Impact of TCP Congestion Control on Bufferbloat in Cellular Networks

Stefan Alfredsson[1], Giacomo Del Giudice[2], Johan Garcia[1], Anna Brunstrom[1], Luca De Cicco[2,3], Saverio Mascolo[2]
[1]*Karlstad University, Sweden,* [2]*Politecnico di Bari, Italy,* [3]*Ecole superieure d'Electricite, France*
{*Stefan.Alfredsson, Johan.Garcia, Anna.Brunstrom*}*@kau.se, dggiacomo@gmail.com, ldecicco@gmail.com, mascolo@poliba.it*

***Abstract*—The existence of excessively large and too filled network buffers, known as bufferbloat, has recently gained attention as a major performance problem for delay-sensitive applications. One important network scenario where bufferbloat may occur is cellular networks. This paper investigates the interaction between TCP congestion control and buffering in cellular networks. Extensive measurements have been performed in commercial 3G, 3.5G and 4G cellular networks, with a mix of long and short TCP flows using the CUBIC, NewReno and Westwood+ congestion control algorithms. The results show that the completion times of short flows increase significantly when concurrent long flow traffic is introduced. This is caused by increased buffer occupancy from the long flows. In addition, for 3G and 3.5G the completion times are shown to depend significantly on the congestion control algorithms used for the background flows, with CUBIC leading to significantly larger completion times.**

I. Introduction

Long queues and additional buffering in the network can be used to increase link utilization and reduce download times. Recently there has, however, been a growing awareness within the networking community that too much buffering may cause problems for delay-sensitive applications. Excessively large and often full buffers, referred to as "bufferbloat", is now recognized as a serious problem in the Internet [1]. Widespread severe over-buffering has also been reported for several parts of the Internet [2], [3], [4], [5].

Bufferbloat results in significantly reduced responsiveness of applications because of excess buffering of packets within the network. It causes both high latency and can also result in appreciable jitter [6]. This is particularly problematic for short TCP flows such as Web traffic or real-time interactive UDP traffic such as VoIP. When such traffic shares resources with greedy TCP transfers it ends up at the end of a full transmission buffer and experiences an increased delay that can severely deteriorate user performance [7].

Cellular networks are becoming an increasingly important Internet access technology. To accommodate varying data rates over time-varying wireless channels they are also normally provisioned with large buffers [8], [9]. The fact that cellular networks typically employ individual buffer space for each user [9], [10] in combination with a low level of user multitasking over cellular connections has in the past limited the impact of these buffers on user performance. However, with the emergence of more and more powerful smartphones, as well as the increasing use of cellular broadband connections for residential Internet access, multitasking over cellular connections is today becoming common. This makes bufferbloat in cellular networks an increasingly important problem. The recent study by Jiang et. al. [5] also confirm that bufferbloat can lead to round trip times (RTTs) in the order of seconds for cellular networks.

The extent of buffer buildup is determined by the rate of incoming packets versus the rate of outgoing packets. Standard TCP congestion control probes the available bandwidth by injecting packets into the network until there is packet loss, which for tail-drop queuing happens when buffers are full. The way buffers fill up are thus highly dependent on the transport protocol behavior and varies between different TCP congestion control algorithms. For example, TCP CUBIC [11] aggressively probes for the available bandwidth leading to a high average buffer utilization, whereas TCP Westwood+ [12] clears the buffers when congestion episodes occur leading to, on average, a reduced buffer occupancy.

In this paper we examine the interaction between the TCP congestion control algorithms used and bufferbloat in 3G/4G cellular networks. Three congestion control algorithms are considered: TCP NewReno, TCP CUBIC and TCP Westwood+. We present an extensive measurement study performed within the 3G (UMTS), 3.5G (HSPA+) and 4G (LTE) networks of one of the leading commercial providers in Sweden, involving more than 1800 individual measurements. In our measurements we study how the response time of a Web transfer is affected by varying levels of competing background traffic and how the congestion control algorithms used affect performance.

Our results indicate that the 3G and 3.5G networks suffer from severe bufferbloat. When background traffic is introduced the Web response time sometimes increases by more than 500%. Furthermore, the congestion control algorithm used for the background flow has a significant impact on the Web response time. The more aggressive congestion control used by CUBIC roughly doubles the Web response time as compared to Westwood+. For low bandwidths (i.e. 3G) the congestion control version used by the Web flow also has a significant impact on performance. In the studied 4G network, bufferbloat is less of a problem.

The remainder of the paper is organized as follows. Further background on the congestion control algorithms used as well as details on what sets our work apart from related measurement studies are described in Section II.

978-1-4673-5828-6/13/$31.00 ©2013 IEEE

Authorized licensed use limited to: Politecnico di Bari. Downloaded on May 12,2026 at 20:59:51 UTC from IEEE Xplore. Restrictions apply.

Section III introduces the measurement setup used and the experimental design. Section IV details our experimental results. Finally, Section V concludes the paper.

II. Background

In this section we introduce the congestion control algorithms evaluated in our measurements and present related work in the area.

A. TCP Congestion Control Algorithms

1) TCP NewReno : The TCP congestion control [13] consists of a *probing phase* and a *decreasing phase*, the well-known Additive Increase and Multiplicative Decrease (AIMD) phases. The TCP congestion control employs two variables to implement the AIMD paradigm: *cwnd* is used to bound the number of outstanding packets, whereas *ssthresh* is a threshold that determines two different laws for increasing the *cwnd*. During the *slow-start phase* the *cwnd* is increased by one packet every ACK reception to quickly probe for extra available bandwidth until *cwnd* reaches *ssthresh*. When $cwnd \geq ssthresh$ the *congestion avoidance* phase is activated and *cwnd* is increased by $1/cwnd$ packets on ACK reception.

The probing phase lasts until a congestion episode is detected by TCP in the form of 3 duplicate acknowledgments (3DUPACK) or timeout events. Following a 3DUPACK episode, TCP NewReno [14] triggers the multiplicative decrease phase and the *cwnd* is halved, whereas when a timeout occurs *cwnd* is set to one segment. The algorithm can be generalized as follows:

1) On ACK: $cwnd \leftarrow cwnd + a$
2) On 3DUPACK:

$$cwnd \leftarrow b \cdot cwnd \quad (1)$$

$$ssthresh \leftarrow cwnd \quad (2)$$

3) On timeout: $cwnd \leftarrow 1;\ ssthresh \leftarrow b \cdot cwnd$

In the case of TCP NewReno a is equal to 1, when in slow-start phase, or to $1/cwnd$ when in congestion avoidance, and b is equal to 0.5.

NewReno is included in the measurements since it is a standardized algorithm, commonly used as a baseline for comparisons with other TCP congestion control algorithms.

2) TCP Westwood+ : TCP Westwood+ [12] is a sender-side modification of TCP NewReno that employs an estimate of the available bandwidth BWE, obtained by counting and averaging the stream of returning ACKs, to properly reduce the congestion window when congestion occurs. In particular, when a 3DUPACK event occurs, TCP Westwood+ sets the *cwnd* equal to the available bandwidth BWE times the minimum measured round trip time RTT_{min}, which is equivalent to set $b = BWE \cdot RTT_{min}/cwnd$ in (1). When a timeout occurs, $ssthresh$ is set to $BWE \cdot RTT_{min}$ and $cwnd$ is set equal to one segment.

The unique feature of TCP Westwood+ is that the setting of *cwnd* in response to congestion is able to clear out the bottleneck queue, thus increasing statistical multiplexing and fairness [12], [15].

The TCP Westwood+ algorithm is included in the measurements as a representative of a congestion control algorithm that uses delay-based information in addition to loss.

3) TCP CUBIC: TCP CUBIC [11] simplifies the dynamics of the congestion window employed by TCP BIC [16] and improves its TCP-friendliness and RTT-fairness. During the probing phase, the congestion window is set according to the following equation:

$$cwnd \leftarrow C \cdot (t - K)^3 + max_win \quad (3)$$

where C is a scaling factor, t is the time elapsed since the last $cwnd$ reduction, max_win is the $cwnd$ reached before the last window reduction, and K is equal to $\sqrt[3]{max_win \cdot b/C}$, where b is the multiplicative factor employed in the decreasing phase triggered by a loss event.

According to (3), after a reduction the congestion window grows up very fast, but it slows down as it gets closer to max_win. At this point, the window increment is almost zero. After that, $cwnd$ again starts to grow fast until a new loss event occurs.

The TCP CUBIC algorithm is included in the measurements since it is the default congestion control in Linux, and is widely deployed on Web servers. It is also representative of a congestion control algorithm that scales well in networks with high bandwidth-delay products.

B. Performance Measurements in Cellular Networks

Although many measurement studies on TCP over cellular networks have been performed, typically they do not focus on bufferbloat and its impact on delay-sensitive traffic. At the best of our knowledge, the recent work by Jiang et. al. [5] provides the first extensive study of bufferbloat in cellular networks and its effect on TCP performance. Measurements from four U.S. carriers and a carrier in Korea are provided showing that a large amount of buffering is being used in all the five cases. In some cases TCP RTTs of up to 10 seconds are observed. The paper further proposes a receiver-side modification that dynamically adjusts the TCP receiver window based on the path characteristics in order to limit the amount of outstanding data and hence the data buffered in the network. Although the importance of the congestion control algorithm used is noticed, it is not the focus of the work in [5]. Our work thus complements [5] by providing a detailed study of the impact of the TCP congestion control algorithm on bufferbloat as well as by providing additional measurements from a leading Swedish carrier.

The performance of different TCP variants over a CDMA 1xEV-DO system is evaluated by Liu et. al. [9]. The authors evaluate the channel characteristics of the 1xEV-DO system and the impact of varying channel data rates on TCP Reno, TCP CUBIC, TCP Westwood, and TCP Vegas. Reno, CUBIC and Westwood show similar throughput performance and are due to large buffers in the network able to obtain a good utilization of the wireless

Authorized licensed use limited to: Politecnico di Bari. Downloaded on May 12,2026 at 20:59:51 UTC from IEEE Xplore. Restrictions apply.

channel. The buffering used is shown to increase the delay for concurrent traffic, especially in the case of CUBIC. Vegas, on the other hand, is able to keep smaller delays at the expense of a reduced throughput. Although for a different cellular technology and not framed in the context of bufferbloat, the results from Liu et. al. support the trends observed in our work. In [17], De Cicco et. al. compare the performance of TCP NewReno, TCP BIC, TCP CUBIC and TCP Westwood+ over a HSDPA network. All congestion control algorithms achieve similar throughput, whereas CUBIC and BIC are observed to exhibit larger RTTs and a larger number of retransmissions. The paper does not consider bufferbloat and the impact on delay-sensitive traffic, but it illustrates the underlying difference between the congestion control algorithms.

In [7], Järvinen et. al. present an investigation of how TCP impacts interactive real-time communication. Based on measurements in a HSPA network, the performance of a constant bitrate voice flow is studied in the presence of concurrent TCP bulk transfer as well as concurrent Web browsing. The queueing delay and resulting increase in latency caused by the TCP bulk transfer is found to make interactive voice communication impossible. Also the impact of a small number of parallel Web object downloads can cause considerable performance issues for interactive voice flows. TCP performance and the impact of different congestion control algorithms is not considered in the paper.

Bufferbloat has also been studied in other types of networks. In [3] it has been shown that widespread severe over-buffering is present in the broadband edge, leading to latencies of up to 4 seconds. Other studies have shown upstream delays on DSL and cable access links in the order of seconds [2], [4].

III. Measurement setup

To examine the impact of the congestion control on bufferbloat in cellular networks, we have carried out a number of measurements on short flows, long flows, and combinations thereof for three network types. These are aimed to represent a usage where multiple flows of different lengths can be active simultaneously, as can be expected in use cases related to cellular-based residential Internet access, and when smartphones simultaneously access both background data and the Web or other interactive services.

A. Metrics

Web browsing is an important application in many use cases. Web page response time is thus a key factor, and in this paper it is used as a primary performance metric. The Web page size obviously affects the response time. In [18] the average data transfer per Web page is reported to be 320 kB, which is typically split into several different host connections. For short transfers, the experiments in this paper use 320 kB of data, sent in a single flow. Such a 320 kB transfer is representative of the base HTML page of popular news sites such as Dagens Nyheter in Sweden or Huffington Post in the US. It also allows to study the impact of the congestion control used by the short-lived Web flow and its interaction with background traffic. In addition to completion time, metrics such as throughput, packet loss ratio (PLR), and round-trip time (RTT) are also reported when appropriate.

B. Measurement collection

The experimental campaign was carried out over three different cellular network technologies (3G, 3.5G, 4G) all provided by the same operator in Sweden. To perform the experiments two computers were used, one laptop with a Huawei E392 USB modem and one server with a connection to the Swedish University backbone. Both computers were running Ubuntu 12.04 with kernel versions 3.1.4 on the laptop and 3.2.0-27 on the server.

Data was collected using tcpdump and later processed with tcptrace to extract flow related information and the relevant metrics. To allow tcpdump to correctly capture traffic, TCP segment offloading (TSO) and related functionality was turned off. Additionally, the TCP metric cache was always flushed before establishing a new TCP connection. Other TCP related variables were left at their default values. Thus the initial window size was 10 segments, and SACK was enabled. Hence, although we refer to NewReno as a base-line congestion control in our measurements, in practice SACK-based recovery was used. Iperf was used to generate the traffic for the long-lived background flows, whereas for the short-lived flows an Apache Web server was used along with the wget utility at the client-side to emulate Web browsing. Outlier rejection was employed to remove data points lying more than an order of magnitude from the median, i.e. when in a 3.5G measurement run values are reported which are consistent with a 3G network connection rather than a 3.5G. To further identify outliers, the IQR (Inter-Quartile Range) was used with a value of 9, which is considerably more conservative than the value 1.5 that is often used.

C. Measurement configuration

A wide range of measurements were made based on the run types illustrated in Figure 1. Runs A and B collect baseline information by measuring transfer characteristics for long and short lived flows without any competing traffic. Run A uses long-lived flows to collect data for 3 minutes for each of the three considered congestion control algorithms, and with three replications each. Similarly, run B collects data for the 320 kbyte short flows, but with 10 replications. In contrast to runs A and B, run C involves both a long-lived background flow and short-lived flows. Finally, run D differs from run C in that five concurrent long flows are now used to generate the background traffic instead of a single flow. Thirteen measurement runs were concatenated into a single measurement campaign with the following run composition: ABCDABCDABCDA, further increasing the number of replications. One such campaign was performed for each of the three different network

Authorized licensed use limited to: Politecnico di Bari. Downloaded on May 12,2026 at 20:59:51 UTC from IEEE Xplore. Restrictions apply.

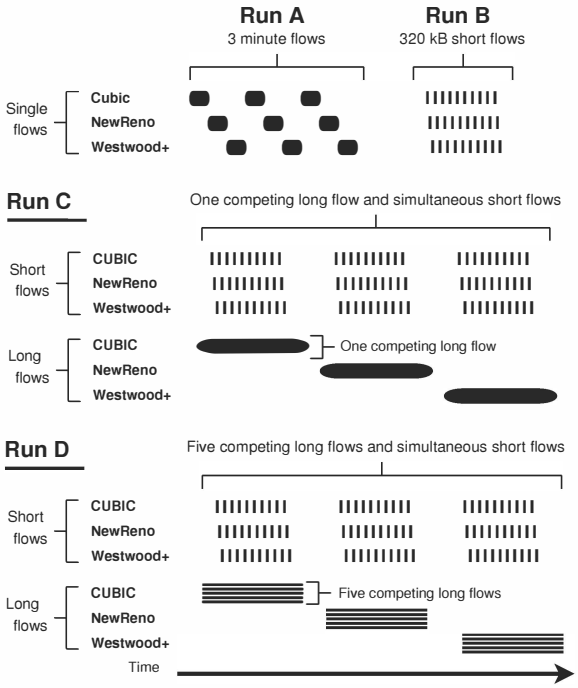


Figure 1. Run types used in the measurements.

technologies. Thus, more than 1800 short flow measurements were collected in these campaigns. A number of additional measurements were also conducted outside of the campaigns, showing similar behavior.

IV. Results

In this section the results from our measurements are presented. Among the mobile broadband access technologies 3G, 3.5G and 4G, the currently most popular in Sweden (where the measurements were performed) is 3.5G. Mobile broadband routers and USB modems are quickly replacing the last mile access, traditionally dominated by ADSL. We therefore choose to focus our discussion on 3.5G, providing first an overview of the results followed by a more detailed analysis. The results for the 3G and 4G measurements are then summarized at the end of this section.

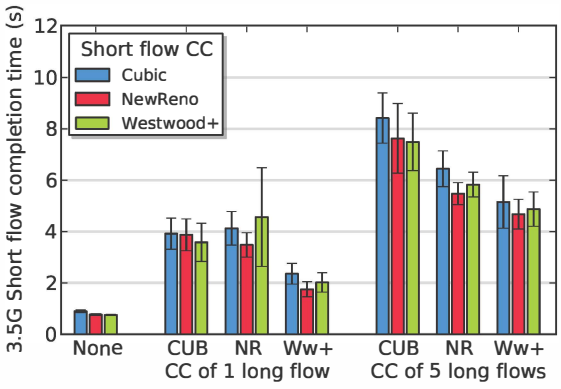


Figure 2. The average Web response times of short flows for different background loads, over 3.5G.

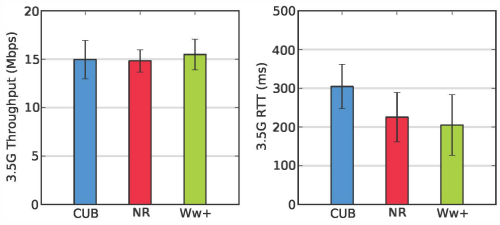


Figure 3. The average throughput and round-trip times of single long flows that use CUBIC, NewReno and Westwood+, over 3.5G.

A. Overview of 3.5G results

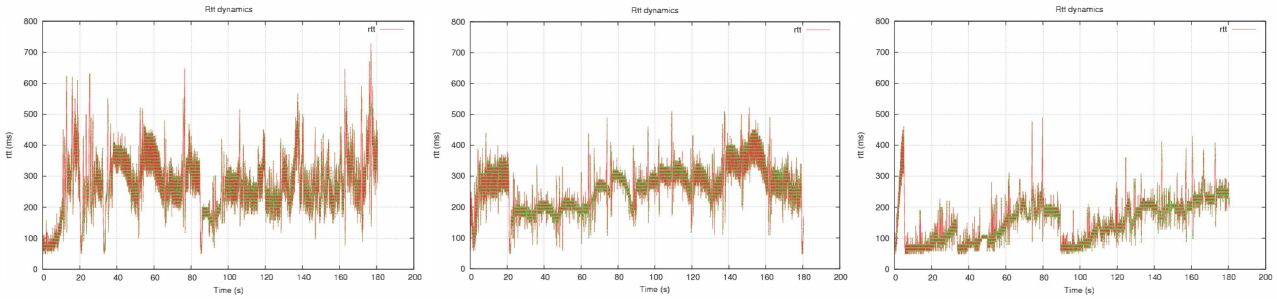
Figure 2 shows an overview of the results from measurement runs B, C, and D. The figure shows the average Web response time and the 95% confidence interval of 30 repeated short flows for each configuration. Shorter time is better. Leftmost in the figure, the first three bars show the average Web response times without concurrent traffic. Second, the middle group of nine bars shows the Web response times of the short flows with one background flow, and all combinations of the three congestion control algorithms. Third, the right group of nine bars also shows the Web response times but for five background flows, for all combinations of congestion controls.

With no concurrent flows the average Web response times are around 0.8 seconds for all three congestion controls. When one background flow is introduced (shown in the second group), the Web response time increases five-fold when CUBIC or NewReno is used for the background flow. When Westwood+ is used, the increase in Web response time is smaller, with roughly a doubling of the response time. The choice of congestion control for the short flow is, however, of limited importance.

When five background flows are introduced, this increases the Web response time of the short flows even further; to about 8 seconds with CUBIC controlling the background flows, about 6 seconds for NewReno and about 5 seconds for Westwood+. Similarly to the results for one background flow, the congestion control of the short flow is of limited importance.

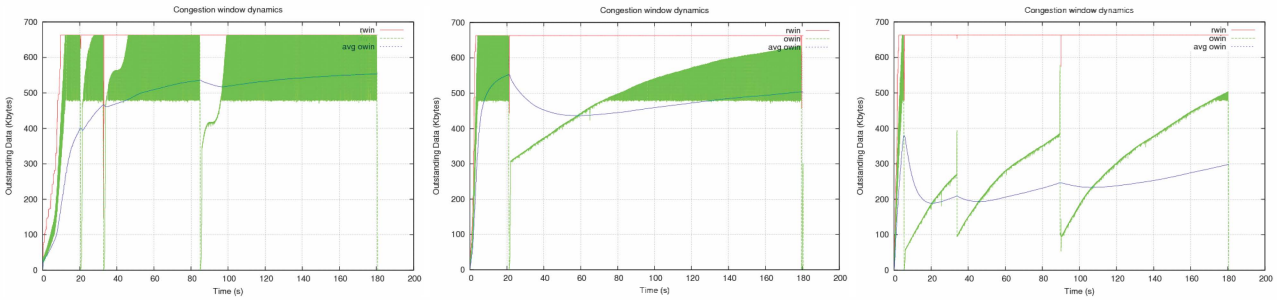
These graphs illustrate that the 3.5G network in our measurements is indeed prone to excessive buffering. The introduction of more background flows has a severe impact on the short flows and results in higher Web response times. The traceroute based method from [5], has been used to verify that the observed queueing delay does indeed stem from buffering in the cellular access network and not from buffering in other parts of the end-to-end path. Additionally, the congestion control algorithm used for the background flow clearly affects the Web response time, where a CUBIC background flow results in significantly higher Web response times compared to a Westwood+ background flow. Varying the congestion control for the short flow does, however, not result in any significant differences in response time.

Authorized licensed use limited to: Politecnico di Bari. Downloaded on May 12,2026 at 20:59:51 UTC from IEEE Xplore. Restrictions apply.



(a) CUBIC (b) NewReno (c) Westwood+

Figure 4. The round-trip time of single long CUBIC, NewReno and Westwood+ flows, over 3.5G.



(a) CUBIC (b) NewReno (c) Westwood+

Figure 5. The amount of outstanding data of single long CUBIC, NewReno and Westwood+ flows, over 3.5G.

B. Analysis of 3.5G results

To further analyze the underlying cause for the difference in Web response times we examine the characteristics of long flows for the different congestion control algorithms. Figure 3 shows the throughput (goodput) and the round-trip times of only long flows (Run A). As shown in the figure, there is no significant difference in throughput between the three congestion control algorithms. However the corresponding average round-trip times differ considerably: the average round-trip time for CUBIC long flows in this scenario is 1.5 times longer than that of Westwood+ long flows.

To illustrate this in more detail, Figures 4(a) – 4(c) compare the round-trip time dynamics for the duration of a select CUBIC, NewReno and Westwood+ flow. While there is a lot of variation in the round-trip times, the difference in round-trip times between different congestion controls from Figure 3 is also visible here. The explanation can be found in the graphs showing the outstanding windows for the respective flows, Figure 5(a) – 5(c). Comparing these, the average outstanding window of the CUBIC flow is roughly twice as large as for the Westwood+ flow after the TCP startup phase.

The throughput of the three flows studied in Figures 4 and 5 is measured at 16.1 Mbit/s for the CUBIC flow, 15.5 Mbit/s for the NewReno flow and 15.7 Mbit/s for the Westwood+ flow, i.e. the throughput is almost the same. This means that the available capacity is utilized in a similar way, and the additional outstanding data in the case of CUBIC is stored in buffers in the network.

Even though there is little difference in throughput, the buffered outstanding data and the increased round-trip

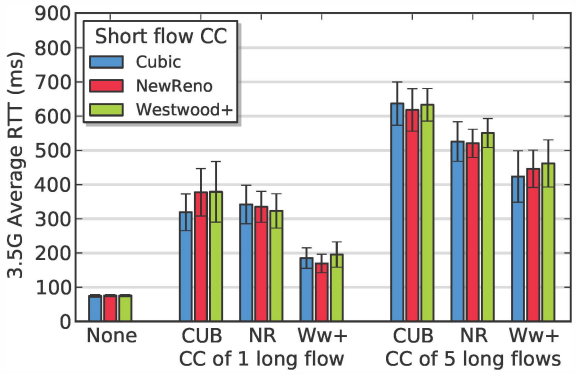


Figure 6. The average round-trip times of short flows for different background loads, over 3.5G.

times in the case of CUBIC impact concurrent short flows negatively. Figure 6 shows the round-trip times of short flows, corresponding to the Web response times shown in the overview in Figure 2. Without any background flow, the average round-trip time of a short flow is about 80 ms. When background flows are present, the round-trip times are increased, in the same magnitude as the Web response times. The way data is buffered in the network by the background flows thus affects the round-trip times, and the difference in Web response times follows as a direct consequence.

This is further evident by studying the round-trip time dynamics of individual short flows. Figure 7 shows the round-trip times of one single short flow without background traffic (Run B of the measurement setup), and three

Authorized licensed use limited to: Politecnico di Bari. Downloaded on May 12,2026 at 20:59:51 UTC from IEEE Xplore. Restrictions apply.

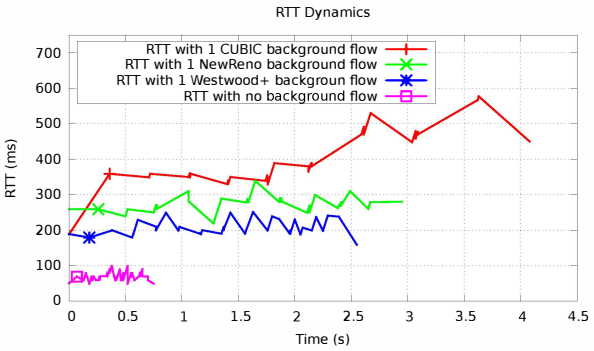


Figure 7. The round-trip time of a CUBIC short flow with no background flows, and with one Westwood+, NewReno and CUBIC background flow, over 3.5G.

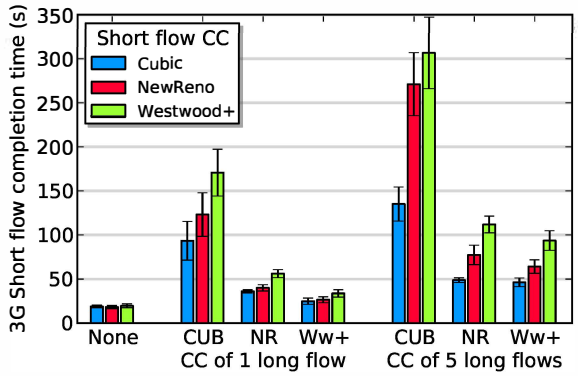


Figure 8. The average Web response times of short flows for different background loads, over 3G.

single short flows competing with a CUBIC, NewReno and Westwood+ background flow, respectively (Run C). Here the difference in round-trip times is clearly visible, as well as the difference in completion times. Each of the four flows transmit the same amount of data, but exhibit varying completion times. The completion times correlate with the RTTs, which in turn were shown previously to depend on the amount buffering caused by any background traffic.

C. Comparison to 3G and 4G

The same measurements as described above were also performed for 3G and 4G, which show some interesting differences. Figure 8 shows the response time overview for 3G. It exhibits the same properties as for 3.5G in Figure 2, although the negative interaction of CUBIC background traffic is more pronounced here. We also observe that the response time, on an absolute scale, is unusable for interactive surfing. With no competing traffic, transferring 320 kbytes of data takes around 19 seconds on average (the three leftmost bars).

One notable difference in Figure 8, as compared to the 3.5G results in Figure 2, is that the congestion control algorithm of the short Web flow now also has a significant impact on the Web response time. When a CUBIC short Web flow competes with five CUBIC background flows it gets an average response time of under 150 seconds. This

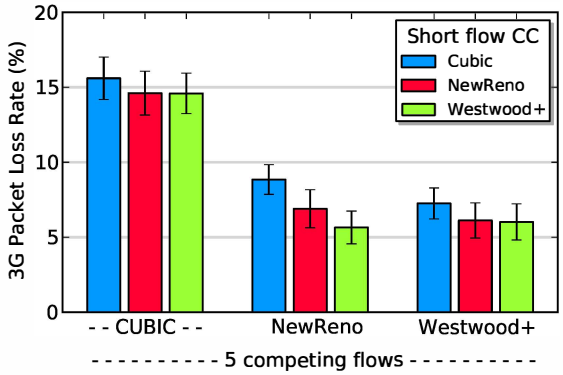


Figure 9. Packet loss rates encountered by short flows when competing with 5 background flows over 3G.

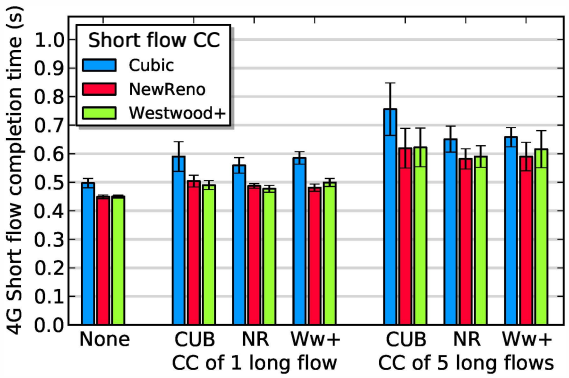


Figure 10. The average Web response times of short flows for different background loads, over 4G.

can be compared to the Westwood+ short Web flow, which gets an average response time of just over 300 seconds when competing with five CUBIC background flows.

The reason for the difference between the two environments is that at the lower 3G bandwidth the presence of heavy background traffic, and in particular CUBIC background traffic, leads to not only increased delays but also to non-negligible amounts of packet loss for the short flows. This is illustrated in Figure 9, which displays the average packet loss rate of short flows when competing with five background flows in the 3G environment. As seen in the figure, the short flows experience around 15% packet loss when competing with the CUBIC long flows. When packet loss occurs the congestion control of the short flow also becomes important and the less aggressive congestion control of Westwood+ leads to the longer response times seen in Figure 8.

The response time overview for 4G is shown in Figure 10. Here, there is only a minor impact when one or five flows are introduced. With regards to the different congestion control algorithms of the background flows, the confidence intervals are overlapping indicating that there is no significant difference. The reasons for this are believed to be due to improved network buffer management and increased bandwidth, but this is subject to further study.

Authorized licensed use limited to: Politecnico di Bari. Downloaded on May 12,2026 at 20:59:51 UTC from IEEE Xplore. Restrictions apply.

V. Conclusions

The rise in cellular-based residential Internet access as well as the increased usage and evolved applications of smart phones are driving demand for cellular Internet connectivity and capacity. In these use cases applications commonly use background data transfer, or longer running non-interactive transfers are mixed with shorter user-interactive transfers. In this context the amount of buffering in the network becomes relevant, and especially with regards to bufferbloat, i.e. excessively large and filled buffers.

We examined the presence and impact of bufferbloat in cellular networks by performing a large-scale measurement campaign using the 3G, 3.5G and 4G networks of a major Swedish cellular network provider. The results showed that the impact of the used congestion control, i.e. NewReno, Westwood+ and CUBIC, was not a major factor when there was only a single flow utilizing the link. In contrast, when shorter Web flows were mixed with longer running background flows the congestion control of the background flows was shown to have a major impact in the 3G and 3.5G environments. For the buffer management used in the studied 3G and 3.5G networks, the efficiency of CUBIC in filling the network pipe did not interact well with competing short flows. When the background flows used CUBIC the penalty to a competing short Web flow was roughly doubled as compared to Westwood+. While the congestion control of the background flows dominated performance, the congestion control of the short flows also had a noticeable impact on performance at the lower 3G bandwidth, with CUBIC giving the best performance.

These results would seem to suggest that in order to optimize the user perceived responsiveness, servers providing background data to cellular devices should use Westwood+ or other less aggressive congestion control algorithms while Web servers and similar can use CUBIC as congestion control. However, before providing any recommendations more measurements using a wider range of operators and networks are necessary. For future work we plan to perform measurement campaigns in additional networks as well as examine other congestion controls such as TCP Vegas [19] and LEDBAT [20].

Acknowledgment

This work was partly funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700) and partly funded by .SE Internetfonden. The views expressed are solely those of the authors.

References

[1] "Bufferbloat Home Page," URL: http://www.bufferbloat.net/ (Apr 2013).

[2] M. Dischinger, A. Haeberlen, K. Gummadi, and S. Saroiu, "Characterizing Residential Broadband Networks," in *Proc. ACM IMC*, 2007.

[3] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the Edge Network," in *Proc. ACM IMC*, 2010.

[4] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, "Broadband Internet Performance: A View From the Gateway," in *Proc. ACM SIGCOMM*, 2011.

[5] H. Jiang, Y. Wang, K. Lee, and I. Rhee, "Tackling Bufferbloat in 3G/4G Networks," in *Proc. ACM IMC*, 2012.

[6] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet," *Communications of the ACM*, vol. 55, no. 1, 2012.

[7] I. Järvinen, B. Chemmagate, A. Y. Ding, L. Daniel, M. Isomäki, J. Korhonen, and M. Kojo, "Effect of Competing TCP Traffic on Interactive Real-Time Communication," in *Proc. PAM*. Springer-Verlag, 2013.

[8] R. Chakravorty, J. Cartwright, and I. Pratt, "Practical experience with TCP over GPRS," in *Proc. IEEE GLOBECOM*, Taiwan, Nov. 2002.

[9] X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang, "Experiences in a 3G Network: Interplay Between the Wireless Channel and Applications," in *Proc. ACM MOBICOM*, USA, 2008.

[10] M. Sågfors, R. Ludwig, M. Meyer, and J. Peisa, "Queue management for TCP traffic over 3G links," *Proc. IEEE WCNC*, 2003.

[11] I. Rhee and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," in *Proc. Protocols for Fast Long-distance Networks*, 2005.

[12] L. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control," *ACM CCR*, vol. 34, no. 2, 2004.

[13] V. Jacobson, "Congestion avoidance and control," *ACM CCR*, vol. 18, no. 4, 1988.

[14] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582 (Standard), Internet Engineering Task Force, Apr. 2012.

[15] S. Mascolo and F. Vacirca, "Congestion Control and Sizing Router Buffers in the Internet," in *Proc. of the IEEE Conference on Decision and Control*, 2005.

[16] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proc. IEEE INFOCOM*, 2004.

[17] L. De Cicco and S. Mascolo, "TCP Congestion Control over HSDPA: an Experimental Evaluation," *arXiv:1212.1621 [cs.NI]*, Dec. 2012.

[18] S. Ramachandran, "Web metrics: Size and number of resources," https://developers.google.com/speed/articles/web-metrics, May 2010.

[19] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, 1995.

[20] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)," RFC 6817 (Experimental), Internet Engineering Task Force, Dec. 2012.

Authorized licensed use limited to: Politecnico di Bari. Downloaded on May 12,2026 at 20:59:51 UTC from IEEE Xplore. Restrictions apply.