Consolidated Review of

Measuring and Mitigating Web Performance Bottlenecks in Broadband Access Networks

1. Strengths:

This paper performs a large-scale measurement study of Web performance bottlenecks of broadband home networks. Given the growing importance of home networks in the Internet ecosystem, such study is crucial in improving user experience in home networks. Measurements from a home network testbed that are difficult to get.

The work systematically identifies the components that contribute to higher latency in web access. The system that is built in order to accurately measure each component is also impressive. The paper is very well written and the analysis is presented clearly.

Very nice data sets that form the basis of this work, and that are already made public.

A timely re-evaluation of a variety of web performance optimizations that were proposed in the earlier days of the web (generally speaking) but that are clearly worthy of consideration in light of today's higher speed broadband access. Comprehensive set of benchmark experiments and results

2. Weaknesses

The data are a bit "dirty" because of the nature of the experiments (i.e., they're running on "live" home access networks), making some conclusions a little bit murky.

One central concern with the work is that it is more of a well-balanced mixture of different already established techniques (such as DNS and TCP caching), and there is a limited amount of novelty in solving the high latency issue.

This paper does not study the performance impact of placing DNS caching, TCP caching and content caching on resource-constrained home routers, which typically have limited CPU, memory and storage capacity.

Many (but not all) measurements are based on synthetic benchmark (not web browser). No qualitatively novel findings the results relative to previous work

3. Comments

This paper is a worthwhile study of web performance on home residential networks. It is difficult for researchers to get access to a testbed that is as large and representative as the FCC/SamKnows deployment, so I think these results are a valuable contribution to the research community, if only to confirm previous more limited studies.

It is commendable that this paper has published the Web performance results from the BISmark experiments and software modules. Such data sets are very valuable for researchers without access to large-scale deployment of home routers. In related work, this paper should cite industry solutions such as keynote (http://www.keynote.com/mykeynote/help/components.asp) and Gomez (now part of compuware, http://www.compuware.com/content/compuware/en_us/application-performance-management/products/application-aware-network-monitoring.html) that provide commercial solutions for measuring different components of Web application performances.

I liked this paper a great deal. The overall improvement shown in Fig 13 was pretty impressive.

A criticism of this paper might be that it uses known and previously established performance improvement techniques, but I agree with the authors that looking at these improvements and issues from the perspective of the home access router is new. I appreciated the comparison of the Mirage and phantomjs tools, especially the "closing the loop" bit at the end of the paper. However, I still found the discrepancies between these tools a bit disturbing. The discussion indicates that one difference is that non-static links aren't downloaded (e.g., any AJAX-retrieved objects), but is that the only difference? A little more discussion on the specific nature of differences would help to understand the discrepancies of Figure 2.

I think the authors could have done more to alleviate the concerns about using the Mirage system vs. a real web browser (or emulator like Phantomjs) for the benchmarks. For example, in Figure 2, not only does Mirage underestimate the page load times (which the authors acknowledge and address), but the load times of Mirage are much more uniform than Phantomjs. I suspect this is because the strict staging of object loads in Mirage ignores much of the webpage optimization work that plays a crucial role in webpage load times for modern browsers. In addition, it isn't clear to me how Mirage handles recursive object loads (e.g., if it loads and iframe, does it also load all the embedded objects in the iframe? Does it recursively stagger these objects loads in the same way?) Other modern browser optimizations such as HTTP pipelining, gzip compression, and SPDY also would make huge differences in how long it takes for web pages to load. I do believe the results are still qualitatively useful despite the lack of analysis of these various options, but I would very much like to see some comparison or at least a discussion of how these optimizations would influence the results. Without such details, it is difficult for a website engineer or ISP operator to take anything away from this paper more than "latency matters more than throughput", which isn't exactly a novel finding. Also, the paper compares Mirage to Phantomjs, but is phantomjs actually representative of a modern web browser? A citation or result would be useful here.

This paper uses 9 targets in the experiments with all but cnn and Google being the home pages of these popular Web sites. Why use edition.cnn.com and www.google.com/mobile, instead of www.cnn.com and www.google.com? This paper uses Phantomjs on an emulated 10Mbps access link with a last-mile latency of 40ms. In Tables 3 and 4, all of the average last-mile latencies are less than 40ms, so it is desirable to use a smaller last-mile latency for running experiments with Phantomjs. This paper proposes to use TCP caching to maintain TCP connections to popular sites for reducing overhead for new connections. However, what is the impact of TCP caching on the server side if millions of home routers use the same strategy? It is true that the caching mechanism on home routers complements existing optimizations on browsers and end hosts. However, the system cost of running optimizations on these different locations could be very different. For example, caching high-quality images of the logos of Web sites on browsers has little impact on the system performances of a laptop or a desktop, but doing the same actions on home routers could require non-negligible resources from resource-constrained home routers.

The differences between SamKnows and BISmark in the last 3 columns of Table 2 should be explained better. Especially for the large difference for the Google row, it was not clear what was behind that, and the difference will certainly cause performance differences between the two platforms.

A related question: are all the SamKnows and BISmark devices all of the same generation of hardware? I was wondering whether that could be a possible reason behind some of the observed variability. The data collected from the various experiments is a little muddy, simply because there's no way to control the network beyond the broadband access point.

The authors did a pretty reasonable job in analyzing the data, given what they could control, but some plots show some of the ugliness. For example, in Figures 7 and 10, there is some small fraction of data points in which a proposed improvement (e.g., DNS caching) results in higher latency. Clearly unexpected, but almost certainly due to factors outside the authors control. Nevertheless, it would have been nice to have some explicit discussion of these oddities. Specifically with respect to Figs 7 and 10, there wasn't any text acknowledging/describing these behaviors.

The system that is built in order to collect the data is very well designed. Especially, the deployment of Mirage and BISmark, and also the experiment setup of Section 6.1 are very good. The main takeaway of the measurement part that latency is more critical to web access performance when throughput is high is already known. In a way, a large number of CDN optimizations (about placement etc.) target the same issue. If one considers this to be the conclusion of measurement study (from Table 5), then the insights presented in all sections before Section 5 is not necessarily novel. It seems that in and before Section 5, there are too many plots but the actual novel findings are not that many. The home caching system is a nice mixture of already established concepts.

DNS cache and TCP connection cache. Both the caches will have serious scalability issues when implemented in home routers. This is not observed in the paper because only top 8-9 websites are chosen for the study but when diverse users access different kinds of content, how will the caches (specially connection cache) scale? There should be some discussion about which is a more practical way of reducing latency.

DNS+connection caching or some CDN-based optimization? It seems that the latter would be a more practical and scalable solution also from an ISP's point-of-view. It is not explained why the error bars significantly large in Fig. 11? Are there any other factors affecting the load time when implemented in practice that are not studied in the work?

4. Summary from PC Discussion

In the PC meeting, the reviewers discussed the fact that this work seemed to be a timely reevaluation of well-known techniques, but in a somewhat different environment. The reviewers were happy to see that measurements will be (or have been) made public to the community, and felt that the analyses of the various causes of latency were quite solid. There were some concerns mentioned that the latency reduction techniques are well known, and that some of the measurement techniques may have introduced some biases. There was also the concern voiced that the proposed techniques may not scale well (both on resource-constrained home routers, and on end servers) if deployed. Still, the reviewers felt that the work was quite solid, and recommended acceptance.

5. Authors' Response

The implementation differences between Mirage and real browsers imply that the page load times that Mirage sees may not reflect the times that any real browser would see. Page load times will always differ across different browsers, and we do not aim to estimate page load time from any particular browser. Our goal is to illustrate how components of network latency (e.g., DNS lookup time, TCP connect time) contribute to Web page load times. Mirage decomposes Web page load time into these components, which will be the same regardless of browser or any optimizations that a browser might perform. Mirage also allows us to evaluate how optimizations that a browser might perform can mitigate various network bottlenecks under different network conditions.

The reviewers note that the SamKnows and the BISmark studies see slightly different characteristics for each Web site, which could be explained either by differences in vantage point or the time that the measurements were conducted. Because we do not compare measurements across the two deployments, however, these differences are not consequential.

Although SPDY and QUIC can mitigate the bottlenecks we reveal, they would still be subject to latency bottlenecks, especially in access networks. In this paper, we study these effects over HTTP only; future work could explore how network performance characteristics interact with SPDY and QUIC.

Some reviewers questioned the novelty of our findings. We believe that our study offers new insights into Web performance, because we conducted measurements in the context of access networks, which are becoming increasingly prevalent. Although some findings may be known in general, our extensive characterization of Web performance over a range of specific network conditions and real deployments shed light on how latency is commonly a bottleneck. Previous work has not quantified the effects of these components in real home networks, nor has it performed such an extensive, controlled evaluation of the effects of these components.

CDNs can reduce end-to-end latency, but they cannot eliminate last-mile latency; in many parts of the world, CDN caches are not close enough to users to eliminate latency effects. We acknowledge that prefetching DNS entries and caching TCP connections may be expensive, but many browsers perform these optimizations already. The paper shows that even lightweight caching can yield high hit rates: prefetching and caching only 20 sites with a timeout of only 20 minutes can yield significant performance gains without imposing prohibitive overhead.