On the nature of unused TCP connections in web traffic

Luis Miguel Torres, Eduardo Magaña, Mikel Izal, Daniel Morató
[luismiguel.torres,eduardo.magana,mikel.izal,daniel.morato]@unavarra.es
Dpto. de Automática y Computación, Universidad de Pública de Navarra. Campus de Arrosadía, Pamplona, España.

***Abstract*— The popularity of the web and the requirements introduced by current web content have pushed for the development of new techniques that meet these challenges and improve the experience of the users. In particular, during the last years, web browsers have taken aggressive measures in order to reduce webpage download times. These measures have had a noteworthy effect on the profile of web traffic. One of the most striking consequences is that nowadays, more than 20% of the TCP connections opened by a browser are left unused. In this paper we describe these connections, explain why they happen and use them as a simple way of identifying the traffic of different web browsers.**

I. Introduction

The web is probably the classic Internet application that has grown and evolved the most during the past two decades. The simple and mostly static webpages of the 1990s have given way to much more complex sites. This complexity is represented, in the first place, by the addition of a wide variety of content types (such as videos or interactive media) to the text and images that classic webpages traditionally hosted. Nevertheless, modern websites not only offer these new content types, but they do so in a dynamic way, keeping their content current and tailoring their offer to each specific visitor.

The network requirements introduced by all this and the ever-increasing popularity of the web have also pushed for updates in the web application protocols; the development of new techniques that help in web operation, such as content distribution networks (CDNs) or analytics services; and the introduction of new features in web browsers with the objective of improving user experience.

However, this evolution has been somewhat uneven. Today, web communications are still governed by the HTTP/1.1 protocol [1] which, despite having been updated through the years, dates from 1999 and was designed for a very different web. This has forced web service providers and web browser designers to react to the new challenges introduced by the evolution of web content as they have appeared. Some of the proposed solutions have become *de facto* standards simply by being used in a majority of web clients or servers.

As a consequence, many aspects of web operation depend on the particular implementation of the website accessed, the servers it is hosted in and the client used to browse trough its webpages. This, combined with the complexity of modern web content, produces very variable traffic profiles that are difficult to characterize and model and which have sparked the interest of the scientific community. Previous work has been done to study the changes on web traffic from a server [2], client [3] and network [4] perspective both using traffic traces [5] and information captured at the application level [6].

In our case, in [7] we studied the sets of connections established by clients during the download of individual webpages. We took a connection-level perspective rather than focusing on application data or studying packet traces. In order to carry out our study, we automatically accessed a set of a thousand popular websites and captured the connections generated during the download of their landing pages. Among other findings, we discovered that a sizable amount (more than 20%) of TCP connections to port 80 were left unused. That is, the connections were properly established between the client and a server but finished without any exchange of application data.

In this paper we seek to characterize the appearance of these unused TCP connections in real web traffic and provide explanations about why they happen. The rest of this paper is organized as follows. Section II presents our experimental data set. Section III describes the case of unused connections in web traffic and how they have a sizable presence through different clients and servers. Section IV explains why web browsers open connections they do not use. Section V presents the characteristics of unused connections as a possible signature able to identify different browsers. Finally, section VI concludes.

II. Data collection

For the experimental measurements presented in this paper, we have used traffic captured at the Internet link of our university network. Although the network of the Public University of Navarre (UPNA) serves a community of close to 10.000 people, most of these users (those in computer labs and those using the university's WiFi network) connect to the Internet through NAT routers, which complicates capturing their individual traffic. Because of this, we have only considered traffic of users with public IP addresses.

We have captured a traffic trace spanning three work days from April the 14th to April the 16th, 2015. More than 200 unique users with public IP addresses were actively browsing the web during said time interval. Rather than capturing their traffic in pcap format we use flow records as described by the IPFIX standard [8] which is inspired by Cisco's NetFlow. In order to obtain these records we use an auditing tool called

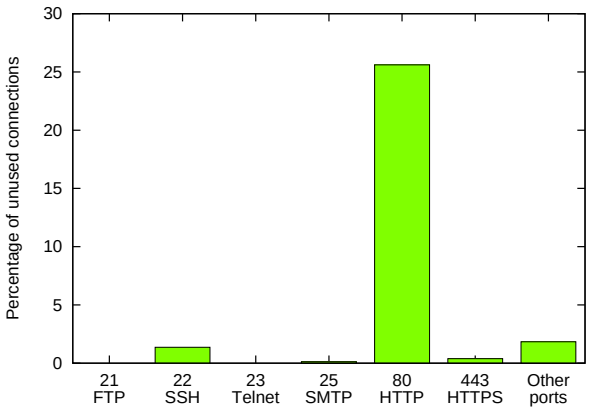

Fig. 1. Percentage of unused connections for different ports

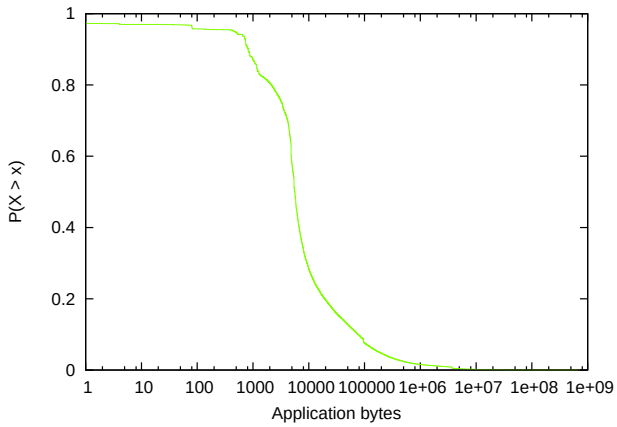

Fig. 2. CCDF of HTTPS connection sizes

Argus [9]. Flow records have the advantage of providing summarized information that is easier to store and process and are less invasive of the privacy of users (in our case, the only application-level data we have extracted from the records is the user-agent field in HTTP headers).

In addition to the classical 5-tuple that identifies a flow (transport protocol, client and server IP addresses and ports), we store, for each of them: the timestamps of their first and last packet; the final TCP state; the total number of bytes and packets; the total number of upstream and downstream application-level bytes; and the first 1000 bytes of upstream application data (from which, as we said before, we only extract the user agents).

We will use data from the flow records described in this section in the remainder of the paper.

III. Unused connections in modern web traffic

We define an *unused connection* as a TCP flow that has been correctly established (by the TCP three-way handshake) and terminated (by the FIN handshake or RST messages) but in which no application data has been exchanged. In order to show how these connections are especially prevalent in web traffic, in Fig. 1 we present the percentage of unused connections for different server ports. As we can see, more than 25% percent of connections to port 80 (HTTP) are unused. This percentage is much lower for any other server port suggesting that these unused connections are closely related to web operation.

Initially, the low percentage of unused connections in HTTPS seemed strange to us given the increasing use of encrypted connections in modern websites. However, it must be taken into account that HTTPS connections may be unused even if application data is exchanged: said application data may correspond only to TLS/SSL overhead. The initial TLS handshake in an HTTPS connection can introduce multiple Kilobytes of overhead depending on the size and number of the certificates exchanged. In Fig. 2 we show the complementary cumulative distribution function (CCDF) of the application bytes in HTTPS connections. As we can see, around 20% of them exchange less than 2000 bytes suggesting that they only include a TLS handshake and no real user data (this would be a similar proportion as the one we have seen for HTTP). However, from our connection-level perspective and given the little amount of application data we capture, we have no reliable way of distinguishing between used and unused HTTPS connections. Therefore, in the remainder of this paper we will focus solely on unused HTTP connections and assume that the conclusions we reach about them are extensible to HTTPS.

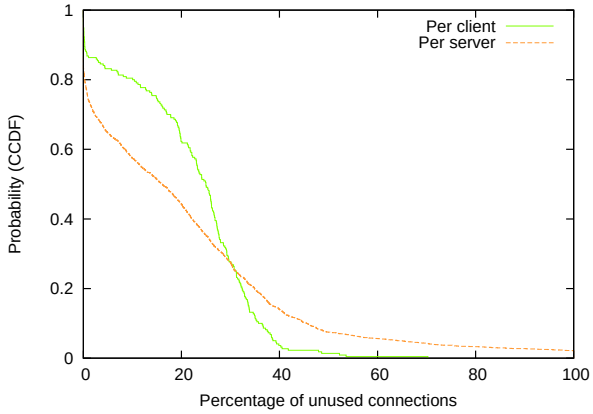

Fig. 3. CCDFs of the percentage of unused connections

Although we have seen that there is a high number of unused HTTP connections, it is also interesting to study if the percentage of unused connections depends on the web client and on the servers he accesses. Fig. 3 addresses that question by representing the CCDFs of the percentage of unused connections for each client in our network and for each of the external servers accessed by them during our three-day traffic trace. In both cases we can see that unused connections occur for a vast majority of clients and servers. In particular, more than 10% of connections are unused for more than 80% of clients (clients without unused connections may be running older web browsers). In the case of servers, the variability is

greater with around 20% not receiving unused connections and almost 5% receiving only unused connections. We will provide possible explanations for this behavior in the following section.

IV. Web browsers and unused connections

Most modern web browsers implement two well known mechanisms directed to reducing webpage load times that can result in unused connections: HTTP parallel connections and a short timeout for TCP connection establishment.

With the introduction of persistent connections in HTTP/1.1, it became possible for web clients to use the same TCP connection for requesting more than one resource to the same server. This eliminates the overhead and latency added by establishing additional connections. However, when downloading multiple resources from the same server it is often convenient to use multiple parallel connections in order to expedite the download. At the time of the introduction of HTTP/1.1, two parallel connections were recommended in these cases. However, nowadays, most browsers open up to six concurrent connections to download content from the same server. Browsers try to predict how many connections will be optimal to open to each server by rapidly scanning the webpages and even using knowledge gathered in previous visits. In spite of this, sometimes their predictions are mistaken and some of the opened connections are left unused.

On the other hand, most implementations of TCP have a SYN retransmit timer of 3 seconds. This means that when the client attempts to establish a connection with a server by sending a SYN packet, it will wait during 3 seconds for a SYN-ACK packet. If the answer does not come in that interval, the client will assume it was lost and it will retry with a second SYN packet. However, for an application such as the web in which the user is waiting for a webpage to load, this 3 second interval seems to long and modern browsers have tried to circumvent it. Most browsers attempt to open a new connection if the server does not answer to the first attempt before a much shorter timeout expires (250ms). Nevertheless, the server may still answer to the first SYN packet after that and, in many cases, two connections are opened. Now, depending on the content that was to be downloaded from that particular server, one of them may not be used.

It seems clear that, for both mechanisms, the unused connections will be opened close in time to used connections to the same servers. In order to see this, we have calculated the time distances between the start timestamps of each unused connection and its closer used connection to the same server (obviously both originating from the same client in our network). We represent the CCDF of these distances as the green continuous line in Fig. 4 (we will come back to the other line in the following paragraphs). Studying the distribution, we realize that more than half of the unused connections have a used connection very near (P50 = 0.052s). These connections seem to be unused parallel connections that were opened almost at the same time as their used counterparts.

Unused connections caused by the short timeout for connection establishment should be opened almost exactly 250ms

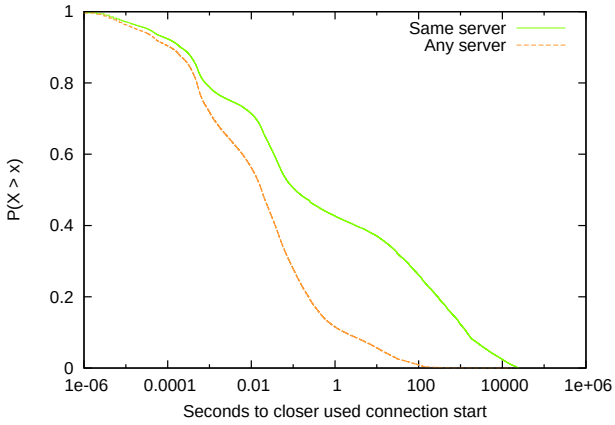

Fig. 4. CCDFs of the distance between used and unused connections

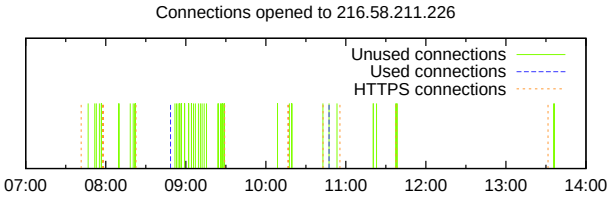

Fig. 5. Used and unused connections in the traffic between a client and a server

away from the closest used connection. Although while analyzing our results we have found some of these cases, their number is not significative enough to affect the shape of the distribution.

In any case, we see that the distribution has a very wide right tail and the two mechanisms we have explained can hardly justify the presence of unused connections that happen seconds, minutes or even hours away from the closest used connection to the same server. In fact, as an example, in Fig. 5 we show the start timestamps of the connections opened during a morning by a client in our network to 216.58.211.226 (a Google server). Green impulses represent unused connections, blue impulses, used ones and orange impulses, HTTPS connections. Even taking HTTPS connections into account it does not seem logical that the browser keeps opening unused connections to the Google server in time intervals where there is no exchange of application data at all.

The possible reason behind this behavior is a third mechanism, more recent than the other two, and which many modern browsers implement: TCP preconnect. With TCP preconnect, browsers attempt to further reduce latency by opening TCP connections before they are needed. In order to achieve this they use multiple predictive features that rely on stored data about previously visited websites and about known user behavior. They also take into account current user actions by, for example, preparing connections to Facebook servers if a user known to visit Facebook starts to type its URL in the navigation bar. In brief, TCP preconnect involves a number of complex techniques and its implementation is quite different depending on the web browser [10].

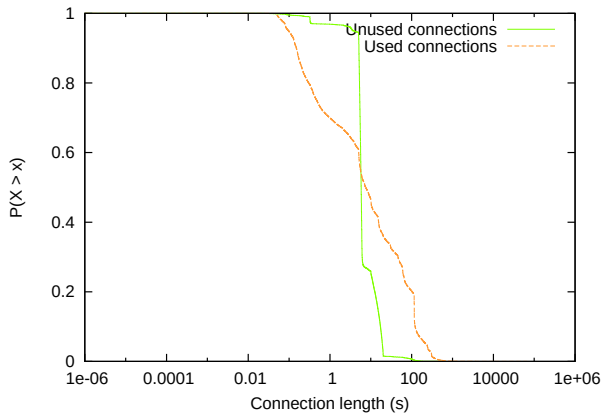

Fig. 6. CCDFs of the length of used and unused connections

Of course, the predictive behavior of TCP preconnect is prone to mistakes and it makes sense that many unused connections are caused by it. If we consider TCP preconnect, unused connections do not necessarily need to be opened close to used connections to the same servers because the browser may have failed in predicting the webpage the user was going to visit. In that case the unused connection should be very near to used connections to other servers. To show this behavior, the orange line in Fig. 4 represents the distribution of the distances between unused connections and used connections to other servers. We see that now, more than 80% of the unused connections are closer than 1 second to a used one.

In fact, TCP preconnect could explain even the unused connections that happen far from used connections to any server. A user that starts writing something in the address bar of his browser may be distracted before deciding to visit a webpage while the browser has already pre-opened some connections. Moreover, some browsers gather information about the first websites a user usually visits and preconnect to their servers as soon as they are launched.

In brief, our results show that unused connections are primarily caused by TCP parallel connections and TCP preconnect with each of these mechanisms been responsible for around half of the unused connections opened.

V. Using unused connections to identify browsers

We have seen that most unused connections in web traffic are a result of techniques for reducing webpage download times. However, they lead to a bigger resource consumption in clients, servers and certain network elements (for example, NAPT routers). This can be specially worrisome if the connections are kept open for a long time. In Fig. 6 we represent the CCDFs of the length of used and unused connections. Although unused connections are longer than many used ones, their length is, at least, limited and the right tail of the distribution is very narrow. In fact, we can see two sharp steps in the distribution that correspond to the default timeouts for these connections in the most used web browsers in our network (5s for Firefox and 15s for Google Chrome).

Fig. 6 inspired us to use unused connections as a simple way of identifying web browsers from a connection perspective. More than any other aspect of web traffic, unused connections depend on the browser's implementation rather than on the webpages visited and the behavior of the user. To test this idea we extracted the user agent from the HTTP connections of the hosts in our network. We selected 100 hosts that showed the same user agent in, at least, 80% of their connections (64 Firefox, 27 Chrome and 9 Safari). For these hosts we calculated three simple metrics: percentage of unused connections, median unused connection length and number of unused connections per different server. We fed these metrics to a Naive Bayes classifier and trained it with half of the hosts. The trained classifier was able to correctly identify the user agents of the remaining 50 hosts without any mistake.

Although this identification method will require further validation it is promising in that it allows to identify the user agent without monitoring application-level data using very simple metrics. Furthermore, the fact that it works supports the conclusion that unused connections are heavily dependent on browser implementation.

VI. Conclusions

When monitoring modern web traffic, the volume of unused connections may seem alarming. In this paper we have provided an explanation for this phenomenon explaining which new features of web browsers are responsible for it. We have also discovered that, unused connections are heavily dependent on web browser implementation and that it may be possible to use them to calculate simple connection-level metrics able to distinguish between different browsers.

References

[1] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," 1997. [Online]. Available: https://www.ietf.org/rfc/rfc2068.txt

[2] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao, "Characterization of a large web site population with implications for content delivery," *World Wide Web*, vol. 9, no. 4, pp. 505–536, 2006.

[3] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Understanding website complexity: measurements, metrics, and implications," in *ACM conference on Internet Measurement*, ser. IMC '11, 2011, pp. 313–328.

[4] F. Schneider, B. Ager, G. Maier, A. Feldmann, and S. Uhlig, "Pitfalls in HTTP traffic measurements and analysis," in *International Conference on Passive and Active Measurement*, ser. PAM'12. Springer, 2012, pp. 242–251.

[5] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network: measurements, models, and implications," *Computer Networks*, vol. 53, no. 4, pp. 501–514, 2009.

[6] S. Ihm and V. S. Pai, "Towards understanding modern web traffic," in *ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '11, 2011, pp. 295–312.

[7] L. M. Torres, E. Magana, M. Izal, and D. Morato, "Characterizing webpage load from the perspective of TCP connections," in *Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2014, pp. 977–984.

[8] J. Quittek, T. Zseby, B. Claise, and S. Zander, "Requirements for IP flow information export (IPFIX)," 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3917.txt

[9] ARGUS: Audit Record Generation and Usage System. [Online]. Available: http://www.qosient.com/argus/

[10] S. Sanders and J. Kaur, "On the variation in web page download traffic across different client types," in *International Conference on Network Protocols*, Oct 2014, pp. 495–497.