Distributed Troubleshooting of Web Sessions Using Clustering

Heng Cui, Ernst Biersack

EURECOM, Sophia Antipolis, France
firstname.lastname@eurecom.fr

Abstract. Web browsing is a very common way of using the Internet to, among others, read news, do on-line shopping, or search for user generated content such as YouTube or Dailymotion. Traditional evaluations of web surfing focus on objectively measured Quality of Service (QoS) metrics such as loss rate or round-trip times; In this paper, we propose to use K-means clustering to share knowledge about the performance of the same web page experienced by different clients. Such technique allows to discover and explain the performance differences among users and identify the root causes for poor performances.

Keywords: Web Browsing, Home Networks Measurement, Troubleshooting

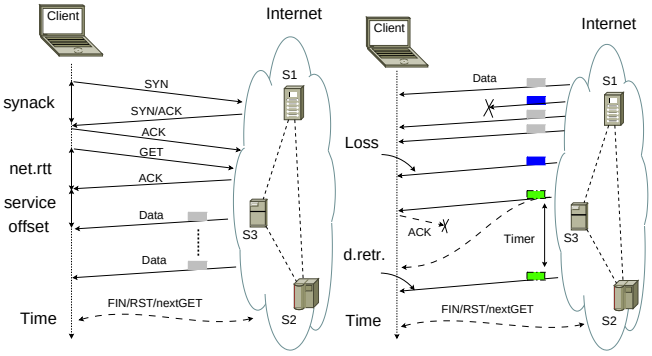
1 Introduction

Web browsing is a very common way of using the Internet access as it allows to access to a wealth of information. Since there is a "human in the loop", the time it takes to render a Web page should be small in order to assure a good user experience.

Traditional evaluations of the web surfing mainly use Quality of Service (QoS) metrics that are easy to measure such as packet loss rate or round trip times. In this paper, we propose a methodology which combines a browser plugin and lower level capturing to evaluate web page performances, and we use clustering to share the performance metrics. We demonstrate that this clustering is suitable to compare and explain the experiences among different clients and further identify root causes for poor performance.

This work builds on our previous work [1] where we presented the measurement architecture but did not carry out any systematic analysis of the measurements. An extended version of this work is available in our technical report [2].

Different methodologies of troubleshooting user's network connections are proposed in the literature. Joumblatt et al. [4] propose HostView, an end-system to collect user generated packets and store all the information into a centralized server. Netalyzr [5] is a web-based diagnostic tool that to analyze and debug end-user's connectivity properties such as bandwidth, proxy, etc. Compared to these works, our work focuses only on Web browsing.



(a) Delay Related Metrics (b) Loss Related Metrics

Fig. 1. Metrics Related to Download One Element In a Web Page

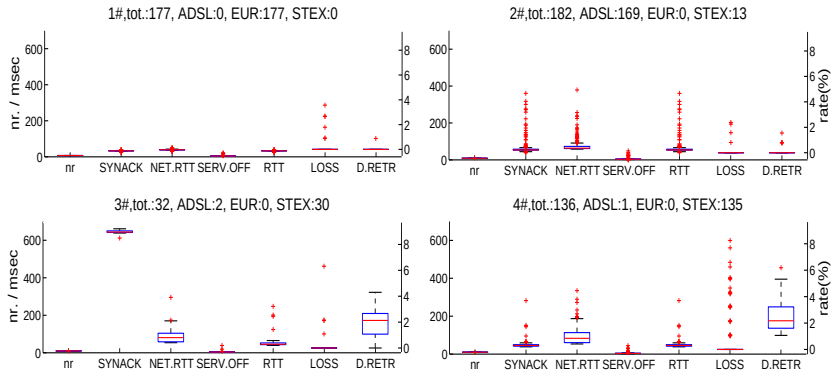
2 Methodology

We set up an experimental platform that uses a firefox plugin and normal packet capture to measure both, page level information such as full page load time, and lower packet level metrics such as RTT or loss rate. The firefox plugin, among others, binds each HTTP query initiated by the browser to its associated firefox window/tab and measures the full load time for that Web page. We use packet capture (libpcap format) to obtain raw packet traces that are loaded into a database for post-processing. For details about the architecture and how we combine the measured records, we refer the readers to our previous work [1].

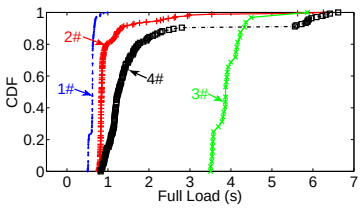
A typical web page can contain up to hundreds of elements. To fully render the Web page, the browser needs to load all these elements. The typical procedure for loading one element is shown in Fig.1. As is shown in Fig.1(a), for download of each element, we extract metrics as shown in Fig.1 from the packet trace. For detailed description of our defined metrics, please refer to our extended report [2]. To describe the performance of a Web *session*, we use the metrics computed for each Web *element* and compute the **mean** over all the values of the given metric to obtain a **Key Performance Index(KPI)**, which consists of

$$[nr., SYNACK, NET.RTT, SERV.OFF., RTT, LOSS, D.RETR.]$$

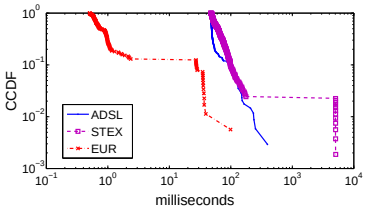
Note that *nr.* is the number of distinct GET requests during a complete web session, which provides an estimation of the size of the Web page in terms of the number of elements. Meanwhile, for the KPI, we focus on metrics captured by TCP connections, and currently we ignore DNS queries since TCP connections already provide rich enough information; and DNS pre-fetching is widely supported by recent browsers and this causes DNS queries to occur before a real web page surfing, which makes the lookup time less useful. However, we plan to study the effects of DNS (e.g. response time, response IP, TTL, etc.) on the web browsing experiences in the future. To compare results among different homes, we use clustering: we first normalize all the measured KPI metrics



(a) 'Google' Session Clustering Signatures



(b) 'Google' Page Load Time



(c) 'Google' DNS Response Time

Fig. 2. 'Google' Sessions in Three Homes

into the range [0,1], and then use the well known *kmeans* algorithm, which is an un-supervised classification algorithm that does not need any training. The tricky point in *kmeans* is how to set a-priori the number of clusters. We use *four clusters throughout the paper,* and refer for a more detailed discussion to our technical report [2]. Furthermore, we run the *kmeans* ten times and keep the one with smallest distance error.

3 Troubleshooting Case Study

In this section, we show a simple example of how to use clustering to troubleshoot web sessions. The main idea behind this is to share browsing knowledge among different clients located in different homes. We emulate users browsing (i) at home connected via an ADSL connection, (ii) in the office at Eurecom connected via a 100 Mb/s link to the Internet and (iii) in a student residence. All the machines are located in France. We refer to the ADSL home, Eurecom office Ethernet and the student residence connections as **'ADSL'**, **'EUR'**, and **'STEX'** respectively. The experiments are done during the same evening in three homes and last for around 8 hours each. Both, the 'EUR' and 'STEX' client computers are connected via a wired connection, while the 'ADSL' client computer

is physically very close to the Access Point and uses a wireless connection. We clear the browser cache at the end of each Web session.

As an illustration of how the comparison of the performance of the access to *the same Web page across different clients* helps identify the influence of problems specific to a client, we use the `Google` Web page, and show the results in Fig.2. We know that Google works very hard to keep the page download times as low as possible by placing servers close to the clients and also by keeping the number of elements of its Web page low. We see that the requests from 'EUR' are grouped in cluster 1 and from 'ADSL' are grouped in cluster 2. On the other hand, the requests in clusters 3 and 4 are issued almost exclusively from the 'STEX' client. The fact that the 'STEX' client experiences higher delays and also loss can be seen in its KPIs. Cluster 3 is interesting because of its large SYNACK values; it turns out that for cluster 3 on average one out of five SYN requests to establish a TCP connection does not get answered and must be retransmitted after timeout. Since the retransmission timeout for the SYN packet is three seconds, we get SYNACK $= \frac{(50+50+50+50+3050)}{5} = 650(ms)$. The long tail for the page load times in cluster 4 is due to the long DNS response time for some of the sessions. Since the 'STEX' client experiences more packet loss than the other two clients. We can clearly identify in Fig.2(c) that around 3% of the DNS queries need to be retransmitted.

4 Future Work

As future work, we plan to deploy our system on more end-users and also check how the number of clusters for kmeans is affected by the number of users. We also plan to extend our system to work in real time and in a distributed fashion by making the agents in the different locations communicate and perform distributed clustering [3].

References

1. H. Cui and E. Biersack. Trouble Shooting Interactive Web Sessions in a Home Environment. In *ACM SIGCOMM Workshop on Home Networks*, Toronto, Canada, August 2011.
2. H. Cui and E. Biersack. On the Relationship Between QoS and QoE for Web Sessions. Technical Report RR-12-263, EURECOM, Sophia Antipolis, France, January 2012. `http://www.eurecom.fr/~cui/techrep/TechRep12263.pdf`.
3. S. Datta, C. Giannella, and H. Kargupta. Approximate Distributed K-Means Clustering over a Peer-to-Peer Network. *IEEE Transactions on Knowledge and Data Engineering*, 21:1372–1388, October 2009.
4. D. Joumblatt, R. Teixeira, J. Chandrashekar, and N. Taft. HostView: Annotating End-host Performance Measurements with User Feedback. In *HotMetrics, ACM Sigmetrics Workshop*, New York, NY, USA, June 2010.
5. C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating The Edge Network. In *IMC '10: Proceedings of the 10th annual conference on Internet measurement*, pages 246–259, New York, NY, USA, 2010. ACM.