Measurement Analysis of TCP Congestion Control Algorithms in LTE Uplink

Ali Parichehreh, Stefan Alfredsson, Anna Brunstrom

Department of Mathematics and Computer Science
Karlstad University, Sweden
Email:{firstname.lastname}@kau.se

***Abstract*—The unprecedented growth of user generated contents yielded by the proliferation of social networks applications, cellular based video surveillance and device-to-device (D2D) communication, makes the cellular uplink communication an attractive topic. In this paper we conduct a systematic evaluation and measurement analysis to characterize cellular uplink traffic and compare its interplay with different TCP congestion control algorithms (CCA), namely NewReno, Cubic, and BBR, in both stationary and mobility scenarios. The evaluation encompasses average throughput, average round trip time (RTT), fairness among simultaneous flows, and packet retransmission. The intended behavior of BBR has been observed in LTE uplink, but some severe issues such as lack of fairness among simultaneous flows and massive on device packet losses have been observed. It is observed that the lack of fairness among simultaneous flows can unpredictably change the throughput of multi-flow applications.**

***Index Terms*—Measurement Analysis, Congestion Control Algorithms, Long Term Evolution, Uplink Communications.**

I. INTRODUCTION

Traditionally, mobile devices support only a single application due to the low computational power. The proliferation of more powerful mobile devices enables running multiple applications by leveraging concurrent TCP streams. Furthermore, user generated contents by social network applications as well as the new use cases defined by 5G networks such as machine to machine (M2M) and device to device (D2D) communication, fog based traffic offloading, and HD video chat/surveillance, cause an inevitable uplink traffic surge in future mobile networks. The Third Generation Partnership Project (3GPP) has recently standardized the advanced version of Long Term Evolution (LTE-A) and LTE-A Pro, as a road toward 5G networks, with a peak data rate of up to 1.5 Gbps in uplink [1]. Therefore studying the performance of multiple simultaneous flows with different CCAs in a real LTE-A network that leverages 20MHz bandwidth in uplink (extendable to 100MHz bandwidth when employing carrier aggregation) is crucial, and deserves further investigations.

In the past decade, a large body of work has been produced by researchers to improve the performance of TCP congestion control algorithms (CCA), ranging from loss based CCAs such as [2], [3], [4], [5] to delay based [6], [7], and mixed loss-delay based protocols [8]. In addition, some CCAs such as [9], [10], [11] leverage machine learning algorithms to find the optimal transmission points. BBR (Bottleneck Bandwidth and Round-trip propagation time) [12] is one of the recently introduced *model-based* congestion control algorithms. It has been shown to achieve a superior performance in high throughput and wired communications such as high-speed wide-area networks [12], compared to the widely used congestion control algorithms. The long term plan of the BBR team is to enable it as a default congestion control algorithm for the Internet [13]. However, it is still questionable whether BBR is able to find an equilibrium point between latency and throughput in unpredictable mobile cellular networks [14] that is an indispensable part of the future Internet. Therefore systematic measurement analysis and experiments are required to better understand the performance of BBR CCA in different scenarios and traffic patterns.

Recent studies on the performance of different TCP protocols (including BBR) are conducted in both stationary and mobility scenarios over live LTE networks as well as emulators with LTE link traces, considering different metrics such as throughput, delay, and fairness. The performance of BBR in highway scenarios is investigated in [15] and results indicate that BBR achieves higher throughput compared to Cubic even in low Reference Signal Received Power (RSRP) regime or handover regions. However, in [14] it is observed that BBR is not able to estimate the available bandwidth and utilize LTE link in an emulation environment. Hence a link coupled TCP CCA is proposed that leverages the architectural trends of 5G networks to enable accurate satisfaction of the requirements of each individual application. In other works, BBR shows lack of fairness among simultaneous flows [16] and it can be especially violated in the startup phase where competing short flows with loss-based CCA struggle to get a fair share of the bandwidth [17], [18].

Although the performance of TCP congestion controls in mobile cellular networks have been investigated in downlink direction, there are only few works considering the performance of TCP CCAs in LTE uplink communications. On-device bufferbloat is an important delay factor investigated in [19], and there is no practical widely deployed solution to mitigate its effect. Further investigations revealed that qdisc based solutions are not effective enough in confrontation with on-device bufferbloating, as qdisc strategies have negligible impacts on the firmware queue [19]. In addition, using the scheduling request based access, users must transmit a grant request, wait for a uplink grant, and then wait to use the grant that inevitably imposes additional delay in uplink communication. Considering the mentioned facts above, in this paper we conduct what is, to the best of our knowledge, one of the first systematic evaluations and analysis of different CCAs in the LTE uplink. We leverage a distributed experimentation platform called MONROE [20] for the evaluation. Our target is to reveal the performance of different CCAs in dealing with bulk transmission in the LTE uplink. The results of this study can be used as an inspiration for design and development of new CCAs covering the requirements of 5G scenarios that actively use uplink communications with multiple flows.

This paper is organized by addressing the experiment environment in section II, including the examined TCP CCA variants and the experimentation testbed. In section III our measurement campaign is explained, and the performance analysis of different CCAs is provided in section IV. Finally, some conclusions are presented in section V.

II. Experiment Environment

Our goal in this experimental study is to verify and compare the behavior of some of the major TCP Congestion Control algorithms (CCAs) and explore their potential drawbacks when employed in the LTE uplink. This section first describes the most important features of the selected CCAs and later details our experimentation testbed.

A. TCP variants

TCP NewReno [2] improves retransmission during the fast-recovery phase of traditional Reno and is considered as baseline CCA in our work. It uses an Additive Increase Multiplicative Decrease (AIMD) scheme: upon the reception of an ACK, the cwnd is increased by a constant (1 in slow start, 1/cwnd in congestion avoidance) and after a congestion event - three duplicate ACKs or a timeout - it is respectively decreased by a multiplicative factor or backoff (50%) or set to 1 segment. One of the most important and problematic features of this TCP variant is that it is RTT-synchronized: the cwnd is only increased when an ACK is received, leading to a slow growth for high RTTs.

Cubic [3] uses a similar approach, but instead of an AIMD scheme, it uses a cubic function to govern the cwnd value. The function is concave when cwnd is below the value for which the last loss happened and convex afterwards. This allows for reducing the losses if the congestion level is stable while still probing efficiently for more bandwidth. This behavior is independent of the reception of ACKs, hence CUBIC is not RTT-synchronized. The backoff after a congestion event is multiplicative but around 40% smaller than for NewReno. Considering that this scheme is the default one in the Linux kernel, it is important to include in our study.

BBR [12] is basically a model-based congestion control algorithm that estimates the maximum bottleneck bandwidth and minimum round trip time to calculate the bandwidth delay product and the optimal transmission rate. In contrast to NewReno and Cubic, BBR does not use congestion window or ACK clocking to control the amount of in flight data. In fact it uses a rate based strategy and probes for more bandwidth by increasing its sending rate using a pacing gain, and directly reduces its rate to drain the potentially created queue.

B. Experimental testbed

MONROE [20] is an open platform which allow experiments on Mobile Broadband (MBB) networks in several European countries. Each MONROE node integrates a 1GHz 64 bit quad core processor, 4GB of RAM and a 16GB HDD in a PC Engines APU single-board computer. It runs a stripped down version of Debian Linux, with a Docker environment that allows deploying experiments by selecting the desired nodes, start time, and execution time through a centralized scheduler. The MONROE scheduler automates the container distribution on selected nodes, runs the experiment, and collects the results. Each MONROE node contains two APUs. The first APU has two Sierra Wireless MC7455 miniPCI express LTE modems (connected via USB), while the other has one MC7455 modem and a Wi-Fi card. We used the first APU supporting two Cat 6 LTE modems connected to two MBB operators. Note that the default queuing discipline in MONROE node is *pfifo_fast* that employ three FIFO queues side by side, where packets can be enqueued in any of the three bands based on their type of service bits or assigned priority.

III. Measurement Campaign

Our measurement campaign encompasses both stationary and mobility scenarios with two MONROE nodes in Karlstad city, Sweden. For the stationary scenario we deployed a stationary MONROE node at Karlstad University (connected to the LTE cells with RSRP=-100$\pm$2dBm). For the mobility scenario, highlighted in Figure 1, we conduct our experiments using a mobile MONROE node deployed on an intercity transportation bus (between Karlstad and Filipstad) with 2 hours total driving time covering 100km distance, including urban and rural areas with gas station and intermediate city stops along side the road, with onboard passengers connected to the MBB network via direct LTE links or an onboard Wi-Fi network.

Different from well-controlled experimental environments and even high speed train scenarios (with periodic patterns) [21], experiments on intercity buses intrinsically contain more unpredictable variables such as number of onboard and ground users and bus velocity variations. Therefore, we selected one mobile node on a preselected bus with a pre-scheduled route (we detected some changes in the schedule and route of the bus during the experiment and those sampled data are excluded from our dataset). We conducted experiments at different time of the day (when bus is moving) in the mentioned road. The road is covered by 254 LTE cells, and as highlighted in Figure 1-a, LTE cells are densely deployed inside the cities. Therefore larger RSRP values are reported in urban areas (e.g., Karlstad city with geographical coordinates 59.38, 13.45). As shown in Figure 1-c, the RSRP values vary from -60 dBm to -120 dBm (mainly in rural area wherein the larger cells provide

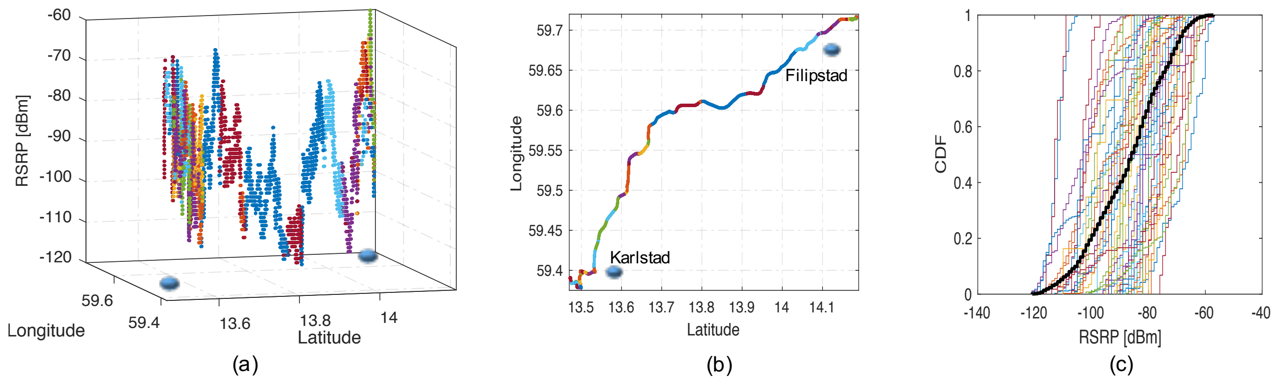

Fig. 1. Measurement campaign environment: a) RSRP values in geographic coordinates, better quality observed at urban areas, b) LTE Cell plan along the road from Karlstad to Filipstad, cells are color-coded, c) Cumulative distribution function (CDF) of each cell's signal strength in terms of RSRP [dBm], CDF of RSRP values of all cells along the road [black line].

coverage for wider areas). We conduct our experiments (from December 2017 to February 2018) using a customized docker container [22] containing RMBT [23] multi-thread traffic flow generator (with configurable CCA) and run experiments with four flows (with the same CCA) at the same time. Experiments with different CCAs are launched consecutively. We collect TCP packet traces (in parallel with socket statistics) of 500 flows for each CCA (1500 flows in total) so that each trace contains 30MB to 100MB payload (depending on the LTE link bandwidth and cell load). In the end, more than 100GB TCP traffic is transmitted in uplink direction.

IV. Comparative Performance Between CCAs

In this study we consider average throughput and average RTT as the main macroscopic metrics to compare the performance of the three investigated CCAs, as throughput and RTT are the main concern for bulk transfer applications (e.g., file transfer or video surveillance) and interactive applications (e.g., Voice over IP, or multiplayer online games). Accordingly we provide a finer analysis on the bytes in flight, fairness among the concurrent flows, and packet retransmissions.

A. Average throughput in stationary and mobility scenario

In this section we investigate the average throughput and RTT at two levels. First we highlight the aggregated throughput of each experiment (aggregated over four concurrent flows) achievable by each CCA, in stationary and mobility scenarios. Then we have a finer analysis on the performance of each flow in terms of throughput and RTT (per flow) using different CCAs.

The aggregated throughput for each experiment (i.e., an RMBT test including four simultaneous flows) is counted by sampling the throughput every five seconds, and the average throughput per flow is counted according to the amount of data transmitted per each flow divided by transmission time in uplink direction. RTT is defined as the time lag between transmitted packet and the received acknowledgment packet from the server, including network propagation and self-inflicted on-device queuing delay.

Figures 2-a and 2-b depict the aggregated throughput (over four concurrent flows) versus average RTT in stationary (left) and mobile (right) scenarios. The slope of the ellipses that cover 95% of the data points highlights the trend of different CCAs. We can see that the throughput in stationary scenario (for the considered stationary MONROE node) is less than the average throughput in mobility scenarios. This is due to the poor channel quality of the stationary node that is around 20 dBm less than the average RSRP in mobility scenarios (see section III). Looking at mobility scenario (Figure 2-b) a skewness for both parameters is observed that implies the variation in throughput and RTT is larger in the mobility scenario. However, BBR is able to maintain the RTT comparatively short, while Cubic experiences longer RTT in some experiments (namely, more than 5% of experiments show RTT longer than 2.5 seconds, when using Cubic CCA). This large RTT is dependent to the variability of the total bytes in flight of concurrent flows as well as the variability of the available bandwidth due to the mobility, path loss and shadowing.

Looking at average throughput and RTT at flow level (see Figure 3) a similar trend is observed. As shown in Figure 3-a, in stationary scenario, Cubic behave similar to NewReno in terms of throughput and utilizing the LTE uplink, but in terms of average RTT it worsens compared to NewReno. In the mobility scenario, as denoted in Figure 3-b, the RTT of Cubic tends to increase more frequently (sometimes twice compared to the stationary scenario). However, BBR is able to detect the variable channel bandwidth, according to the maximum delivery rate and minimum RTT (which here are basically functions of channel capacity [namely path loss, and modulation and coding scheme]) and adapt the transmission rate accordingly. This causes a higher throughput while yielding a minimum impact on the RTT. The Cumulative Distribution Function (CDF) shown in Figure 3-c reveals that for 50% of the flows in the stationary scenario BBR is not able to show a significant improvement in achievable throughput as compared to NewReno and Cubic. But in the mobility scenario (see Figure 3-d) BBR fairly outperforms other CCAs. Finally

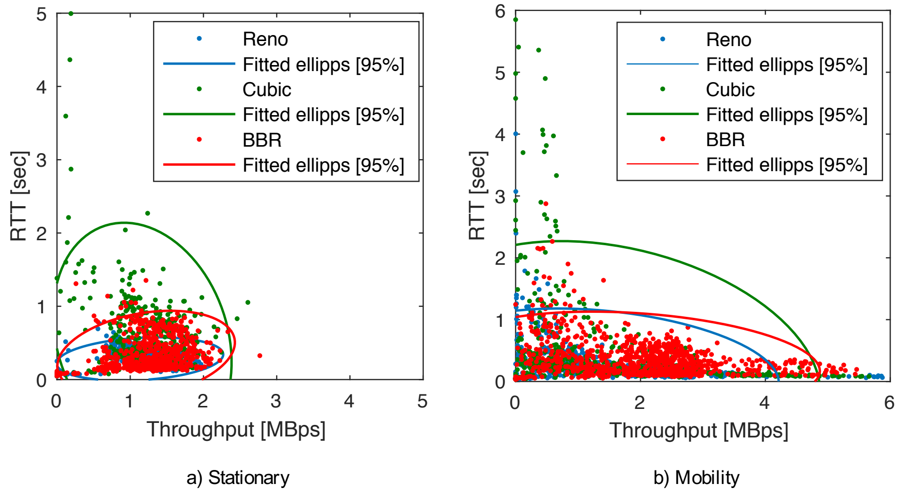

Fig. 2. Aggregated throughput over four flows versus round trip time (RTT) in stationary (a) and mobility (b) scenarios.

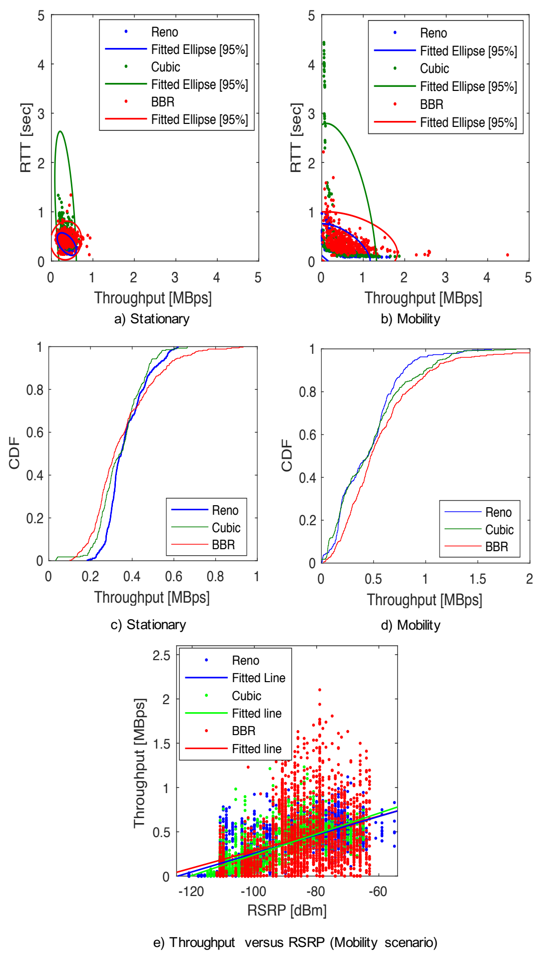

Fig. 3. Average throughput versus round trip time (RTT) in stationary (a) and mobility (b) scenarios. CDF of available throughput per flow (c) and CDF of RTT (d). Average throughput versus RSRP value at bottom (e).

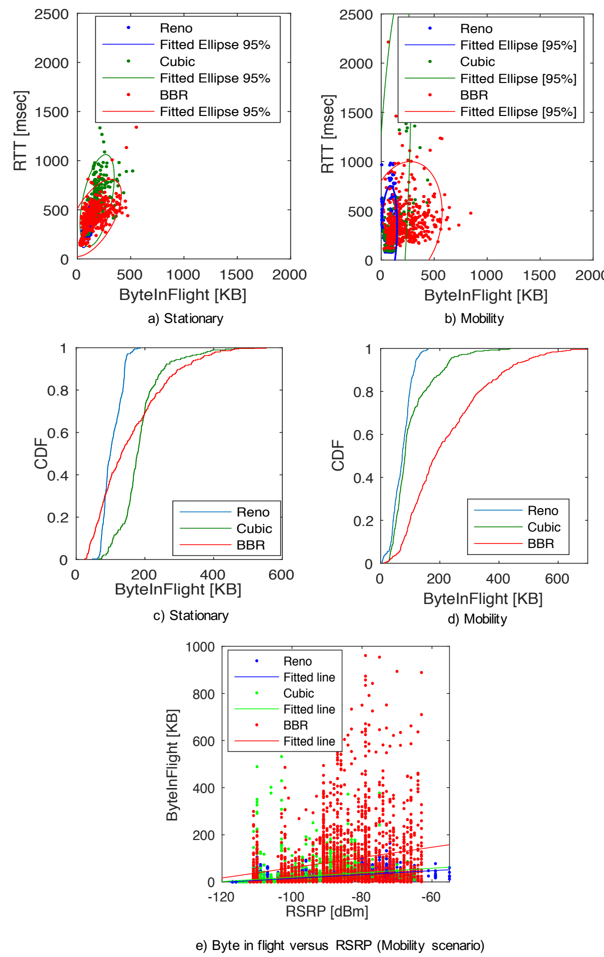

Fig. 4. Byte in flight comparison in stationary (a) and mobility (b) scenarios. CDF of bytes in flight per flow (c) and CDF of RTT (d). Average bytes in flight versus RSRP value at bottom (e).

Figure 3-e represents that when channel quality increases, average throughput fairly increases for all CCAs.

B. Bytes in Flight vs RTT

Figure 4 compare the bytes in flight values versus RTT in stationary (Figure 4-a) and mobile scenario (Figure 4-b) for the three CCA algorithms. Ellipses covering 95% of data points are added to the plots for explanatory reason. It is well-known that Cubic and NewReno detects congestion with losses while Cubic is not RTT-synchronized. Considering the huge buffers used in the modem firmware, Cubic builds a larger on-device queue that yields larger RTT (up to 4 seconds) compared to NewReno and BBR. Although NewReno follows a similar strategy in increasing the cwnd, its RTT-synchronized clock prevents it from quickly ramping up the cwnd. Therefore, in both stationary and mobile scenarios, NewReno does not impose large bytes in flight, and performs with shorter RTT compared to the other CCAs.

However, BBR behaves differently and estimates the bottleneck by feeding the last delivery rate into a max filter and uses a min filter to estimate the RTT over a time window. By leveraging this mechanism especially for bulk transmission (that normally last more than 10 second), BBR is able to maintain the RTT lower than Cubic while efficiently utilizing the LTE uplink with transmitting a larger amount of data. Figures 4-c and 4-d highlight the CDF of the bytes in flight for different CCAs. As is seen when comparing with Figures 3-d and 4-d, there is some correlation between bytes in flight and the average throughput. However, Cubic in stationary scenario shows a larger bytes in flight compared to NewReno, while they result in comparable throughput. The origin of this behavior is verified in section IV-E. The correlation between the amount of bytes in flight and RSRP values is shown in Figure 4-e. Although a large fraction of the bytes in flight are queued in the on-device buffer, and the size of this buffer is independent of the RSRP values, there is a clear correlation between the amount of bytes in flight and RSRP value similar to the correlation between RSRP and throughput (Figure 3-e). However BBR behaves more aggressively in filling the on-device buffer.

C. Throughput vs handover

As we have seen in Figure 1, in the mobility scenario the Monroe node travels across 254 LTE cells and at cell boundaries handover (HO) takes place. According to the LTE radio resource allocation specification [24], the mobile user measures the RSRP values of the neighboring cells and request for the handover to the target cell according to the configuration of the defined mobility-related events. In order to compare the throughput inside the cells and at cell boundaries, we sampled the throughput when the RSRP value of the current cell is above average (between -70dBm to -75dBm). We also sampled the goodput right at the point that handovers take place (within a period of ± 1 seconds). Generally we observe performance degradation at cell boundaries where the HO takes place compared to the time that a user is dwelling inside the cell. This performance degradation is observed in all CCAs shown in Figure 5. However the variability is larger when using BBR CCA compared to Cubic and NewReno (similar to the throughput in Figure 3-b). This larger variability is due to the lack of fairness in BBR that some flows overestimate the available bandwidth (see section IV-D). However, performance degradation at handover region is a function of poor network coverage at cell boundaries as well as the inevitable interruption caused by handover (when breaking the connection from serving cell and synchronizing to the target cell, say 20ms to 40ms link interruption). All the above mentioned facts beside other uncontrolled phenomenon such as the load of the serving/target cells and scheduling policy of the eNodeBs makes it difficult to find a rigorous conclusion on the effect of handover on specific CCAs. We believe a comparison with an in-lab experiment would be required to pinpoint the effect of handover when isolating the other uncontrolled parameters.

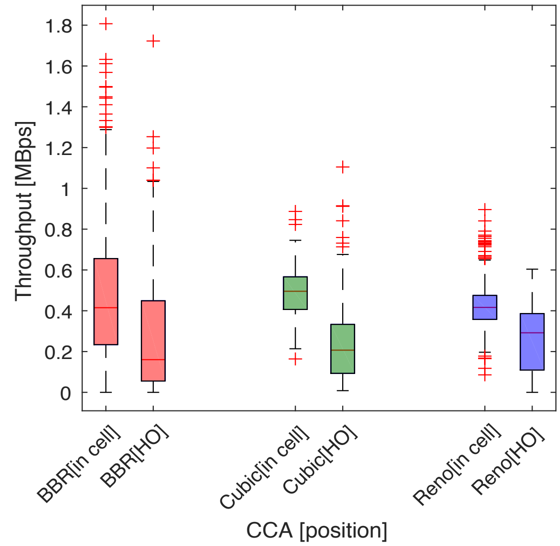

Fig. 5. Comparison of average throughput in cell versus at HO region, BBR (red), Cubic (green), and NewReno (blue).

D. Fairness comparison among multiple concurrent flows

As we mentioned earlier, the proliferation of more powerful mobile devices enables running multiple applications by leveraging concurrent TCP streams simultaneously, although a single application might also use multiple TCP flows at the same time. Therefore it is important to provide a fair share of the network bandwidth to the concurrent TCP flows. In this section, we use Jain's fairness index [25] to verify different CCAs in terms of fairness, that is defined as

$$\text{Fairness}(\bar{X} = [x_1, x_2, ..., x_N]) = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \sum_{i=1}^{N} x_i^2}, \quad (1)$$

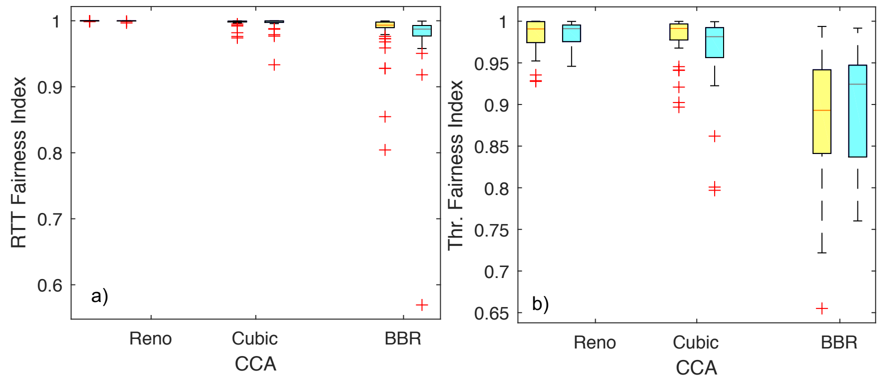

Fig. 6. Fairness among multiple concurrent TCP flows, comparison among different CCA, a) RTT fairness and b) Throughput fairness: Stationary (yellow) and mobile (blue).

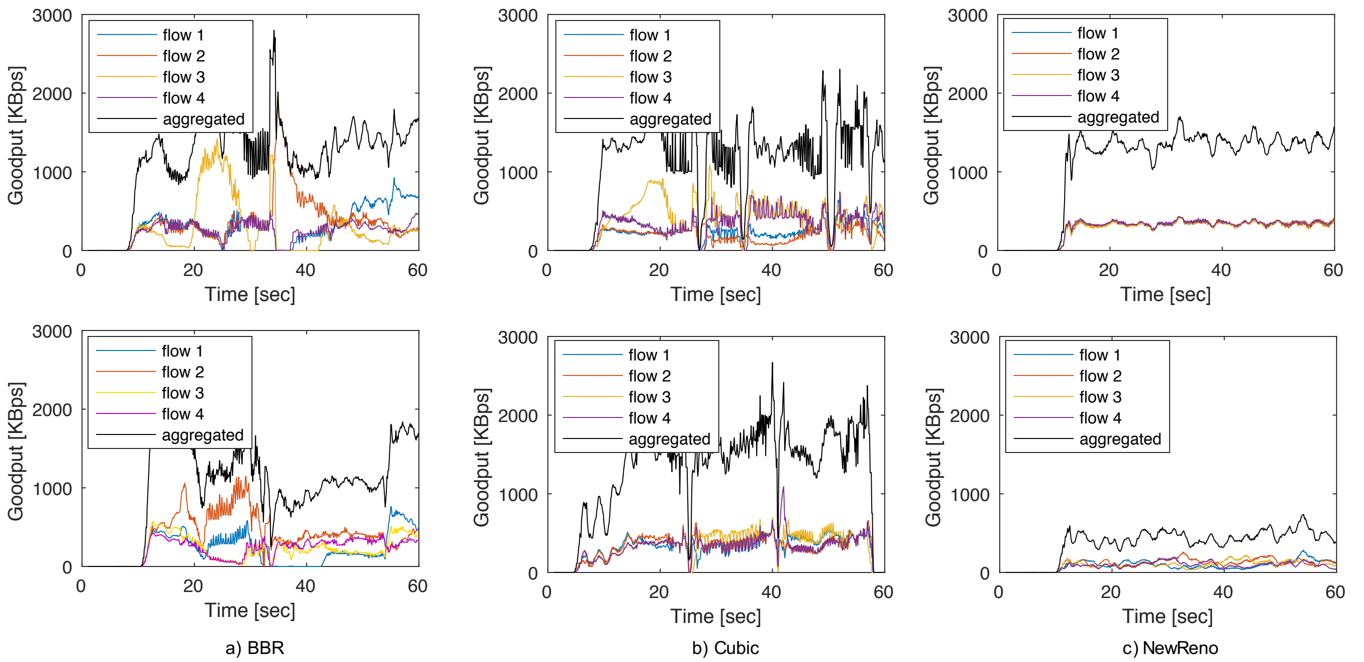

Fig. 7. Detailed view of fairness in terms of achieved goodput for four concurrent flows in stationary (top) and mobile (bottom) scenarios. BBR (a), Cubic (b), NewReno (c).

where x_i is the average RTT/throughput of the i-th flow and N is the number of simultaneous flows. The fairness index varies between zero and one. When the fairness index is close to zero, the CCA is unfair and allocates more portions of the LTE channel bandwidth favorably to some of the flows. If the fairness index is close to one, the CCA is perfectly fair among different flows. We calculate the average RTT and throughput similar to the one calculated in section IV-A.

Figure 6, presents the Jains' fairness index on the average RTT and average throughput in boxplot with the rectangles depicting the standard deviations and the middle line inside the boxes indicating the mean value of fairness among concurrent flows. The whiskers show the best and worst cases of experienced fairness among the flows. Although the behavior of different CCAs on the average RTT parameter is very similar and BBR behaves close to Cubic and NewReno (due to the fact that multiple flows use shared buffer both at qdisc and firmware levels), there is a significant variations in terms of channel utilization and throughput granted to each flow when using BBR.

Figure 7 shows the goodput of each flow in detail for six randomly selected experiments. As is shown, NewReno is able to provide a quite good fairness among the concurrent flows at the expense of low goodput. On the other side, BBR offers larger goodput in favor of some flows while others suffer from channel starvation. In these experiments, some of the flows unpredictably get larger bandwidth share than the other flows that in turn, only gain very small rates for prolonged timespans. This behavior could be observed in all repetitions of experiments. Basically, BBR is based on a model that reflects

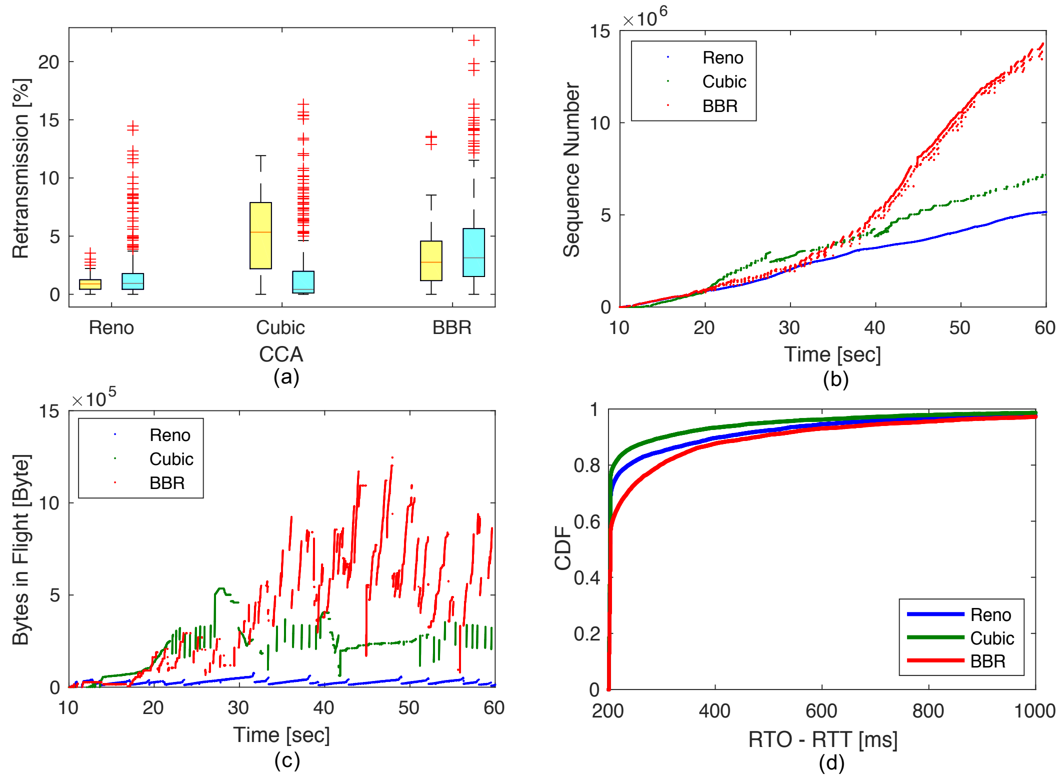

Fig. 8. TCP retransmission in LTE uplink: a) Retransmission percentage in stationary (yellow) and mobility (blue) scenarios, b) Time Sequence plot of randomly selected flows, c) Bytes in flight for different CCA algorithms, d) distribution of difference between RTO and RTT.

the bottleneck behavior for an individual flow (not the behavior of bottleneck for multiple flows and aggregate traffic). Therefore some flows may underestimate the available bandwidth while others overestimate it and increase the total bytes in flight. This accordingly increases the packet retransmission rate (see section IV-E) if the aggregate sending rate is larger than the available bandwidth, and also causes a lack of fairness between concurrent TCP flows. The lack of fairness among the flows can change the aggregated throughput unpredictably as shown in Figure 7-a.

Here we can conclude that the larger variability in throughput of BBR (see Figure 3-b) comes from both the variability in throughput between flows in the same experiment (due to lack of fairness and unpredictable behavior of concurrent flows), and from the variability between experiments (influenced by mobility that is dependent to the path loss, cell load, and large scale fading parameters, e.g., shadowing).

E. Abnormal TCP behavior

In this section we pinpoint an abnormal behavior during bulk transmission in LTE uplink. As shown in Figure 8-a, a huge mass of retransmissions happens in both the stationary (yellow) and mobility (blue) scenarios, especially when using Cubic and BBR. For example, in BBR scenario 4% of the TCP packets are retransmitted (on average). Similarly, TCP Cubic incurs massive retransmissions rather periodically while NewReno illustrates minimum retransmissions. Looking at Figure 8, a significant difference between the amount of bytes in flight can be observed and is the root cause of the massive retransmissions. Due to the RTT-synchronized nature of NewReno, it increases its congestion window size rather slowly and the amount of the bytes in flight is not significant. In contrast to NewReno, Cubic is more aggressive in ramping up with the congestion window size as it is not RTT synchronized. This causes an increased number of bytes in flight and hence the modem buffer drops some of the queued packets. Due to the large amount of bytes in flight, the receiver starts sending a burst of duplicated ACKs (sometimes the amount of the duplicated ACKs is over 100 for a single packet loss). If RTO is not updated upon receiving duplicated ACKs, long latencies (due to the large built up queue) of the retransmitted packets in fast retransmission phase can cause retransmission timeout and trigger the slow start phase. This undesired slow start phase can be observed in the periodic pattern of Cubic in the time sequence diagram as well as bytes in flight (Figures 8-b and 8-c), respectively. On the other hand, BBR simply ignores duplicated ACKs and retransmissions in counting the transmission rate. Therefore the amount of the bytes in flight in BBR is large compared to Cubic and NewReno. Considering the fact that BBR is not able to drain the built-up queue (unless using ProbeRTT mechanism that happens every 10 seconds), this leads to a massive packet loss in the modem buffer. This massive loss of packets and frequent retransmissions can accordingly increase the drain of battery in mobile devices.

Another interesting point to note is that the retransmission rate of Cubic in the stationary scenario is larger compared to the one in mobility scenarios. Basically the stationary channel condition in the stationary scenario allows Cubic to enlarge the congestion window size (looking at Figure 4-c and 4-d highlight that the bytes in flight of Cubic in stationary scenario is larger than other CCAs, as well as Cubic in mobility scenarios) and this rapid ramping up of congestion window size increases the periodic packet drops in the modem buffer.

Figure 8-d highlights the distribution of difference between the RTO (dumped from TCP socket statistics) and the RTT in mobility scenarios. As is shown, sometimes up to a 1 second difference between the RTO and the RTT value exists. In addition, a larger difference is observed when using BBR in mobility scenario. Therefore the RTO estimation algorithm is not able to estimate the RTO precisely (compared to the NewReno and Cubic algorithms) and needs further improvement when using BBR CCA.

V. Conclusion

In this paper, we presented a systematic evaluation and analysis of the performance of three different congestion control algorithms (namely NewReno, Cubic, and BBR) in LTE uplink with multiple concurrent flows. Although it has been shown that BBR can deliver a superior performance compared to the loss-based or delay-based congestion control algorithms in several scenarios, the observed behavior for multiple flows in LTE uplink does not completely meet expected performance metrics. In LTE uplink BBR overload the on-device queue (bottleneck link) with increased in flight TCP packets. Considering the fact that BBR is not able to drain the build-up queue (unless using *ProbeRTT* mechanism that happens every 10 sec), this leads to a massive packet loss. Furthermore, BBR is based on a model that reflects the bottleneck behavior for an individual flow (not the behavior of bottleneck for all flows and traffic) and this causes a biased fairness among the TCP flows in uplink communication. This lack of fairness unpredictably changes the aggregated throughput at application level.

Acknowledgments

This work is partly funded by the EU H2020 research and innovation programme under grant agreement No. 644399 (MONROE).

References

[1] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-advanced Pro and the Road to 5G*. Academic Press, 2016.

[2] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582, Apr. 2012.

[3] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-friendly High-speed TCP Variant," *SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, Jul. 2008. [Online]. Available: http://doi.acm.org/10.1145/1400097.1400105

[4] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*. New York, NY, USA: ACM, 2001, pp. 287–297.

[5] C. Caini and R. Firrincieli, "TCP Hybla: a TCP enhancement for heterogeneous networks," *International Journal of Satellite Communications and Networking*, vol. 22, no. 5, pp. 547–566, 2004.

[6] A. Baiocchi, A. P. Castellani, and F. Vacirca, "YeAH-TCP: Yet Another Highspeed TCP," in *Proceedings of Fifth International Workshop on Fast Long Distance Networks (PFLDnet)*, Los Angeles, USA, 2007.

[7] D. A. Hayes and G. Armitage, "Revisiting TCP Congestion Control Using Delay Gradients," in *NETWORKING 2011*, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 328–341.

[8] S. Liu, T. Başar, and R. Srikant, "TCP-Illinois: A loss- and delay-based congestion control algorithm for high-speed networks," *Performance Evaluation*, vol. 65, pp. 417–440, Jun. 2008.

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

[10] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, "Adaptive congestion control for unpredictable cellular networks," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 509–522.

[11] K. Winstein and H. Balakrishnan, "TCP ex machina: computer-generated congestion control," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 123–134.

[12] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *ACM Queue*, vol. 14, no. 5, p. 50, 2016.

[13] *BBR Congestion Control: An update*. Presentation in ICCRG at IETF 98th meeting, 2017. [Online]. Available: https://www.ietf.org/proceedings/98/slides/slides-98-iccrg-an-update-on-bbr-congestion-control-00.pdf

[14] J. D. Beshay, A. T. Nasrabadi, R. Prakash, and A. Francini, "Link-Coupled TCP for 5G networks," in *IEEE/ACM International Symposium on Quality of Service (IWQoS)*. IEEE, 2017.

[15] L. Feng, C. Jae, Won, and X. Jiang, "Driving TCP Congestion Control Algorithms on Highway," in *The Technical Conference on Linux Networking (NETDEV2.1)*, 2017.

[16] M. Hock, R. Bless, and M. Zitterbart, "Experimental evaluation of BBR congestion control," in *Network Protocols (ICNP)*. IEEE, 2017.

[17] H. Haile, P. Hurtig, K.-J. Grinnemo, A. Brunstrom, E. A. Narbona, F. Liberal, and A. Arvidsson, "Impact of TCP BBR on CUBIC Traffic: A Mixed Workload Evaluation," in *The International Teletraffic Congress (ITC 30)*, 2018.

[18] E. Atxutegi, F. Liberal, H. K. Haile, K.-J. Grinnemo, A. Brunstrom, and A. Arvidsson, "On the use of tcp bbr in cellular networks," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 1–6, 2018.

[19] Y. Guo, F. Qian, Q. A. Chen, Z. M. Mao, and S. Sen, "Understanding on-device bufferbloat for cellular upload," in *Proceedings of the 2016 Internet Measurement Conference*. ACM, 2016, pp. 303–317.

[20] Ö. Alay, A. Lutu, R. García, M. Peón-Quirós, V. Mancuso, T. Hirsch, T. Dely, J. Werme, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, A. S. Khatouni, M. Mellia, M. Marsan, R. Monno, and H. Lonsethagen, "Measuring and assessing mobile broadband networks with MONROE," in *International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2016.

[21] A. Parichehreh, U. Spagnolini, P. Marini, and A. Fontana, "Load-stress test of massive handovers for LTE two-hop architecture in high-speed trains," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 170–177, 2017. [Online]. Available: https://doi.org/10.1109/MCOM.2017.1600773CM

[22] C. Midoglu, L. Wimmer, A. Lutu, O. Alay, and C. Griwodz, "Monroe-nettest: A configurable tool for dissecting speed measurements in mobile broadband networks," in *5th International Workshop on Computer and Networking Experimental Research Using Testbeds (CNERT)*, 2018, pp. 1–6.

[23] L. Wimmer *et al.*, "RMBT: RTR Multithreaded Broadband Test," 2018, accessed: 2018-02-26. [Online]. Available: https://github.com/MONROE-PROJECT/Experiments/tree/master/experiments/nettest/

[24] G. T. 36.331, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Proto-col Specification (Release 13)," 2018. [Online]. Available: http://www.3gpp.org/

[25] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley & Sons, 1991.