Towards Understanding TCP Performance on LTE/EPC Mobile Networks

Binh Nguyen[1], Arijit Banerjee[1], Vijay Gopalakrishnan[2], Sneha Kasera[1],
Seungjoon Lee[2], Aman Shaikh[2], and Jacobus Van der Merwe[1]

[1]School of Computing, University of Utah

[2]AT&T Labs - Research

{binh,arijit,kasera,kobus}@cs.utah.edu

{gvijay,slee,ashaikh}@research.att.com

Abstract - The popularity of smartphones and smartphone applications means that data is the dominant traffic type in current mobile networks. In this paper we present our work on a systematic investigation into facets of the LTE/EPC architecture that impact the performance of TCP as the predominant transport layer protocol used by applications on mobile networks. We found that (1) load increase in a cell causes dramatic bandwidth reduction on UEs and significantly degrades TCP performance, (2) seamless handover causes significant TCP losses while lossless handover increases TCP segments' delay.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Applications

Keywords

TCP performance; mobile networking; LTE/EPC networks

1. INTRODUCTION

The emergence of smartphones and smartphone app markets resulted in data overtaking voice as the predominant traffic type on mobile networks [16]. The growth and importance of data traffic is expected to continue as packet-based mobile network architectures, specifically networks based on long term evolution (LTE) and evolved packet core (EPC) technologies, offer substantially higher user throughput and lower delay compared to previous technologies. For example, the Cisco mobile data traffic forecast white paper [13] reported that in 2013 the average mobile downstream speed was 1,387 Kbps (doubled from 2012) and 4G connections account for 30 percent of all mobile data traffic even though they represent only 2.9 percent of mobile connections.

The growing importance of data in mobile networks suggests the need to understand the impact of mobility protocols and the mobile environment on the performance of data traffic and applications. Indeed this need is widely recognized by the networking research community with a plethora of recent papers dealing with various aspects of characterizing and measuring [26, 12] and improving [24, 8, 20] data performance on mobile networks. Unsurprisingly, several of these works have uncovered performance problems associated with the mobile networking environment, and have suggested possible causes. The bulk of these studies were performed *outside* of the mobile network proper, and as a result speculations about the root cause of observed performance problems have by and large not been verified.

In our work presented in this paper, we take a step towards a systematic understanding of the interactions between facets of the mobile network and Transmission Control Protocol (TCP). We attempt to *look under the hood* to investigate the impact of different mobile network architecture and environment aspects on the performance of TCP. We use a combination of the NS-3 simulation environment and a standard Linux kernel TCP stack to perform detailed simulations of two common scenarios: First, we consider the mobility-specific pathological case of a significant load increase in a particular cell cite. For example, this might occur because of neighboring base station (eNodeB in LTE nomenclature) failure or malfunction, or simply because of a convergence of mobile devices at the cell site in question. Second, we consider the impact of LTE handovers (due to user mobility) on the performance of ongoing TCP sessions. Towards this end, we delve deeper to consider the interaction between TCP and the user-plane protocols associated with the LTE radio access network.

We make the following contributions in this paper:

- We study the effects of load increase in a cell and find that load increases could lead to bandwidth throttling on UEs, sudden increases of end-to-end delay, and TCP timeouts.

- We extend the current NS-3 simulator to support lossless handover and RLC AM (Radio Link Control Acknowledged Mode).

- We study the behavior of TCP and performance impacts under two kinds of handovers supported by LTE: seamless and lossless. We show that seamless handovers cause significant losses that reduce TCP throughput, while TCP segments in lossless handovers experience significant delay increase.

Figure 1: Single UE simulation topology

2. METHODOLOGY

In order to systematically explore the interactions between the mobile network architecture and TCP, we have to look *under the hood* to consider mobile network elements and protocols. Since real-world mobile networks are proprietary and instrumenting network components (e.g., mobility network stack of an eNodeB) is difficult if not impossible, we turn to simulation using the NS-3 network simulator [10]. In addition to modeling TCP and other traditional network protocols, NS-3 includes support for both the radio access network (RAN) and the evolved packet core (EPC). Using NS-3, we simulate different scenarios and extract relevant system variables to understand the interactions across the layers.

We also use the Network Simulation Cradle (NSC) plug-in [14], which allows the execution of a full Linux networking stack within NS-3 simulations. This allows realism in our simulations by enabling us to emulate a typical Linux host and its default settings. For example we use TCP CUBIC [9], the default TCP variant in Linux, in our experiments even though it is not available in NS-3.

On the RAN side, NS-3 models both user plane (i.e., PDCP, RLC, MAC, and PHY) and control plane protocols (i.e., Non-access stratum (NAS), Radio Resource Control (RRC)). At the PHY layer, NS-3 enables fading and propagation models. It also supports realizing User Equipment (UE) instances using different position allocation models (e.g., grid position allocation, random box position allocation, etc) and mobility patterns. Unlike the RAN, NS-3 does not model the EPC in detail other than to provide a basic Serving- and Packet Data Network Gateway. However, since our focus in this paper is primarily on the RAN interactions and TCP, we can still leverage NS-3's capabilities in our study.

NS-3 Additions and changes: We make the following changes in NS-3 to support our experiments:

- 3GPP standards support two variants of handover: seamless and lossless. NS-3, by default, only supports seamless handover. We implement lossless downlink PDCP handover following 3GPP specifications [2].

- Lossless handovers function with RLC AM (Radio Link Control Acknowledged Mode). We therefore extend the incomplete NS-3 RLC AM implementation according to the 3GPP specification [1].

- We add hooks to the NSC kernel module to retrieve TCP variables such as congestion window (cwnd), slow-start threshold (ssthresh), CUBIC's parameter such as Wmax, K values, and TCP Retransmission Timeout (RTO) estimation.

Test Setup: Our experimental setup includes a simulated server, a simulated EPC (i.e., SGW and PGW), a simulated RAN with multiple eNodeBs, and one or more simulated UEs. The topology of our single UE experiment is shown in Figure 1. We assumed 70ms mean end-to-end RTT, with a 1:1.3 ratio between the Internet delay (S/P-GW to Server) and the cellular core network delay (eNodeB to S/P-GW) [12]. We assume that the latency between eNodeB and UE is on average 3ms [6]; this latency assumes a 0% HARQ error rate and does not include any queuing delay on the eNodeB. We consider the radio link between the eNodeB and the UE to be the

bottleneck link due to the usual limitation of wireless resources. For simplicity, we use a single server for all UE traffic flows. We model both pedestrian and vehicular mobility in our experiments: We use the Levy-Walk model [21] (which resembles Brownian motion in a small area ($400 \ m^2$ area)) to capture pedestrian mobility, while modeling vehicular mobility as a straight line with the UE moving with mean velocity of 40km/h (5km/h variance). To capture the effect of location in a cell, we repeat each experiment by placing UEs at random initial positions (using random distances between UEs and eNodeB).

RAN parameter	Value
Number of RBs	50 (one UE), 100 (multiple UEs)
MAC scheduler	proportional fair
RLC transmission queue	512KB
RLC mode	RLC AM (default)
Hybrid ARQ (HARQ)	Enabled
eNodeB Max Tx. power	46 dBm
eNodeB noise figure	5 dB
UE Max Tx. power	23 dBm
UE noise figure	9 dB

Table 1: RAN settings used in our simulations.

Radio Access Network settings: We mimic a realistic setting for the Radio Access Network. For our single UE experiment, we use 50 radio resource blocks to mimic the real world bandwidth obtained by a UE in LTE — if the full radio resource with 100 RBs is used, a UE would get unrealistically high bandwidth. These are standard values in LTE and depend on the channel bandwidth (10 MHz or 20MHz). We use a 512 KB transmission queue at the eNodeB. On the PHY layer, we use the Friis propagation loss model [19] and a trace-driven fading model. We use LTE band 4 (downlink central frequency 2132.5 MHz, uplink central frequency 1732.5 MHz) between UEs and the eNodeB. ENodeB and UE's transmission powers and noise figures mimic a macro cell setting [11]. The RAN parameters used are presented in Table 1.

Workload: The server and UEs have an application that sends and receives data. We use both UDP and TCP applications as follows. During the initial run of an experiment, the UDP application is used to determine the capacity of the radio link by having the sender application on the server sends packets at a high rate (i.e., 200Mbps) in order to saturate the link. In subsequent runs of the experiment (with identical parameters) we use the TCP application to evaluate the impact of the network conditions on TCP performance. For our TCP application, both the UEs and server use CUBIC with default TCP parameters except for the tcp_timestamps option, which we disable to reduce transmission overhead. We monitor both TCP layer variables and the performance at the application level. We use a single long lived TCP flow in our experiments and focus mainly on downloads to the UE (as opposed to uploads) since the bulk of the data in LTE today flows in the downlink direction [12].

3. EXPERIMENTS

The context, setup, metrics and varying factors, as well as the results of each experiment we performed are described below. For each experiment we also endeavor to discuss the practical implications of our results.

3.1 Load increase in a cell

While sudden load increase can happen with any network, sudden load increases are potentially more frequent and severe in mobile networks for two reasons. First, UEs can be within range of multiple
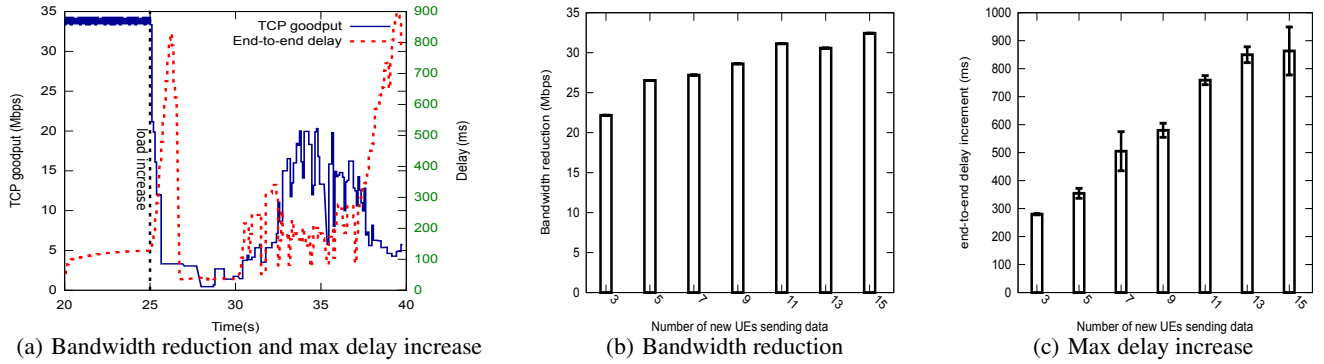
(a) Bandwidth reduction and max delay increase (b) Bandwidth reduction (c) Max delay increase

Figure 2: Load increase results

eNodeBs so that failure of an eNodeB can lead to load increase as UEs switch to neighboring eNodeBs. Second, unrelated to failure, UEs are mobile and can simply congregate at a certain location, thus rapidly increasing the load on a particular eNodeB.

Since the resources on the radio interface to an eNodeB is shared among UEs, the number of UEs simultaneously sending/downloading traffic has a significant impact on the effective bandwidth available to a given UE.

Experimental setup: To study the effect of load increase in a cell, we start each experiment with two UEs downloading data. After a period of time, additional UEs (3 to 15) start downloading data simultaneously. The UEs are placed at a uniformly random distance within a 250 meter range from the eNodeB and follow the Levy-walk mobility pattern as described in Section 2. The radio resource used in the cell is 100 RBs, and the eNodeB uses MIMO (Multiple Input Multiple Output) with transmission diversity (transmission mode 2 [4]) with a maximum bitrate of 75 Mbps.

Metrics: We measured the bandwidth change and the maximum delay increase a UE would experience during the load increase event. Bandwidth change is defined as the difference of the bandwidth 2 seconds before and after the load increase event. Maximum delay increase is defined as the difference of the delay before and the maximum delay shortly after the load increase event.

Results: Figure 2(a) shows TCP goodput and end-to-end delay of a UE in an experiment where 2 UEs were downloading data from the beginning, and at 25s into the simulation, 11 *additional* UEs started downloading data simultaneously. The 11 new UEs compete for the shared radio resource, which results in a significant drop in the available bandwidth for the original UEs. Specifically, once the new UEs start downloading, TCP goodput of one of the initial UE drops from ≈33.5 Mbps to ≈3 Mbps, and the end-to-end delay jumps up by 800% to become around 850ms.

Figure 2(b) shows the magnitude of bandwidth changes when additional UEs were introduced to a cell where there were 2 UEs in the cell initially. The 2 initial UEs experience high bandwidth reductions (consistently larger than 20 Mbps) when new UEs are introduced to the cell. Also, bandwidth throttling resulted in a large number of TCP in-flight segments (of the previous high bandwidth-delay-product link) suddenly building up the queue at the bottleneck link. As shown in Figure 2(c), this sudden increase in queuing delay can be as high as 900ms.

These sudden delay increases could lead to TCP (spurious) time-outs. Figure 3 shows a TCP timeout due to this reason. Before 25s, there were only 2 UEs downloading traffic. At 25s (dashed black vertical line), 11 other UEs started downloading data simultaneously. The original UE's bandwidth was throttled and its segment delay jumped dramatically from around 120 ms to around 850 ms
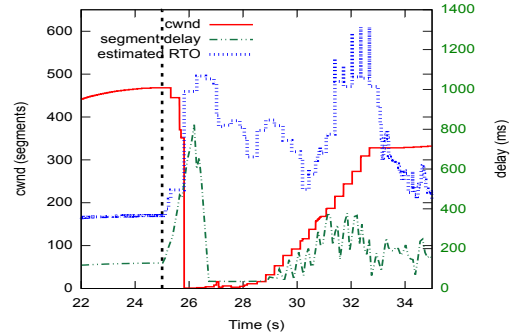

Figure 3: TCP timeout caused by load increase

within 1s. The TCP estimated Retransmission Timeout (RTO) value was computed *before* the delay increase event and was not updated when the delays were changing. As a result, at around 26s, TCP RTO timer expires, the estimated RTO doubles, and TCP resets its congestion window to 1 and enters the slow-start phase.

Implications: Load increase in a cell can significantly throttle the bandwidth available to a UE and thus increase the delay, especially when the eNodeB maintains a large per-UE queue [15]. Such sudden delay increases can invalidate the current TCP estimated RTO value that was calculated before the delay peak and therefore cause unnecessary TCP timeouts, even when *no packets are lost*. TCP timeout causes the congestion window to collapse and the result is link underutilization and unnecessary retransmissions [23, 17]. We plan to experiment with a broader set of scenarios and design an intelligent scheduling mechanism to mitigate the TCP performance degradation. For example, when many new UEs arrive and initiate control-plane signaling (to set up radio bearers), an eNodeB can anticipate additional competing traffic flows and start allocating fewer resource blocks to existing UEs.

3.2 LTE PDCP handover

The data traffic exchanged between UE and eNodeB traverses a LTE Radio Protocol stack consists of four layers: Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), Medium Access (MAC) protocol, and Physical Layer (PHY). The PDCP layer passes packets between the IP layer and the RLC layer and provides security and header compression for traffic that needs it (e.g., VoIP). During a handover, the PDCP layer ensures re-transmissions for traffic that demands it. The RLC layer provides concatenation, segmentation, in-order delivery, and retransmissions (i.e., with RLC Acknowledged Mode). The MAC layer performs multiplexing, demultiplexing and scheduling transmission opportunities amongst
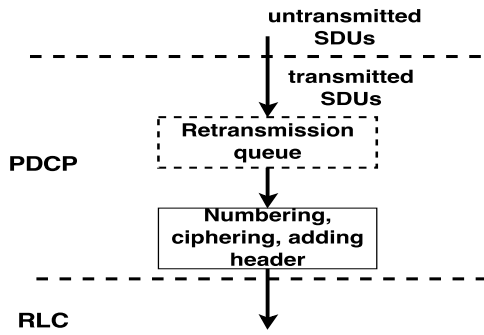
Figure 4: PDCP retransmission queue

UEs (i.e, MAC scheduler). The MAC layer also performs HARQ (Hybrid Automatic Repeat reQuest) function that retransmits lost frames.

During an impending handover event, the *source* eNodeB decides to handover a UE to a neighboring *target* eNodeB based on the signal strength measurement reported from the UE or the current eNodeB load. During a handover, if the UE is receiving data, the source eNodeB can have some data in the queue for the UE. The PDCP queuing is depicted in Figure 4. Depending on how to handle this data, there are two types of handover in LTE.

● Seamless handover: Only untransmitted PDCP Service Data Units (SDUs) (those that did not cross the boundary to enter the PDCP layer in Figure 4) are forwarded to the target eNodeB during handover using X2. (The X2 interface is used for inter-eNodeB communication, including handovers, over a wired network). All data in the PDCP retransmission queue (dashed box in figure 4) are dropped.

● Lossless handover: In addition to the untransmitted PDCP SDUs, all data in the PDCP retransmission queue are forwarded to the target eNodeB.

With seamless handover, the source eNodeB can potentially drop a large number of segments. This can result in data loss at the TCP layer and cause unnecessary TCP back-offs, especially with a large retransmission queue. We investigate these aspects next.

Experimental setup: We simulate handovers due to mobility by letting a UE move between two eNodeBs. We use a cell selection algorithm [3] with 3dB handover margin (minimum signal strength difference between target cell and current cell) and 256ms of 'time-to-trigger' (minimum wait time before triggering a handover when the above condition is met). To simulate macro cell scenarios, the distance between two eNodeBs is set to 750m, and UE's velocity is randomly chosen from a normal distribution with a mean of 40 km/h and variance of 5 km/h. We experiment with varying sizes of the PDCP retransmission queue at the source eNodeB to understand the relationship between the queue-sizes and TCP performance. We use 6 ms as the delay between source and target eNodeBs. We repeat each experiment 20 times.

Metrics: We use TCP goodput, throughput and end-to-end delay as metrics of TCP performance. The metrics are calculated in a period of 5 seconds after a handover to understand the performance impact due to a handover.

Results: *Seamless handover.* Figure 5 shows congestion window, throughput, and packet delay of TCP CUBIC when the retransmission queue is discarded during a handover. At about 31s a handover happened (vertical dashed line). As all SDUs in the transmission buffer were discarded during a seamless handover, many TCP segments were lost. As a result, TCP reduced its congestion window
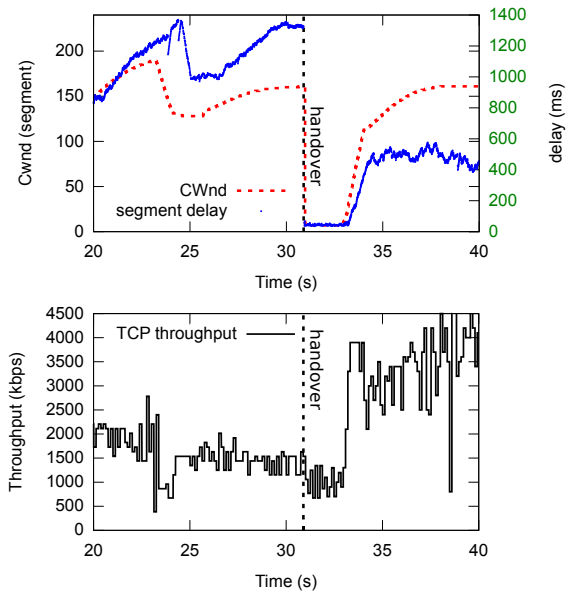

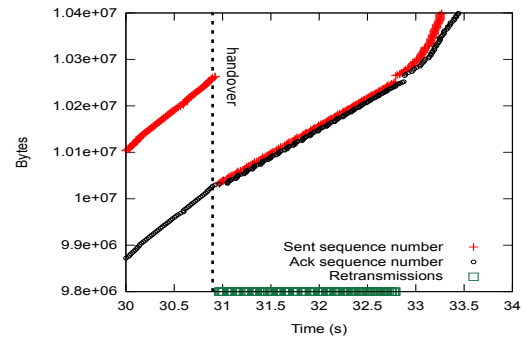**Figure 5: TCP delay and throughput in a seamless handover**

Figure 6: TCP sequence number in a seamless handover

(cwnd) and retransmitted lost segments. This reduced the throughput shown in the bottom sub-figure in Figure 5. Before handover, segment end-to-end delay was high (over 1500ms, as seen in the upper figure) due to the queueing delay at the source eNodeB and the relatively low bandwidth as the radio condition is not favorable when the UE was at the edge of the cell. After handover, segment end-to-end delay dropped significantly to less than 70ms. This is because after handover the segments followed the new path (Server→target eNodeB→UE) which has zero queuing delay.

Figure 6 shows the sent and acknowledged sequence numbers of TCP during the seamless handover. As a large number of TCP segments were discarded at the source eNodeB when the handover happened, the TCP sender retransmitted these segments later on (boxes from 31s to about 32.7s). The number of retransmissions depends on the size of the PDCP retransmission queue. In the experiment depicted in Figure 6, a 256KB PDCP retransmission queue was used and there were 180 retransmissions after handover (TCP segment size is 1500B).

Lossless handover. Figure 7 shows TCP in action during a lossless handover with the same UE mobility scenario and eNodeBs allocation as described above. As all frames in the retransmission queue were forwarded to the target eNodeB, we observed no losses at the TCP layer, and TCP continued sending new segments. Hence, compared to the seamless handover case, TCP throughput is higher
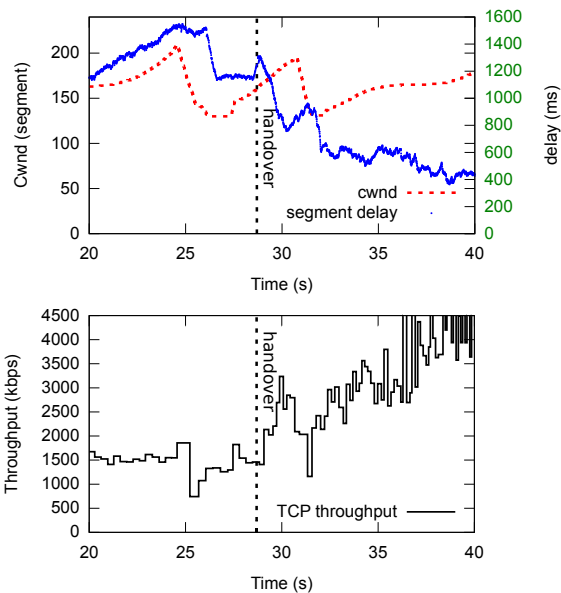
Figure 7: TCP delay and throughput in a lossless handover

Figure 8: TCP goodput during handovers

Figure 9: End-to-end delay during handovers

(bottom sub-figure in Figure 7). Specifically, the average throughput for the 5 second period after the handover is 2421 kbps for the lossless case compared to 1922 kbps for the seamless case. Unlike the seamless handover case, after the lossless handover the end-to-end delay did not drop immediately (as seen in the upper figure). Instead, the delay stayed at a high value since segments were forwarded but not dropped (the delay is 825 ms on average). The delay then gradually reduced because the UE was moving towards the target eNodeB and thus had a better available bandwidth.

End-to-end delay and TCP goodput. Figure 8 shows TCP goodput for two types of handovers for different PDCP retransmission queue sizes. Each experiment scenario was repeated 20 times with a random UE speed (mean 40km/h, variance 5km/h) and the average goodput and standard deviation were shown. The figure shows that in general TCP goodput is higher in lossless handovers. Moreover, when the PDCP retransmission queue size increases, TCP starts performing worse in seamless handovers (the difference is obvious when the PDCP queue size is larger than 256KB). Note that since TCP in seamless handovers recovers faster when the PDCP retransmission queue is small, and since we compute the average goodput 5s after the handovers, the difference in term of goodput between seamless and lossless handovers might not be that obvious when the retransmission queue size is small.

Figure 9 shows the average and standard deviation of end-to-end delay for the two types of handovers for the same set of experiments. We observe that lossless handover experiences high end-to-end delays as we increase the PDCP queue size. This is because lossless handover essentially shifts the queue from the source eNodeB to the target eNodeB, which helps maintain high goodput (Figure 8) but increases the delay as a result. Surprisingly, the larger the retransmission queue the smaller seamless handover end-to-end delay. This is because during the retransmission period in seamless handover (figure 5 after 31s) TCP segment delay is small. Moreover, with a large retransmission queue, seamless handovers cause more losses and a long retransmission period. The longer the period the smaller average segment delay observed.

Implications: Our results illustrate that radio-link handover can cause significant performance variation for ongoing TCP flows. Specifically, seamless handover can cause significant TCP goodput
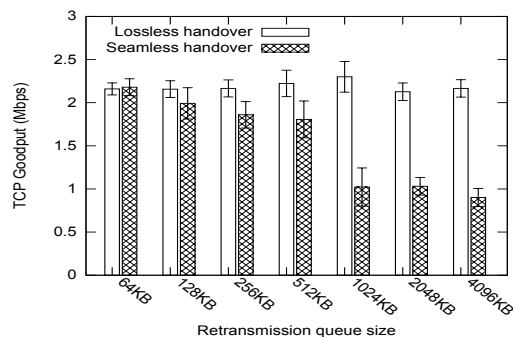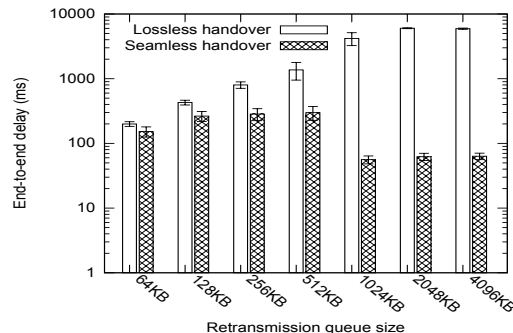
loss; lossless handover can maintain TCP goodput, but TCP segment delay is significantly higher. Another possible implication of sudden delay increase due to lossless handover is TCP (spurious) timeouts because additional delay over the X2-link can cause the total round-trip time to become larger than the current RTT estimate. We plan to continue with our investigation in this aspect of spurious timeouts and how to mitigate the TCP performance implications.

For delay-tolerant applications (e.g., HTTP, FTP) that use TCP as the transport protocol, lossless handover is beneficial in term of goodput. However, delay-intolerant applications such as VoIP may prefer delay over high goodput. Unfortunately, many of these applications (i.e, VoIP, HTTP, FTP) are running in Over-The-Top manner as undifferentiated data applications and they can't tell the network which type of handover is desirable for them. We leave the effect of handovers on the performance of these applications as a future work.

There are other handover scenarios worth exploring. For example, handover could happen regardless of UE mobility, e.g., because of cell balancing, cell failure, etc. In these cases, the UE's radio condition and mobility pattern might be different compared to handovers due to mobility.

4. RELATED WORK

The inherent random and dynamic nature of the wireless and mobile environment poses unique challenges to communication protocols like TCP which were originally designed for static wired environments. Ramjee et al. [7] described the challenges associated with optimizing TCP performance over 3G networks that exhibit significant delay and rate variations. To mitigate the effect of losses in the wireless channel, cellular networks use sophisticated protocols for local re-transmissions etc. However, there is no cross layer interaction of these protocols with the de-facto transport protocol TCP, which perceives wireless losses as indication of congestion.

Erman et al. [8] have shown that the lack of harmony between the TCP layer and the underlying cellular network operation (specifically the different state machines) negatively impacts application performance. Winstein et al. [24] showed how TCP-like reactive congestion control leads to additional delay due to built-in reliability mechanisms in the cellular medium access control layers. Huang et al. [12] did a large scale study on a commercial LTE network and they found various inefficiencies in TCP over LTE such as undesired slow start. Other research works analyzed the impact of radio link control protocols and parameters on the performance of UMTS systems [22, 25]. Researchers have also identified that large delays associated with handovers in cellular environment lead to TCP performance degradation and proposed some optimizations to reduce handover-related delay [18, 5].

However, the majority of existing studies either were based on measurements of higher level application behavior, or looked at different protocol layers in isolation. In this paper, we delved into the intricate details of cross layer interactions between TCP and lower layer protocols in LTE. Specifically, we investigated specific scenarios encountered in a cellular environment such as sudden load increase in a cell and handovers and shown through detailed simulations how the operation of the LTE radio access protocols in these scenarios impact TCP performance.

5. CONCLUSION

In this paper, we took a step towards a systematic study of the interactions of the mobile networks and TCP. In particular, we studied the impacts of sudden load increase and handovers in mobile networks on TCP performance.

There are still many dimensions worth exploring such as: the magnitude of load increase due to different reasons such as disaster or special events; how different applications perform during the two different handover mechanisms; and how UEs and eNodeBs assign most efficient and effective radio protocols parameters to traffic from different types of applications.

6. REFERENCES

[1] Radio Link Control (RLC) Protocol Specification (Release 8), 3GPP TS 36.322. *V8. 0.0, Dec* (2007).

[2] Packet Data Convergence Protocol (PDCP) Specification. Tech. rep., Tech. Rep 3GPP TS 36.323, 2009.

[3] Requirements for support of radio resource management. Tech. rep., Tech. Rep 3GPP TS 36.133, 2011.

[4] Physical layer procedures. Tech. rep., Tech. Rep 3GPP TS 36.213, 2014.

[5] BALAKRISHNAN, H., SESHAN, S., AND KATZ, R. H. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks 1*, 4 (1995), 469–481.

[6] BLAJIĆ, T., NOGULIĆ, D., AND DRUŽIJANIĆ, M. Latency Improvements in 3G Long Term Evolution. *Mipro CTI, svibanj* (2006).

[7] CHAN, M. C., AND RAMJEE, R. TCP/IP performance over 3G wireless links with rate and delay variation. *Wireless Networks 11*, 1-2 (2005), 81–97.

[8] ERMAN, J., GOPALAKRISHNAN, V., JANA, R., AND RAMAKRISHNAN, K. K. Towards a SPDY'Ier Mobile Web? In *Proceedings of the ACM CoNEXT* (2013), pp. 303–314.

[9] HA, S., RHEE, I., AND XU, L. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review 42*, 5 (2008), 64–74.

[10] HENDERSON, T. R., LACAGE, M., RILEY, G. F., DOWELL, C., AND KOPENA, J. Network simulations with the ns-3 simulator. *SIGCOMM demonstration* (2008).

[11] HOLMA, H., AND TOSKALA, A. *LTE for UMTS-OFDMA and SC-FDMA based radio access*. John Wiley & Sons, 2009.

[12] HUANG, J., QIAN, F., GUO, Y., ZHOU, Y., XU, Q., MAO, Z. M., SEN, S., AND SPATSCHECK, O. An in-depth study of LTE: Effect of network protocol and application behavior on performance. In *Proceedings of the ACM SIGCOMM* (2013), ACM, pp. 363–374.

[13] INDEX, C. V. N. Global mobile data traffic forecast update, 2013-2018. *White Paper, February* (2014).

[14] JANSEN, S., AND MCGREGOR, A. Simulation with real world network stacks. In *Simulation Conference, 2005 Proceedings of the Winter* (2005), IEEE, pp. 10–pp.

[15] JIANG, H., LIU, Z., WANG, Y., LEE, K., AND RHEE, I. Understanding bufferbloat in cellular networks. In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design* (2012), ACM, pp. 1–6.

[16] KEVIN FITCHARD. Data now 85% of mobile traffic but 39% of revenue: What gives? `http://tinyurl.com/m3xg469`, March 2012.

[17] LUDWIG, R., AND KATZ, R. H. The Eifel algorithm: making TCP robust against spurious retransmissions. *ACM SIGCOMM Computer Communication Review 30*, 1 (2000), 30–36.

[18] PACIFICO, D., PACIFICO, M., FISCHIONE, C., HJALRMASSON, H., AND JOHANSSON, K. Improving TCP Performance During the Intra LTE Handover. In *GLOBECOM 2009. IEEE* (2009).

[19] PARSONS, J. D., AND PARSONS, P. J. D. *The mobile radio propagation channel*. J. Wiley, 2000.

[20] QIAN, F., SEN, S., AND SPATSCHECK, O. Silent TCP Connection Closure for Cellular Networks. In *Proceedings of ACM CoNEXT* (2013), pp. 211–216.

[21] RHEE, I., SHIN, M., HONG, S., LEE, K., KIM, S. J., AND CHONG, S. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking (TON) 19*, 3 (2011), 630–643.

[22] ROSSI, M., SCARANARI, L., AND ZORZI, M. On the UMTS RLC parameters setting and their impact on higher layers performance. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th* (2003), vol. 3, IEEE, pp. 1827–1832.

[23] SAROLAHTI, P., KOJO, M., AND RAATIKAINEN, K. F-RTO: an enhanced recovery algorithm for TCP retransmission timeouts. *ACM SIGCOMM Computer Communication Review 33*, 2 (2003), 51–63.

[24] WINSTEIN, K., SIVARAMAN, A., AND BALAKRISHNAN, H. Stochastic forecasts achieve high throughput and low delay over cellular networks. In *Proceedings of USENIX NSDI* (2013), pp. 459–472.

[25] XU, H., CHEN, Y.-C., XU, X., GONEN, E., AND LIU, P. Performance analysis on the radio link control protocol of UMTS system. In *Vehicular Technology Conference. 2002 IEEE 56th*, vol. 4, pp. 2026–2030.

[26] XU, Y., WANG, Z., LEONG, W. K., AND LEONG, B. An End-to-End Measurement Study of Modern Cellular Data Networks. In *Proceedings of the 15th Passive and Active Measurement Conference (PAM 2014)* (March 2014).