Abstract—The fast growth of Internet traffic, the growing importance of cellular accesses and the escalating competition between content providers and network operators result in a growing interest in improving network performance and user experience. In terms of network transport, different solutions ranging from tuning TCP to installing middleboxes are applied. It turns out, however, that the practical results sometimes are disappointing and we believe that poor testing is one of the reasons for this. Indeed, many cases in the literature limit testing to the simple and rare use case of a single file download, while common and complex use cases like web browsing often are ignored or modelled only by considering smaller files. To facilitate better testing, we present a set of metrics by which the complexity around web pages can be characterised and the potential for different optimisations can be estimated. We also derive numerical values of these metrics for a small set of popular web pages and study similarities and differences between pages with the same kind of content (newspapers, e-commerce and video) and between pages designed for the same platform (computer and smartphone).

Index Terms—Performance enhancing proxy, web metrics, traffic model, flow concurrency, congestion control

I. INTRODUCTION

The increasing importance of internet and mobile web applications in modern life, together with increasing difficulties for internet service providers and mobile network operators to comply with the high demands imposed on them to meet end-user expectations with steadily decreasing time to deliver content in a PEP, essentially handles a large number of flows with single objects as a smaller number of flows with multiple objects, i.e., we obtain a transmission scheme with object multiplexing as in SPDY and QUIC. Both concepts rely on the timing of flows hence traffic models with these aspects included are needed. We remark that these models should include all traffic when applied by network operators in PEPs, but that we need to separate traffic by domain or server when applied by content providers.

Early work on web traffic metrics and models primarily focused on static properties such as the distribution of page and image sizes and URL counts. Bray [8] is an example of such an early work. He makes an attempt to measure said static properties; other examples include Arlitt and Williamson [9], Mah [10], and Cunha [11]. As the web has evolved, later work, in addition to static properties, also consider temporal properties, e.g., Barford and Crovella [12], Lee and Gupta [13], and Smith et al. [14]. More recent work also take into account the dynamic properties inherent with Web 2.0 technologies such as AJAX, CSS and JavaScript frameworks: Ihm and Pai [15] and Pries et al. [16] analyse how web traffic has evolved from static to more dynamic contents, and its implications for those web traffic models being used. Related to this work, Butkiewicz et al. [17] propose metrics more appropriate for Web 2.0 traffic. Another trend in recent work is an increasing interest in the characteristics of mobile web traffic, e.g., Johnson, Seeling and Knox [18], [19] study the characteristics of this traffic and how it differentiates from computer web traffic.

The focus of this work is PEPs which, besides reducing round trip times by splitting direct flows (server–proxy or proxy–client, respectively), can be used as a platform for deploying CCAs which are tailor-made for the current network. Examples of such PEPs include the TRL-PEP proposed by Ivanovich et al. [20], and the more recently proposed mobile network accelerator put forth by Liu and Lee [21] which enables on-the-fly techniques to enhance congestion control in mobile networks. Moreover, recently, Islam et al. [7], suggested a coupled congestion controller, ctrlTCP, that builds on previous works on ensemble sharing and coupled congestion control such as Eggert et al. [4] and Savoric et al. [5], and which is very well suited for inclusion in a PEP.

The gains from such solutions depend not only on the properties of single flows but also on their timing relative to each other. Although there are some work that have studied...
methodologies for measuring flow concurrency, e.g. Trammell and Schatzmann [22], there seem to be few or none that actually study the metrics of flow concurrency. The major contribution of this paper is that it proposes metrics that take into account the inherent transfer concurrency in web traffic at different levels of granularity. The proposed metrics differentiate between flows that overlap in time but without interfering with each other, coinciding flows, and flows that both overlap and interfere with each other, contending flows. Moreover, the proposed metrics recognize that transfer concurrency in web transfers exists at several levels of granularity, ranging from page level all the way down to byte level.

To demonstrate the usefulness of the proposed metrics, a small experiment campaign was conducted and the results suggest some web traffic characteristics that, as far as we know, have not yet been revealed. For example, that the number of coinciding flows often is much larger than the number of competing flows, thus the overlap of flows does not seem to imply flow contention; the significantly larger number of concurrent web transfers generated by the computer version of a web site as compared to its smartphone version; and, the unbalance that seems to exist between the number of upstream and downstream flows generated by smartphone versions of web sites but not by their computer counterparts.

The remainder of the paper is organized as follows. Sections II and III describe and motivate our proposed metrics in terms of content characteristics and transfer concurrency, respectively. To showcase the applicability of these metrics, we conducted a series of experiments, the results of which are presented and discussed in Section IV. Finally, Section V concludes the paper and outlines future work.

II. Content Characteristics

The definition of “content” partly depends on the view and we consider five hierarchically arranged views from page to byte:

- **Page** includes all content sent to a user as a result of a request for a web page. We characterize pages in terms of domains, addresses, flows, packets and bytes.
- **Domain** splits the content by DNS domain names, and includes all content sent to a user from a particular DNS domain name as a result of a request for a web page. We characterize domains in terms of addresses, flows, packets and bytes.
- **Address** splits the content by IP addresses, and includes all content sent to a user from a particular IP address as a result of a request for a web page. We characterize addresses in terms of flows, packets and bytes.
- **Flow** splits the content by TCP flows, and includes all content sent to a user over a particular TCP flow as a result of a request for a web page. We characterize flows in terms of packets and bytes.
- **Packet** splits the content by packets and includes the content sent to a user in a packet as a result of a request for a web page. We keep separate records for each direction and characterize packets in terms of bytes.

### Byte
We keep separate records for each direction. For each view, we count the number of lower view components and derive their cumulated probability distributions. That is, for pages we count domains–bytes, for domains we count addresses–bytes etc. Comparing to the HTTP archive [23], it is noted that the latter mainly adopts a page view hence some but not all of the above metrics are present.

We remark that we do not consider “objects” since they are not a part of TCP but of HTTP (and would be difficult or impossible to characterize from packet trace files in the presence of encryption). It is also noted that the number of addresses and flows depend not only on the content but also on policies with respect to, e.g. domain sharding, load sharing and parallel flows, and that the number of packets may be impacted by network fragmentation. Our aim is thus to be able to characterize the complexity of actual content (cf. the single download) while our numeric results merely constitute some examples.

III. Concurrency Characteristics

Congestion control in TCP is typically constructed to fulfill two goals, viz. (1) to maximise throughput rates and (2) to share congested resources. As seen above, web pages consist of many domains, IP addresses, flows and packets, and the result is some kind of competition for transmission capacity and buffer space.

We characterize the competition seen in our experiments in terms of coinciding flows and contending flows. Coinciding flows are flows which, at least partly, are in progress at the same time but which do not necessarily have packets in transit at the same time. Concurrent flows may thus compete for resources in stateful middleboxes like PEPs but they need not compete for transmission resources. Contending flows, on the other hand, are flows which indeed have packets in transit at the same time and which thus, at least to some degree, compete for transmission resources.

We also view concurrency from two points of view. The page view of flow concurrency means that we consider concurrency with respect to all flows of a page, while the domain view only considers concurrency with respect to flows with the same domain. The former view relates to concurrency at PEPs (which handle all flows of a page) while the latter view relates to concurrency at servers (which handle, or could handle, all flows from a content provider). We also note that exploiting the latter may impact the former.

**Coinciding flows** with respect to a tagged flow \( f \) are other flows in progress at the same time as \( f \), and we consider two metrics, total overlap and average overlap. Fig. 1 depicts four flows F1–F4 and we can see that F1 overlaps in time with no other flow while flows F2–F4 to some extent all overlap, with each other.

We define the duration of a flow as the time between the first and the last packet (in either direction) and the total overlap of a flow \( f \), \( O_t(f) \), as the total number of flows with which \( f \) overlaps and the average overlap of a flow \( f \), \( O_a(f) \), as the time averaged number of flows with which \( f \) overlaps. In
Fig. 1 we thus have that, e.g. \( O_t(F1) = O_a(F1) = 0 \) and \( O_t(F2) = 2 \) (flows F3 and F4 overlap) while \( O_a(F2) = 1.4 \) (one overlapping flow for 60% of the time and two overlapping flows for 40% of the time).

For both \( O_t(f) \) and \( O_a(f) \) we compute the mean and the coefficient of variation (CoV) as well as the most extreme values, and their Pearson product-moment correlation coefficients (CoC) with respect to the number of packets and bytes of \( f \), in both directions and in total, as well as to the arrival and departure times of \( f \) in absolute and relative terms.

Finally we remark that all metrics of concurrency above also depend on the environment: server response times, network delays and drops, browser execution times etc. Similar to above, our aim is thus to be able to characterise the complexity of actual concurrency (cf. the single download) while our numeric results merely constitute some examples. Again comparing to the HTTP archive [23], it is noted that concurrency aspects are not treated at all in the latter.

IV. APPLICATIONS

To demonstrate the usefulness of our proposed metrics, we selected three different content categories of home pages, viz. newspapers, e-commerce and video, and for each category we selected the starting page of the three most popular sites in Sweden according to Alexa [24]. The sites whose starting pages were selected are listed in Tab. II.

The experiment setup is depicted in Figure 3 and consisted of a home network with a 100 Mbps symmetric fibre connection to the Internet, a WLAN router D-Link DIR-655, a Dell Latitude E6220 laptop with a Gigabit Ethernet interface and a wireless adaptor running MS Windows 7 Professional and “My WiFi router” 3.0, and a smartphone LG Nexus 5X with Android 7.1.2.

The laptop was connected to the fibre terminal via a router using Ethernet cables, while the smartphone was connected via WiFi to the laptop and then via the same router and Ethernet cables to the fibre terminal. All traffic thus passed through the laptop, hence all traffic traces could be collected by running Wireshark on the laptop.

To ensure that our packet traces contained complete pages, we flushed the content caches prior to each request. Moreover, to ensure that our packet traces were not contaminated by irrelevant background traffic several measures were taken.
First, we minimised the number of background applications (like Windows update). Second, we defined “relevant time intervals” from just before the download request to just after the rendering was completed, marked these events in our traces by ping messages, and then ignored all traffic but TCP flows set up and closed down within this interval. Third, we flushed the client DNS caches prior to each request, and then ignored all traffic with IP addresses other than those obtained by DNS requests during the relevant time interval.

The resulting traces where run through a specially written programme that reads pcap files, detects ping messages, reads DNS messages, identifies and follows TCP flows and computes the various metrics of interest regarding content and concurrency as detailed above.

A. Categorisation

To begin with, we examined whether the sites in the three studied content categories have the same content and concurrency characteristics. To this end, we applied a K-means clustering to both data sets, and the results for $K = 3$, corresponding to our three categories, are summarised in Tab. III.

The clusters on content are based on the seven metrics for pages (number of domains, addresses, flows, downstream packets and bytes, and upstream packets and bytes) while the clusters on concurrency are based on the two metrics for flow-packet concurrency (number of flows and number of packets). In a first attempt, we only used averages (red, green and blue in Tab. III), and, in a second attempt, we added the CoVs (cyan, magenta and yellow in Tab. III). The last row gives the percentages of the total variance that can be explained by the clusters. It is seen that the clusters captured most of the variance, but that they were different, and that neither of them followed the manually defined categories. We thus conclude that page grouping by content category is not meaningful from a modelling perspective, and henceforth ignore the notion of categories.

B. Content

Next, we examined the content characteristics of all sites, irrespective of content category; we investigated whether there were significant differences between computer and smartphone versions of web sites, and whether the same content characteristics held true at several levels of granularity.

Tab. I shows a summary of the page content characteristics in terms of averages (upper entry) and CoVs (lower entry) over all downloads of the selected pages. It is seen that there were significant differences between the versions aimed at computers and smartphones. First, computer versions had more domains, addresses, flows and upstream data (packets
and bytes) whereas, somewhat surprisingly, smartphone versions had more downstream data (packets and bytes). From the per-page data (not shown), it follows that this “anomaly” partly was an effect of the very large smartphone versions of YouTube and DailyMotion (video), but that the same relationship also applied to Expressen (newspaper) and Ebay (e-commerce). Second, comparing downloads to uploaded data, we note that computer versions were more balanced than smartphone versions; each uploaded packet (byte) corresponded to 4.5 downloaded packets (15 downloaded bytes) for computer versions, and to 6.5 downloaded packets (26 downloaded bytes) for smartphone versions. While this order of magnitude may seem natural, the per-page data reveals large variations, and that the opposite order applies to four of the pages: Aftonbladet (newspaper), Blocket and Tradera (e-commerce) and SVT (video). It is also seen that many of the CoVs were large or very large. Noting that none of our metrics can take negative values, it is clear that the large CoVs indicate the existence of some very large values.

C. Concurrency

Finally, we examined the concurrency characteristics of all sites. In the same way as was done in Section IV-B, we studied the differences between computer and smartphone versions of web sites, and whether the same concurrency characteristics were seen at different levels of granularity.

We start by considering the potential concurrency for an arbitrary flow \( f \) related to an arbitrary page \( p \) and an arbitrary domain \( d \); with \( M_p \) flows related to \( p \), we have \( M_p \) flows all of which have a potential page concurrency of \( M_p - 1 \) and, similarly, with \( N_d \) flows related to \( d \), we have \( N_d \) flows all of which have a potential domain concurrency of \( N_d - 1 \).

The results from our experiments are given in Tab. IV. It is seen that the potential concurrency was considerable and that it was larger for computer than for smartphone versions. We also note that the variations were considerable and was more pronounced when limiting the scope to domains.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Mean</th>
<th>CoV</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>486.246</td>
<td>0.787</td>
<td>1096</td>
</tr>
<tr>
<td></td>
<td>36.369</td>
<td>3.776</td>
<td>747</td>
</tr>
<tr>
<td>Smartphone</td>
<td>295.585</td>
<td>0.432</td>
<td>482</td>
</tr>
<tr>
<td></td>
<td>12.696</td>
<td>1.432</td>
<td>89</td>
</tr>
</tbody>
</table>

Next, we considered observed concurrency, and our first metric, flow coincidence, which captures overlap in time. Tab. V gives the results from our experiments, in terms of cumulated total \( O_t \) and time average \( O_a \), for all flows of a page and for all flows from the same domain.

It is seen that the number of coinciding flows was considerably larger for computer versions than for smartphone versions, and that the number of overlapping flows in both cases were large. We also note that the total overlaps were considerably larger than the averaged ones. It is emphasised that our findings have direct implications on the memory requirements in PEPs and other stateful middleboxers and we note that the results suggest that the prevalence of different terminals (computers or smartphones) has a significant impact, and that even single pages can consume considerable resources.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Mean</th>
<th>CoV</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>185.0</td>
<td>0.847</td>
<td>1572</td>
</tr>
<tr>
<td></td>
<td>108.5</td>
<td>5.759</td>
<td>292</td>
</tr>
<tr>
<td></td>
<td>4.751</td>
<td>4.933</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>26.99</td>
<td>0.771</td>
<td>473</td>
</tr>
<tr>
<td></td>
<td>14.21</td>
<td>5.106</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>2.827</td>
<td>3.320</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>1.950</td>
<td>2.649</td>
<td>20</td>
</tr>
</tbody>
</table>

We now continue with the second metric, flow contention, which captures packet interaction.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Mean</th>
<th>CoV</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>10</td>
<td>2.019</td>
<td>2.649</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>2.811</td>
<td>3.301</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1601</td>
<td>1.168</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1601</td>
<td>1.168</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>3059</td>
<td>1.152</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1601</td>
<td>1.168</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>3059</td>
<td>1.152</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab. VI shows the number of contending flows per direction, and it is seen that the number of coinciding flows typically was much higher than the number of contending flows, \( i.e. \), that overlap in time did not necessarily imply competition for resources, and that this difference was much more pronounced for computers than for smartphones. To see this, note that the average flow coincidence for a computer web page amounts to 185.0 flows in total (Tab. V) whereas the average flow contention with \( \Delta t = 30 \) ms only amounts to 16.01 flows on the downlink and 23.12 flows on the uplink (Tab. VI), \( i.e. \) the contention is about 10% of the coincidence. For smartphones the corresponding numbers are 26.99 coinciding flows compared to 10.41 and 15.01 contending flows in the two directions respectively, \( i.e. \) the contention is about 50% of the coincidence. These results have direct implications on ensemble sharing, which only is applicable to simultaneously active flows, and the results confirm that such technologies can be applied per end user both on entire pages (in PEPs) and on domain specific content (in servers). We also note that the significant concurrency is an important aspect of web traffic models for PEPs (and other performance enhancing solutions) since possible improvements (from, \( e.g. \) reducing round trip times) could be significant for single flows.
but may vanish for web pages (because of, e.g. congestion in one or both directions).

V. CONCLUSIONS

As seen from the reams of proposals for novel web enhancing technologies such as improved CCAs and advanced PEPs, improving web performance indeed is a challenging problem. This paper argues that a big obstacle is the lack of adequate web traffic metrics, and, as a consequence of this, a lack of relevant web traffic models beyond flow size. To this end we add, e.g., domain count to study the impact of DNS lookups and possible PEP caching, address count to study the impact of PEP TCP persistence policies, flow count to study the need for PEPs to open and close flows etc.

In particular, this paper also points to the lack of easy-to-use, yet relevant, metrics that reflect the concurrency characteristics of web traffic. To this end we explicitly differentiate between coincidence (which impacts, e.g., memory requirements in PEPs) and contention (which impacts, e.g., ensemble sharing in PEPs).

Finally, it is noted that both content and currency must be taken into account for PEPs to improve performance. That is, support for aggressive behaviour is useful for cases of underload, e.g., inflating the congestion window for single file transfers, whereas support for cautious behaviour is useful in cases of overload, e.g., reducing spurious timeouts for multiple file transfers in parallel.

To showcase the use of these metrics, a small experiment campaign was conducted. The results provide some novel insights, e.g. that smartphone-adapted web pages often encompass more data in terms of packets and bytes than the corresponding web pages for computers; that there is not always a strong correlation between the number of domains involved in the access of a web site and the total volume of traffic; that flow concurrency depends on if it is defined as coinciding flows or contending flows; that the number of coinciding flows depends on the terminal (computer or smartphone) and can be very large even for single pages; that the number of contending flows is less sensitive to the type of terminal and significantly smaller; and, finally, that clustering based on our metrics of content and concurrency does not result in distinct clusters for newspapers, e-commerce or video.

To further elaborate on these insights, more extensive experiment campaigns in different environments are planned. Future work will also consider ways to leverage the web traffic metrics in this paper in the design of web enhancing technologies such as PEPs and CCAs.

ACKNOWLEDGEMENT

The work presented in this paper is supported by COST action 1304 Autonomous Control for a Reliable Internet of Services (ACROSS).

REFERENCES


