Poster: HTTP/2 Performance in Cellular Networks

Utkarsh Goel
Montana State University
utkarsh.goel@montana.edu

Moritz Steiner
Akamai Technologies, Inc.
moritz@akamai.com

Mike P. Wittie
Montana State University
mwittie@cs.montana.edu

Martin Flack
Akamai Technologies, Inc.
mflack@akamai.com

Stephen Ludin
Akamai Technologies, Inc.
sludin@akamai.com

1. INTRODUCTION

HTTP/2 (h2) was standardized in 2015 as an improvement to HTTP/1.1 (h1) to achieve faster webpage load times (PLTs) [5]. Previous studies have shown both improvement and degradation in PLT when using h2 with respect to h1 [6, 8]. The disagreement about h2 performance from these studies motivates further investigation as to whether and under what conditions h2 brings the performance benefits that were originally envisioned [5].

In this paper, we investigate performance of h2 and h1 by first understanding the dynamic nature of cellular network characteristics (in terms of loss, latency, and bandwidth) and then exposing its implications on the PLT when using h2. Our goal is to understand how h2 impacts PLT when cellular network delay is interpreted as loss by server-side TCP sockets. Therefore, for this paper we focus only on connections with loss, as interpreted by TCP. Specifically, we make the following three contributions in this paper.

Dataset: Our analysis of real world cellular network characteristics is based on 6200 TCP connections captured over several hours on an Akamai CDN cluster hosted inside a datacenter of a major cellular network provider in the US. We observe that about 2000 connections (32%) experience loss, out of which about 500 connections experience loss more than once within the first few seconds. The median connection duration, total number of TCP segments and bytes exchanged between clients and servers during our capture is about 2.3 seconds, 26 segments, and 15 KB respectively.

Measurement: We model the emulation on real-world TCP traces to adequately represent the correlation of loss in cellular networks, as interpreted by TCP. Our simulation is designed to emulate cellular networks in terms of packet loss interpreted by the server (in the form of retransmissions), time between loss events, round trip time (RTT), and bandwidth. From such a comprehensive view of cellular network characteristics, we argue that our technique improves existing network emulation techniques.

Insights: Results from our investigation of h2 performance are threefold. First, for a webpage with several hundreds of small sized objects, h2 outperforms h1, except in the scenarios when cellular networks frequently experience high loss rates. Second, for a webpage with large objects, h2 PLTs are significantly higher than h1. And third, for a webpage with a relatively large number of both small and medium sized objects, h2 outperforms h1; however as the connection starts experiencing loss, the performance gain in h2 degrades, resulting in PLTs slower than h1.

MobiCom'16 October 03-07, 2016, New York City, NY, USA

ACM ISBN 978-1-4503-4226-1/16/10.

DOI: http://dx.doi.org/10.1145/2973750.2985264

2. EXPERIMENTAL SETUP

We selected an Akamai CDN cluster located inside the network of a major cellular ISP in the US [2]. On each server in the cluster, we ran TCPDump for several hours at different times of day to capture incoming and outgoing TCP segments. Given that the selected cellular network does not split TCP connections between clients and servers for HTTPS sessions [7], we only capture TCP segments for HTTPS traffic. Our TCP traces reflect the characteristics of a real world cellular network as the TCP connections to the selected Akamai CDN cluster are not influenced by any interference from the public Internet [2].

Next, from each packet capture file, we use tshark to extract the number of frames and bytes exchanged, the number of frames retransmitted by the server, and the time interval between acknowledgments [4]. We extract the above metrics at 70 ms intervals, where the first interval starts when the *TCP SYN* is received by the server. The choice of 70 ms matches the median RTT between clients in the selected cellular network and Akamai CDN servers [7]. From the collected packet captures, we make four observations.

First, about 32% of the TCP connections experience loss, which is high compared to wired last-mile networks.

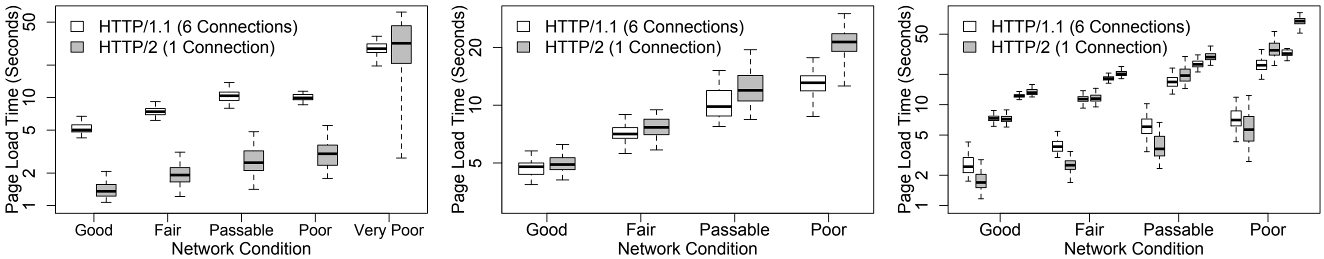
Second, losses in cellular networks are often clustered, such that when a loss event occurs, many consecutive TCP segments are retransmitted by server. From our analysis we observe that many connections experience multiple retransmissions at different times during the connection. For example, for the time interval finishing at 420 ms, we observe that for multiple TCP connections servers retransmit 5, 10, and even 20 packets.

Third, we observe that TCP connections experience retransmissions at multiple times during their lifetime. Our analysis shows that for several connections, subsequent retransmissions appear within one second interval.

And **fourth**, when a retransmission event appears, the majority of connections experience about 5-15% packet loss during 70 ms time intervals.

2.1 Emulating Cellular Networks

In our emulation we introduce clustered loss only at specific times during the connection, in addition to modifying bandwidth, and RTT every 70 ms. We develop distributions of 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} percentile values of retransmission rate, time gap between retransmission clusters, throughput, and RTT, as observed across all TCP connections respectively. Using these distributions, in Table 1, we develop scenarios to emulate various conditions.



(a) 365 objects of 2 KB each (b) 10 objects of 435 KB each (c) 136 objects of 1 KB to 620 KB each

Figure 1: Distribution of page load times when loaded over `h1` and `h2` under various network conditions.

Table 1: Emulation of various network conditions.

Condition	Loss	Time Gap	Throughput	RTT
Good	p10	p90	p90	p10
Fair	p25	p75	p75	p25
Passable	p50	p50	p50	p50
Poor	p75	p25	p25	p75
Very Poor	p90	p10	p10	p90

For example, when emulating a *Good* experience, we select the 10^{th} percentile (p10) loss distribution, 10^{th} percentile RTT (p10) distribution, 90^{th} percentile (p90) throughput (used as bandwidth in our emulation) distribution, and 90^{th} percentile (p90) distribution of time gap between loss.

We design our emulation to dynamically update network links in terms of the above four distributions, as shown in Table 1. Specifically, at every 70 ms we update RTT by randomly selecting a value from its respective distribution. Since bandwidth in cellular networks is attributed to base stations and is therefore not dependent on loss and RTT [9], we also change bandwidth every 70 ms. Finally, we introduce packet loss only when the selected time gap value has passed during emulation.

Next, we setup a network topology using three machines to emulate a client, a server, and a bridge. On the client, we run Chromium Telemetry to load different webpages over 100 times using Google Chrome [1]. On the server, we configure Apache to support `h1` and `h2`. On the bridge, we emulate network conditions using `TC NETEM`. Depending on the emulated scenario, we configure the bridge to modify the network loss, time of the next loss, RTT, and bandwidth after every 70 ms per the scenarios defined in Table 1.

3. PRELIMINARY RESULTS

In Figure 1(a), we compare PLTs for a webpage with 365 objects of average size of 2 KB, loaded in turn over `h1` and `h2` connections across the different emulated conditions. We observe that PLTs over `h2` are lower than PLTs over `h1`, because for `h1` with 6 TCP connections, the server can only send 6 objects in parallel, whereas in the case of `h2` with many streams multiplexed onto one connection, the server sends a large number of objects in parallel. Further, as the network condition becomes *Very Poor*, the PLTs increase for both `h1` and `h2`, but more so for `h2`. For `h1`, the impact of packet loss on any one of the 6 connections only affects that particular connection; in the case of `h2`, since all object downloads are multiplexed over a single TCP connection, packet loss affects all ongoing object downloads.

Next, we investigate how `h2` performs with multiplexed responses using all large objects. In Figure 1(b), we compare PLTs for a webpage with 10 large objects of size 435 KB each. In general, we observe that `h1` outperforms `h2` across all network conditions, especially in *Poor* conditions where loss occurs frequently. We argue that for `h2` with one connection, the initial congestion window (ICW) size of the server during the TCP slow start is only one-sixth of the cumulative window sizes of `h1` with 6 parallel TCP connections. Thus, the server sends six times less data over `h2` during the TCP slow start phase. We confirm the claim as when we compare PLTs without any loss, we observe that `h1` still outperforms `h2`. Further, as the network conditions worsen, the congestion window of the single `h2` connection does not grow. For `h1` the cumulative congestion window size remains larger than the window size of single `h2` connection.

Finally, in Figure 1(c), we compare PLTs for a webpage (modeled from HTTP Archive data [3]) with 136 objects whose size ranges from 1 KB to 620 KB. In addition to measuring PLTs over `h1` and `h2` in different network conditions, we investigate the impact of webpage size on PLT, while keeping the number of objects constant. The first, second, and third pairs of boxplots in Figure 1(c) under each network condition represent the distribution of PLTs for webpages of size 2 MB, 8 MB, and 12 MB, respectively. From the figure we observe that for a 2 MB page, PLTs over `h2` are lower than PLTs over `h1`, as (similarly to Figure 1(a)) server sends multiple small sized objects in parallel during the TCP slow start, whereas server sends only six objects in parallel in the case of `h1`.

Although for the 8 MB page, PLTs over `h2` and `h1` are comparable under *Good* and *Fair* network conditions, however, as the conditions worsen, PLTs over `h2` become larger than `h1`. Moreover, for the 12 MB page, the PLTs over `h2` are always higher than `h1`. Similarly to Figure 1(b), under lossy conditions the congestion window on the server does not grow as much and as fast as it grows cumulatively for six `h1` connections, thus affecting the PLTs over `h2` when downloading large objects. We observe that the slow start phase is less important here as most of the PLT comes from the congestion avoidance phase. To confirm whether the ICW impacts PLTs, we loaded webpages 25 MB in size (not shown) and observed no statistical significance in the difference between `h2` and `h1` PLTs.

Acknowledgments We thank National Science Foundation for supporting this work through grants CNS-1555591 and CNS-1527097.

4. REFERENCES

[1] Chromium Telemetry. https://goo.gl/gFxs7x, Dec. 2014.

[2] Akamai Accelerated Network Partner. https://goo.gl/ZhkNH1, Jun. 2016.

[3] HTTP Archive. http://httparchive.org/interesting.php, Jun. 2016.

[4] Tshark. https://goo.gl/YmGrjE, Jun. 2016.

[5] M. Belshe, R. Peon, and E. M. Thomson. Hypertext Transfer Protocol Version 2 (HTTP/2), RFC 7540, May 2015.

[6] H. de SaxcÃl’, I. Oprescu, and Y. Chen. Is HTTP/2 really faster than HTTP/1.1? In *IEEE INFOCOM*, Apr. 2015.

[7] U. Goel, M. Steiner, M. P. Wittie, M. Flack, and S. Ludin. Detecting Cellular Middle-boxes using Passive Measurement Techniques. In *Passive and Active Measurements Conference (PAM)*, Mar. 2016.

[8] M. Varvello, K. Schomp, D. Naylor, J. Blackburn, Alessandro, Finamore, and K. Papagiannaki. Is The Web HTTP/2 Yet? In *Passive and Active Measurements Conference (PAM)*, Mar. 2016.

[9] K. Winstein, A. Sivaraman, and H. Balakrishnan. Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks. In *USENIX NSDI*, Apr. 2013.