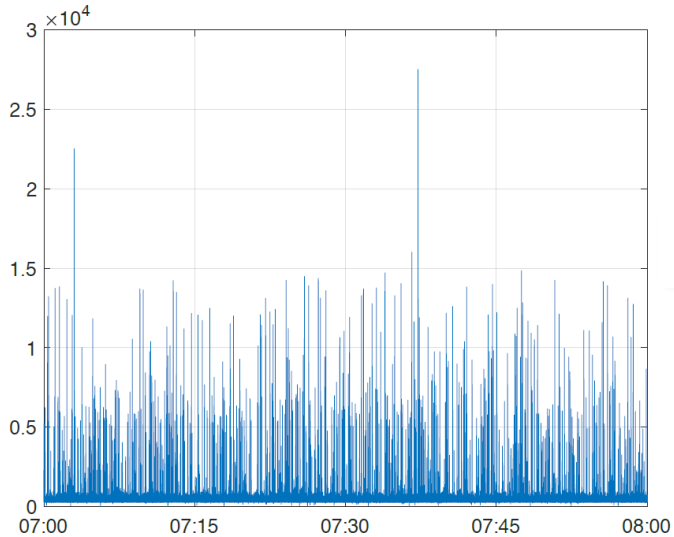

Fig. 3. SRT time series (in μs).

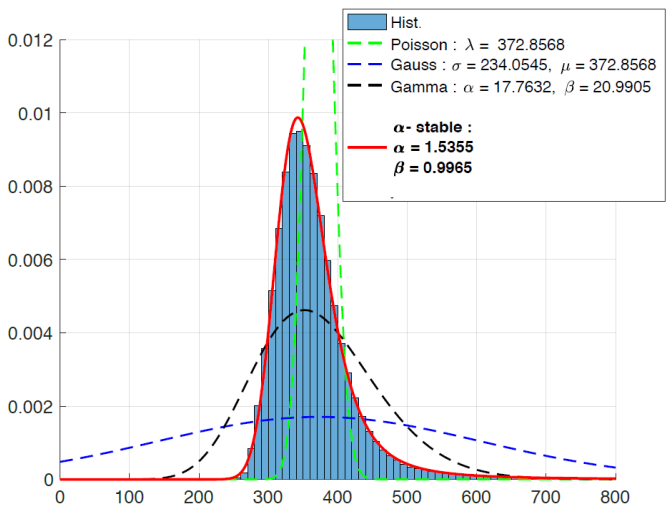

Fig. 4. SRT (in μs) distribution fittings.

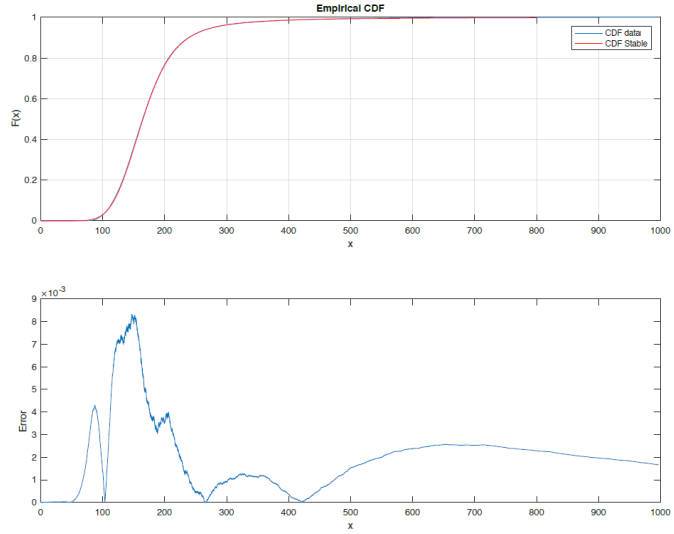

Fig. 5. CDF fitting absolute error.

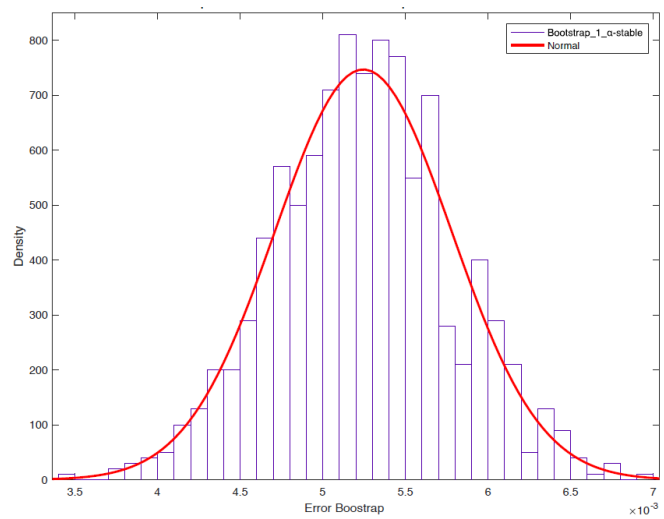

Fig. 6. CDF fitting absolute error PDF.

D. Model accuracy

To check the model accuracy, we used the Kolmogorov-Smirnoff approach [16]. This method uses the cumulative distribution function (CDF) error to estimate the distribution fitting accuracy. In Fig. 5 it can be seen the difference between the time series CDF and the α-stable distribution CDF in absolute value.

The maximum of the absolute CDF error is a measure of the accuracy for the fitting. However, to really assess the error, we have to break the dependence between the SRT time series and the fitting distribution. This is usually made by re-sampling both distributions and by getting again the CDF fitting error. This approach, known as *bootstrapping* [17] produces a random variable for the error that can be used as an estimation of the quality of the model. Fig. 6 shows the CDF error as a random variable. As shown, it resembles a Gaussian distribution with 0.52% mean, which is an excellent result that confirms that α-stable distributions provide a very good fit to model SRT.

E. SRT variability

Once the fitting model for the SRT is decided, we proceeded by analyzing the SRT average evolution during a long period. What we saw is presented in Fig. 7 for a whole week (from Monday to Sunday). SRT average per hour changes every day following a similar shape. The best explanation is that SRT average is following somehow the infrastructure utilization. During daytime, it has clearly higher values than during nighttime. In the weekend, given that the server did not receive connections, the SRT was similar to the nighttime. These facts lead to link SRT and server load. To check this hypothesis, we have set up a controlled environment, where we can measure

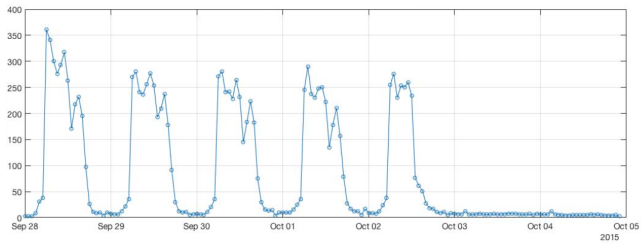

Fig. 7. SRT average per hour (in μs).

the SRT of connections sent to a server where we can set the load, as explained in the following section.

III. Server Load Correlation with SRT

In order to measure the correlation between SRT and server load we have set up an environment with a client and a server, both multithreaded and implemented in Python 3, with one thread per connection, connected through a Gigabit Ethernet network switch (code is available upon request). The server runs a Linux Kernel 3.2. The client sends TCP connection requests (SYN segments) to the server with Poisson rate $\lambda = 5$ connections/s. This approach with Poisson arrivals and its PASTA (Poisson arrivals see time averages) property avoids artifacts in the SRT measurement that could be caused if deterministic inter-arrival times were used [18]. After the TCP connection is established, the client downloads a random number of bytes (uniformly distributed from 0 to 11.5 Megabytes) from the server, and closes the connection (FIN) once the download is completed. SRT is measured by capturing the traffic with Wireshark in the server, where we also use the Linux `stress` command to simulate server load, and `mpstat` to measure the load of the system every two seconds. This command measures the CPU time used by the kernel (*sys*), applications (*usr*), etc. With this set-up, we expect the environment to be easily replicated by other researchers.

A. TCP connection inter-arrival time

To verify that SYN segments sent by the client are truly Poisson, we have analyzed their inter-arrival time when received at the sever. Fig. 8 shows the survival function of the inter-arrival time with logarithmic vertical axis. It can be seen that the distribution fairly follows an exponential function, so SYN packet arrival can be considered Poisson. We have also checked that the squared coefficient of variation is a value near 1, which also confirms the Poisson nature for the arrivals.

B. Server load experiment

We have run several times five different tests with different server loads, one without using the `stress` command, and other four in which we increase one worker on each test. Therefore, we run `stress --cpu` W `--io` W `--vm` W `--hdd` W, where W is the number of workers, from 1 to 4. Note that the `stress` command is affecting CPU, I/O, memory and hard disk at the same time, in order to mimic how

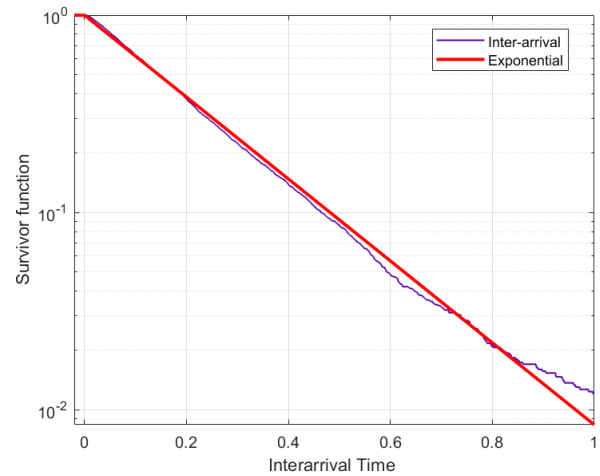

Fig. 8. Connection inter-arrival time fit.

TABLE I
Stress Data

W	Values (%)						
	usr	***sys***	***iowait***	***irq***	***soft***	***idle***	***load***
0	4.25	1.37	0.17	0.00	0.34	97.38	6.13
1	17.04	17.72	21.84	0.00	0.39	48.00	52.00
2	31.9	24.68	23.50	0.00	0.54	19.69	80.31
3	45.26	35.94	14.98	0.00	0.55	3.27	96.73
4	53.89	45.33	0.11	0.00	0.60	0.07	99.93

a system is usually loaded. The averaged load figures in the tests are shown in Table I. Load percentages for *steal*, *nice* and *guest* were null in all cases. The server load evolution during the five tests is shown in Fig. 9. In the case with the highest load, there is low variation, being most of the time 100%.

C. SRT variability with server load

Using the environment described above, we got SRT samples within a 10 min. window for each load. Based on the arrival rate, this sampling process provided about 3000 observations per round. We needed such amount of samples to achieve good confidence bounds when fitting heavy-tailed distributions [19]. Then, we obtained the α-stable parameters (α, β, γ, δ) of the distributions that fitted these data sets. Fig. 10 shows the SRT distributions for one of the tests. As shown in Fig. 9, the stress conditions were stationary during the window span, with relatively small fluctuations. We expected a correlation between the server load and the parameters as the TCP stack, even running in kernel space, will be affected by the `stress` command. In Fig. 11 we can see an example of the change of every parameter when stress conditions increase.

The only parameter of the α-stable distribution that does not seem to change substantially with the load is β, whereas γ and δ tend to increase. α shows a heavy tailed distribution in all iterations with load. This is, SRT distribution is asymmetric with a heavy tail on the right, and higher loads imply a longer

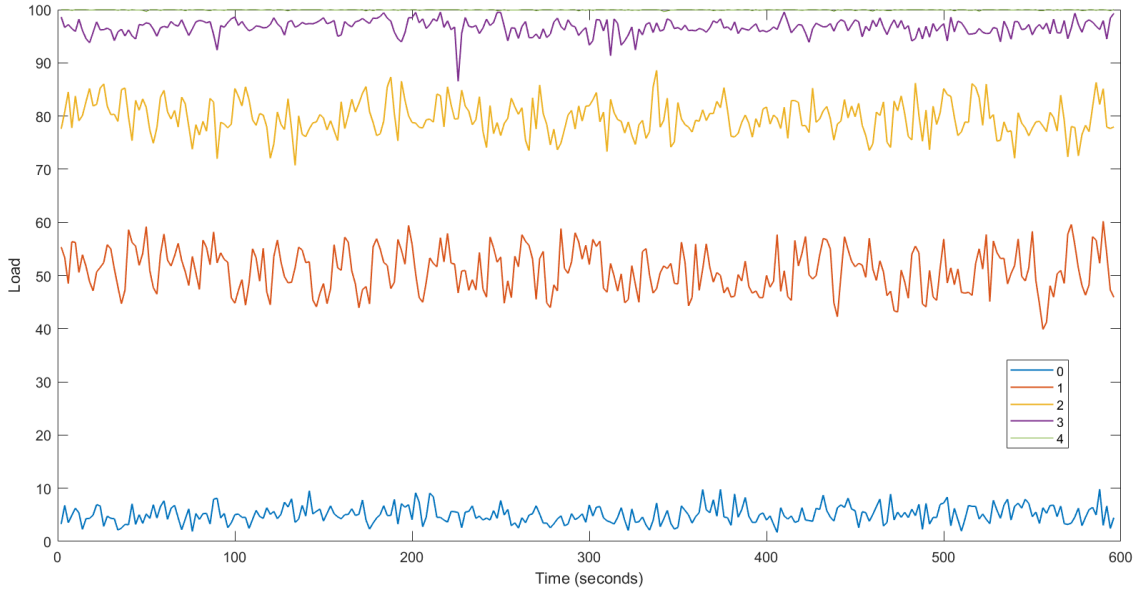

Fig. 9. Load time series evolution dependence with `stress` parameter W.

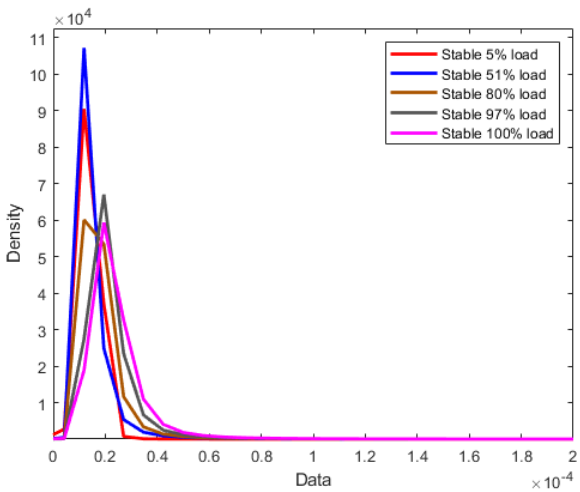

Fig. 10. α-stable fits for SRT data (in s) obtained in one of the tests for several workloads. Tail of the distributions is not shown.

and more variable SRT. These obtained values support the use of α-stable distributions.

Variations in γ and δ can be explained as a result of the load increase. When the server is loaded, TCP stack has less time available to response to SYN segments, and the time slot the scheduler assigns to the stack is then more random. The effect is the location of the distribution (δ) is increased, as it is somehow related to the distribution mean. The less predictable slot time available for the TCP stack may be the reason why the scale (γ) is also increasing. Counter-intuitively enough, we have noticed a characteristic behavior for low loads in several measurements. The δ parameter does not seem to decrease proportionally for very low CPU load, which is an open issue to be addressed.

We have observed some consistent fluctuations in α values, but they are not significant enough to be considered as a load indicator without further research. Our best educated guess is that they may be related to some behavior resembling architecture features and probably this is not easily generalized.

In conclusion, we can rely on γ to estimate the server load.

IV. Server Load Estimation Method

Based on the results provided above, we have defined the following 3-step method to estimate the server load:

- First of all, given the dependence of this estimation on the used hardware and software, it is necessary to obtain the correlation curves of the α-stable parameters with the load for the servers. This training phase would be similar to what is shown in previous section, obtaining as a result a regression curve useful to estimate the server load.
- After the training phase, SRT samples have to be collected at the vantage point. This collection can be done non-intrusively by capturing with a probe the SYN and SYN+ACK segments of real TCP connections arriving to the server.
- Once we have the collected SRT samples, we can fit them to an α-stable distribution and obtain its parameters. Based on the value of γ and the regression curve obtained in the first step, we can finally estimate which is the server load.

There is a trade-off between the accuracy of the estimation and the time it takes to collect the SRT samples. However, usually there are more connection attempts in the most loaded

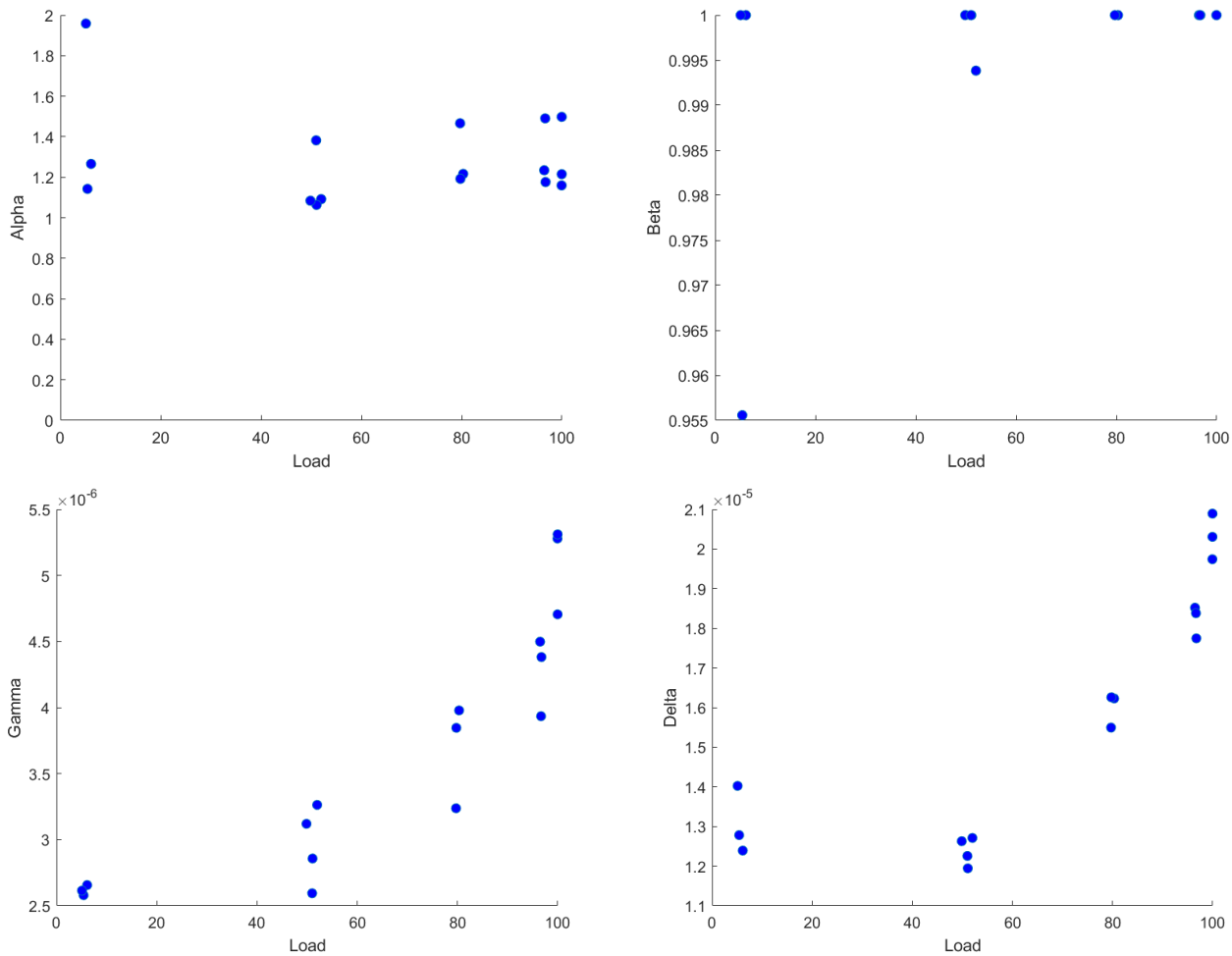

Fig. 11. α-stable parameters vs. server load

times, so we are going to have more samples when we need them more to diagnose the cause of the problems.

If SRT samples are collected passively, it is important to reduce the bias caused by the non-poissonian nature of the TCP connection arrivals [20]. For this, we propose to apply a Bernoulli process that randomly chooses among the measurements with probability p, given that Bernoulli arrivals also see time averages [21]. Another possibility is to issue SYN packets from a testing device and measure the SRT distribution. This second approach is more intrusive, but it may assure that a Poisson distribution is used to get the figures and there are no artifacts in the measured SRT. Finally, it is also advisable a combination of both approaches, to cover high and low load periods, or when the applications work with persistent connections.

V. Conclusions

In this paper, we have presented a novel approach, based on the SRT time, to estimate server load without measuring it internally. This approach is very valuable to identify bottlenecks causing problems in distributed systems, as well as to balance the load in server clusters.

For this, we measured the time from the SYN to the SYN+ACK TCP segments at the server side. We have identified that SRT varies along the day in servers, following their workload. SRT is distributed with a heavy tail, which is well modeled by a α-stable distribution. Finally, we have found that server load is correlated with both γ and δ parameters of the α-stable distribution.

Based on these results, we have defined an estimation method, which follows an initial training phase, where the server load is characterized, and then, a monitoring phase where SRT samples are taken to find the load distribution of the server, based on the obtained α-stable parameters.

As future work, we plan to study in detail how accurate our method is in order to estimate the server load. This analysis can be done with respect to the number of samples or the probability to choose a sample. This is especially important if we use a Bernoulli process to reduce the bias of the estimation. Other idea is to study how this method behaves when the server load is not stationary. Finally, it is also important

to deal with the security implications of these results, as cybercriminals could use this technique to know when the servers are more loaded and attack them at that time.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under the project TRÁFICA (MINECO/FEDER TEC2015-69417-C2-1-R).

REFERENCES

[1] M. Conti, Q. Q. Li, A. Maragno, R. Spolaor, "The dark side (-channel) of mobile devices: A survey on network traffic analysis," *IEEE Communications Surveys & Tutorials*, Vol. 20, no. 4, Jun 2018, pp. 2658-713.

[2] C. So-In, "A survey of network traffic monitoring and analysis tools," Cse 576m computer system analysis project, Washington University in St. Louis, 2009.

[3] A. Adhikari, S.V. Bianco, L. Denby, L., C.L Mallows, J. Meloche, B. Rao, S.M. Sullivan, Y. Vardi, "Distributed monitoring and analysis system for network traffic," U.S. Patent 7,031,264, 2006.

[4] P. Malloy, A. Cohen, R. Gehl, J. Strohm, R. Elsner, "Interactive network monitoring and analysis," U.S. Patent Application 11/639,863, 2007.

[5] J. Liu, C. Zheng, L. Guo, X. Liu, Q. Lu, "Understanding the Network Traffic Constraints for Deep Packet Inspection by Passive Measurement," in Proc. 3rd International Conference on Information Systems Engineering. 2018.

[6] S. Taylor, J. Metzler, "Eliminating the mean time to innocence," *Network World*, Mar 10, 2009.

[7] R. Alshahrani, H. Peyravi, "Cluster Load Estimation for Stateless Schedulers," in Proc. IEEE 17th International Symposium on Network Computing and Applications (NCA), November 2018.

[8] Y. M. Teo, R. Ayani, "Comparison of Load Balancing Strategies on Cluster-based Web Servers," *SIMULATION*, Vol. 77, No. 5-6, pp. 185-195, 2001.

[9] K. Shen, T. Yang, L. Chu, "Cluster load balancing for fine-grain network services," in Proc. 16th International Parallel and Distributed Processing Symposium, Ft. Lauderdale, FL, 2002.

[10] C. Kopparapu, *Load Balancing Servers, Firewalls, and Caches*, chap. 2, Wiley, 2002

[11] T. Høiland-Jørgensen, B. Ahlgren, P. Hurtig, A. Brunstrom, "Measuring Latency Variation in the Internet," in Proc. 12th International on Conference on emerging Networking EXperiments and Technologies (CoNEXT '16). ACM, New York, NY, USA, pp. 473-480. 2016.

[12] J. Aikat, J. Kaur, F. D. Smith, and K. Jeffay, "Variability in TCP round-trip times," in Proc. 3rd ACM SIGCOMM conference on Internet measurement. ACM, 2003.

[13] D. Perdices, D. Muelas, I. Prieto, L. de Pedro, J. E. López de Vergara, "On the Modeling of Multi-Point RTT Passive Measurements for Network Delay Monitoring," *IEEE Trans. Network and Service Management*, Vol. 16, No. 3, pp. 1157-1169, September 2019.

[14] F. Simmross-Wattenberg, J. I. Asensio-Pérez, P. Casaseca-de-la-Higuera, M. Martín-Fernández, I. A. Dimitriadis and C. Alberola-López, "Anomaly Detection in Network Traffic Based on Statistical Inference and α-Stable Modeling," *IEEE Trans. Dependable and Secure Computing*, Vol. 8, No. 4, pp. 494-509, jul-ago. 2011.

[15] J. P. Nolan, *Stable Distributions - Models for Heavy Tailed Data*, chap 1. Birkhauser, 2018.

[16] F.J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American Statistical Association*, Vol. 46, No. 253, pp. 68-78, 1951.

[17] M.A. Stephens, "EDF Statistics for Goodness of Fit and Some Comparisons," *Journal of the American Statistical Association*, Vol. 69, No. 347, pp. 730-737, 1974.

[18] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "RFC 2330: Framework for IP performance metrics," IETF Request For Comments 2330, 1998.

[19] J.P. Nolan, "Fitting Data and Assessing Goodness-of-Fit with Stable Distributions," in Proc. Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics, Washington DC, 3 Jun 1999.

[20] S. Hagihara, Y. Fushihara, M. Shimakawa, M. Tomoishi, N. Yonezaki, "Web Server Access Trend Analysis Based on the Poisson Distribution," Proc. 6th International Conference on Software and Computer Applications. New York, 2017.

[21] A. Makowski, B. Melamed, W. Whitt, "On averages seen by arrivals in discrete time," in Proc. 28th IEEE Conference on Decision and Control, Tampa, FL, USA, 1989, pp. 1084-1086 vol.2.