We see that the benefits vary across different websites. Our packet traces indicate that the amount of outstanding data (and hence throughput) is quite similar in both the cases. The number of retransmitted packets seem similar under good conditions, but disabling the parameter runs the risk of having lots of retransmissions under congestion or poor channel conditions since the `cwnd` value is inaccurate after an idle period. In some instances, `cwnd` grows so large with the parameter disabled, that the receive window becomes the bottleneck and negates the benefit of a large congestion window at the sender.

6.2.3 *Impact of TCP variants*

We replaced TCP Cubic with TCP Reno to see if modifying the TCP variant has any positive impact on performance. We find in Table 2 that there is little to distinguish between Reno and Cubic for both HTTP and SPDY over 3G. We see that the average page load time across all the runs of all pages is better with Cubic. Average throughput is quite similar with Reno and Cubic, with SPDY achieving the highest value with Cubic. While this seemingly contradicts the result in Figure 9, note that this result is the average across all times (ignoring idle times), while the result in Figure 9 considers the average at that one second instant. Indeed the maximum throughput result confirms this: HTTP with Cubic achieves a higher throughput than SPDY with Cubic. SPDY with Reno does not grow the congestion window as much as SPDY with Cubic. This probably results in SPDY with Reno having the worst page load time across the combinations.

	Reno		Cubic	
	HTTP	SPDY	HTTP	SPDY
Avg. Page Load (msec)	9690.84	9899.95	9352.58	8671.09
Avg. Throughput (KBps)	121.88	119.55	115.36	129.79
Max. Throughput (KBps)	1024.74	528.88	889.33	876.98
Avg. `cwnd` (# segments)	10.45	24.16	10.59	52.11
Max. `cwnd` (# segments)	22	48	22	197

Table 2: Comparison of HTTP and SPDY with different TCP variants.

6.2.4 *Cache TCP Statistics?*

The Linux implementation of TCP caches statistics such as the slow start threshold and round trip times by default and reuses them when a new connection is established. If the previous connection had statistics that are not currently accurate, then the new connection is negatively impacted. Note that since SPDY uses only one connection, the only time these statistics come into play is when the connection is established. It can potentially impact HTTP, however, because HTTP opens a number of connections over the course of the experiments. We conducted experiments where we disabled caching. Interestingly, we find from our results that both HTTP and SPDY experience reduced page load times. For example, for 50% of the runs, the improvement was about 35%. However, there was very little to distinguish between HTTP and SPDY.

7. RELATED WORK

Radio resource management: There have been several attempts to improve the performance of HTTP over cellular networks (e.g. [10,12]). Specifically, TOP and TailTheft study efficient ways of utilizing radio resources by optimizing timers for state promotions and demotions. [5] studies the use of caching at different levels (e.g., nodeB, RNC) of a 3G cellular network to reduce download latency of popular web content.

TCP optimizations: With regards to TCP, several proposals have tried to tune TCP parameters to improve its performance [14] and address issues like Head of Line (HOL) blocking and multi-homing. Recently, Google proposed in an IETF RFC 3390 [4] to increase the TCP initial congestion window to 10 segments to show how web applications will benefit from such a policy. As a rebuttal, Gettys [6] demonstrated that changing the initial TCP congestion window can indeed be very harmful to other real-time applications that share the broadband link and attributed this problem to one of "buffer bloat". As a result Gettys, proposed the use of HTTP pipelining to provide improved TCP congestion behavior. In this paper, we investigate in detail how congestion window growth affects download performance for HTTP and SPDY in cellular networks. In particular, we demonstrate how idle-to-active transition at different protocol layers results in unintended consequences where there are retransmissions. Ramjee et al. [3] recognizes how challenging it can be to optimize TCP performance over 3G networks exhibiting significant delay and rate variations. They use an ACK regulator to manage the release of ACKs to the TCP source so as to prevent undesired buffer overflow. Our work inspects in detail how SPDY and HTTP behave and thereby TCP in cellular networks. Specifically, we point out a fundamental insight with regards to avoiding spurious timeouts. In conventional wired networks, bandwidth changes but the latency profile does not change as significantly. In cellular networks, we show that spurious timeout is caused by the fact that TCP stays with its original estimate for the RTT and a tight retransmission timeout (RTO) estimate derived over multiple round-trips during the active period of a TCP connection is not only invalid, but has significant performance impact. Thus, we suggest using a more conservative way to manage the RTO estimate.

8. CONCLUSION

Mobile web performance is one of the most important measures of users' satisfaction with their cellular data service. We have systematically studied, through field measurements on a production 3G cellular network, two of the most prominent web access protocols used today, HTTP and SPDY. In cellular networks, there are fundamental interactions across protocol layers that limit the performance of both SPDY as well as HTTP. As a result, there is no clear performance improvement with SPDY in cellular networks, in contrast to existing studies on wired and WiFi networks.

Studying these unique cross-layer interactions when operating over cellular networks, we show that there are fundamental flaws in implementation choices of aspects of TCP, when a connection comes out of an idle state. Because of the high variability in latency when a cellular end device goes from idle to active, retaining TCP's RTT estimate across this transition results in spurious timeouts and a corresponding burst of retransmissions. This particularly punishes SPDY which depends on the single TCP connection that is hit with the spurious retransmissions and thereby all the cascading effects of TCP's congestion control mechanisms like lowering `cwnd` etc. This ultimately reduces throughput and increases page load times. We proposed a holistic approach to considering all the TCP implementation features and parameters to improve mobile web performance and thereby fully exploit SPDY's advertised capabilities.

9. REFERENCES

[1] 3GPP TS 36.331: Radio Resource Control (RRC). `http://www.3gpp.org/ftp/Specs/html-info/36331.htm`.

[2] Squid Caching Proxy. `http://www.squid-cache.org`.

[3] Chan, M. C., and Ramjee, R. TCP/IP performance over 3G wireless links with rate and delay variation. In *ACM MobiCom* (New York, NY, USA, 2002), MobiCom '02, ACM, pp. 71–82.

[4] Chu, J., Dukkipati, N., Cheng, Y., and Mathis, M. Increasing TCP's Initial Window. `http://tools.ietf.org/html/draft-ietf-tcpm-initcwnd-08.html`, Feb. 2013.

[5] Erman, J., Gerber, A., Hajiaghayi, M., Pei, D., Sen, S., and Spatscheck, O. To cache or not to cache: The 3g case. *IEEE Internet Computing 15*, 2 (2011), 27–34.

[6] Gettys, J. IW10 Considered Harmful. `http://tools.ietf.org/html/draft-gettys-iw10-considered-harmful-00.html`, August 2011.

[7] Google. SPDY: An experimental protocol for a faster web. `http://www.chromium.org/spdy/spdy-whitepaper`.

[8] Kalampoukas, L., Varma, A., Ramakrishnan, K. K., and Fendick, K. Another Examination of the Use-it-or-Lose-it Function on TCP Traffic. In *ATM Forum/96-0230 TM Working Group* (1996).

[9] Khaunte, S. U., and Limb, J. O. Statistical characterization of a world wide web browsing session. Tech. rep., Georgia Institute of Technology, 1997.

[10] Liu, H., Zhang, Y., and Zhou, Y. Tailtheft: leveraging the wasted time for saving energy in cellular communications. In *MobiArch* (2011), pp. 31–36.

[11] Popa, L., Ghodsi, A., and Stoica, I. HTTP as the narrow waist of the future internet. In *Hotnets-IX* (2010), pp. 6:1–6:6.

[12] Qian, F., Wang, Z., Gerber, A., Mao, M., Sen, S., and Spatscheck, O. TOP: Tail Optimization Protocol For Cellular Radio Resource Allocation. In *IEEE ICNP* (2010), pp. 285–294.

[13] Shruti Sanadhya, and Raghupathy Sivakumar. Adaptive Flow Control for TCP on Mobile Phones. In *IEEE Infocom* (2011).

[14] Stone, J., and Stewart, R. Stream Control Transmission Protocol (SCTP) Checksum Change. `http://tools.ietf.org/html/rfc3309.html`, September 2002.

[15] W3techs.com. Web Technology Surveys. `http://w3techs.com/technologies/details/ce-spdy/all/all.html`, June 2013.

[16] Wang, X. S., Balasubramanian, A., Krishnamurthy, A., and Wetherall, D. Demystifying Page Load Performance with WProf. In *Usenix NSDI'13* (Apr 2013).

[17] Welsh, M., Greenstein, B., and Piatek, M. SPDY Performance on Mobile Networks. `https://developers.google.com/speed/articles/spdy-for-mobile`, April 2012.

[18] Winstein, K., Sivaraman, A., and Balakrishnan, H. Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks. In *Usenix NSDI'13* (Apr 2013).

APPENDIX

A. CELLULAR STATE MACHINES

The radio state of every device in a cellular network follows a well-defined state machine. This state machine, defined by 3GPP [1] and controlled by the radio network controller (in 3G) or the base station (in LTE), determines when a device can send or receive data. While the details of the

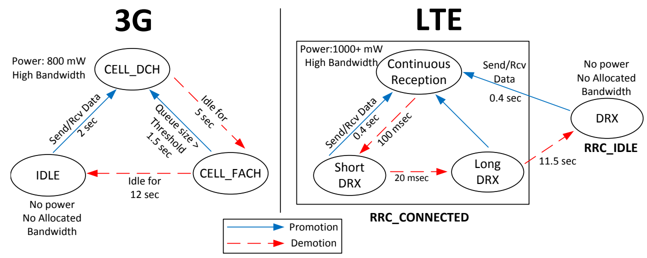

Figure 18: The RRC state machines for 3G UMTS and LTE networks

states, how long a device remains in each state, and the power it consumes in a state differ between 3G and LTE, the main purpose is similar: the occupancy in these states control the number of devices that can access the radio network at a given time. It enables the network to conserve and share available radio resources amongst the devices and for saving the device battery at times when the device does not have data to send or receive.

3G state machine: The 3G state machine, as shown in Figure 18, typically consists of three states: $IDLE$, Forward access channel ($CELL\_FACH$) and Dedicated channel ($CELL\_DCH$). When the device has no data to send or receive, it stays in the $IDLE$ state. The device does not have radio resource allocated to it in $IDLE$. When it wants to send or receive data, it has to be *promoted* to the $CELL\_DCH$ mode, where the device is allocated dedicated transport channels in both the downlink and uplink directions. The delay for this promotion is typically ∼2 seconds. In the $CELL\_FACH$, the device does not have a dedicated channel, but can transmit at a low rate. This is sufficient for applications with small amounts or intermittent data. A device can transition between $CELL\_DCH$ and $CELL\_FACH$ based on data transmission activity. For example, if a device is inactive for ∼5 seconds, it is *demoted* from $CELL\_DCH$ to $CELL\_FACH$. It is further demoted to $IDLE$ if there is no data exchange for another ∼12 secs. Note that these state transition timer values are not general and vary across vendors and carriers.

LTE state machine: LTE employs a slightly modified state machine with two primary states: RRC_IDLE and $RRC\_CONNECTED$. If the device is in RRC_IDLE and sends or receives a packet (regardless of size), a state promotion from RRC_IDLE to $RRC\_CONNECTED$ occurs in about 400 msec. LTE makes use of three sub-states within $RRC\_CONNECTED$. Once promoted, the device enters Continuous Reception state where it uses considerable power (about 1000mW) but can send and receive data at high bandwidth. If there is a period of inactivity (e.g., for 100 msec), the device enters the short Discontinuous Reception ($Short\_DRX$) state . If data arrives, the radio returns to the Continuous Reception state in ∼400 msec. If not, the device enters the long Discontinuous Reception ($Long\_DRX$) state. In the $Long\_DRX$ state, the device prepares to switch to the RRC_IDLE state, but is still using high power and waiting for data. If data does arrive within ∼11.5 seconds, the radio returns to the Continuous Reception state; otherwise it switches to the low power (< 15 mW) RRC_IDLE state. Thus, compared to 3G, LTE has significantly shorter promotion delays. This shorter promotion delay helps reduce the number of instances where TCP experiences a spurious timeout and hence an unnecessary retransmission(s).