

Tackling the Challenge of Bufferbloat in Multi-Path Transport over Heterogeneous Wireless Networks

Simone Ferlin-Oliveira, Thomas Dreibholz, Özgü Alay
Simula Research Laboratory, Martin Linges vei 17, 1364 Fornebu, Norway
[{ferlin,dreibh,ozgu}](mailto:{ferlin,dreibh,ozgu}@simula.no)@simula.no

Abstract—Today, most of the smart phones are equipped with two network interfaces: Mobile Broadband (MBB) and Wireless Local Area Network (WLAN). Multi-path transport protocols provide increased throughput or reliability, by utilizing these interfaces simultaneously. However, multi-path transmission over networks with very different QoS characteristics is a challenge. In this paper, we studied Multi-Path TCP (MPTCP) in heterogeneous networks, specifically MBB networks and WLAN. We first investigate the effect of bufferbloat in MBB on MPTCP performance. Then, we propose a bufferbloat mitigation algorithm: Multi-Path Transport Bufferbloat Mitigation (MPT-BM). Using our algorithm, we conduct experiments in real operational networks. The experimental results show that MPT-BM outperforms the current MPTCP implementation by increasing the application goodput quality and decreasing MPTCP’s buffer delay, jitter and buffer space requirements.¹

I. INTRODUCTION

More than 25 years ago, window-based congestion control has been added to TCP to avoid network congestion collapses. The congestion control restricts the amount of in-flight data: losses trigger a reduction of this amount (“multiplicative decrease”), while successful transmissions allow for an increase (“additive increase”). At that time the router queues were reasonably small due to scarce and expensive memory capacity. However, today, memory capacities increase while prices decrease. This led to significantly larger router queue sizes and *bufferbloat* [1]: senders experience no losses and can increase their amount of in-flight data, growing latency and deteriorating application interactivity.

Multi-path transport utilizes multiple paths in the network, e.g. different ISPs, to improve the overall throughput. The IETF is currently standardizing multi-path transport as Multi-Path TCP (MPTCP) [2] for TCP and Concurrent Multi-path Transfer for the Stream Control Transmission Protocol (CMT-SCTP) [3] for SCTP [4]. Having underlying paths with different quality of service (QoS) characteristics requires sophisticated mechanisms to handle sender and receiver buffers, e.g. buffer splitting and chunk rescheduling [5], non-renegable selective acknowledgements [6] and others [2]. This is particularly the case in mobile broadband (MBB) networks, with packet delays of up to several seconds [7]. In this paper, we therefore analyze the behavior of MPTCP in real operational networks using the NORNET EDGE [8] testbed. We consider bufferbated MBB networks and WLAN, and identify issues of multi-path transport in terms of throughput, delay and buffer requirements. Based on this analysis, we propose an algorithm that copes with the challenges of multi-path transport in bufferbated wireless networks and show its applicability. Although our analysis is based on MPTCP, our approach is

generic and can also be adopted by other multi-path transport protocols, like e.g. CMT-SCTP, as well.

II. BACKGROUND

A. Multi-Path TCP

Multi-path transport has shown to be beneficial for bandwidth aggregation and to increase end-host robustness [2], [3], [9]. Although many devices support multi-path, the most prominent transport protocol, i.e. TCP, is still single path.

MPTCP [2] is a major extension of TCP that supports multi-path transmission. From the application perspective, MPTCP utilizes a single standard TCP socket, whereas lower in the stack, several subflows (conventional TCP connections) may be opened. MPTCP uses two levels of congestion control: at the subflow level, each subflow is in charge of its own congestion control. At the MPTCP level, coupled congestion control is used to provide fairness among subflows [10].

B. Heterogeneous Networks

MPTCP has been evaluated for data center networks on paths with similar characteristics [2]. However, MPTCP can be also used in scenarios where it is challenging to effectively utilize the paths [3], [5], [11]. One example is a smart phone with both WLAN and MBB interfaces showing different characteristics in terms of bandwidth, delay and packet loss.

In this paper, the smart phone use case is used to evaluate MPTCP performance over real operational MBB networks and WLAN. Although MBB networks are nowadays technologically closer to 4G and beyond, many MBB networks still provide 3G coverage. Therefore, we focus on 3G, specifically, 3G Universal Mobile Telecommunications System (UMTS). Theoretically, 3G to 3.5G offer peak throughputs from 5 up to several hundreds of Mbit/s and less than 40 ms round-trip time (3GPP Release 7: HSPA+) with low loss rates [7]. To handle channel variation and avoid losses, MBB operators apply large buffers, resulting in high delays, causing bufferbloat [12] and affecting the QoS performance of network protocols.

Another common access technology is WLAN. In this paper, we consider IEEE 802.11a/g that offers a peak throughput of 54 Mbit/s. It has similar delays as 3G UMTS (with HSPA+) networks; however, it is lossier compared to 3G UMTS [13].

C. Transport Protocol Limitations under Heterogeneity

Multi-path transport protocols have to handle paths with different QoS properties [3]. Receive window limitation and head of line (HOL) blocking are the two main factors impacting performance. Both are shortly introduced here.

In order to fully utilize the capacity of a path, a sender has to keep at least the data amount given by the Bandwidth-Delay Product (BDP) in flight. The BDP for a path i can be expressed as $BDP_i [B] = BW_i [B/s] \times \tau_i [s]$, where BW_i is the bandwidth and τ_i is the delay of link i . However, to avoid overloading the receiver, MPTCP applies window-based flow control: the

¹The authors would like to thank Olivier Bonaventure, Amund Kvalbein and Christoph Paasch for their friendly support.

maximum amount of acceptable data is signaled as advertised receiver window to the sender. Although in MPTCP, the notion of BDP is slightly different, as it aggregates the BDP of all subflows considering the highest RTT among them. Clearly, if the receive window is smaller than the BDP, the path capacity remains underutilized; see also [3, Chapter 2]. The advertised receiver windows depend on the receive buffer size; for multi-path transport, they must be sufficient for *all* paths to avoid blocking [3], [5].

1) Head-of-Line Blocking

Like TCP, MPTCP also provides ordered delivery. That is, all segments must be in order before the application reads them. In case of loss, all subsequent segments are held in the receive buffer until the lost packet is successfully received. This is denoted as HOL blocking [14]; it may reduce goodput and increase delay as well as jitter.

Heterogeneous paths worsen the problem, since segments arrive out-of-order from different paths. MPTCP applies two levels of receive buffering: on subflow level and on MPTCP level. First, each subflow is in charge of reordering its segments. Then, MPTCP reorders the segments from each of the subflows and finally delivers them to the application. Clearly, HOL blocking on one path (i.e. on the subflow level) also effects the overall performance of MPTCP in terms of goodput, buffer size requirements, delay and jitter.

D. Dealing with Path Heterogeneity

The Linux MPTCP implementation [2] is the reference implementation of MPTCP. Its current version 0.88 realizes a mechanism called *opportunistic retransmission and penalization* [2]. It tries to compensate for the receive window limitation due to Round Trip Time (RTT) differences by resending unacknowledged segment(s) on another subflow and halving the congestion window of the slow subflow, similar to chunk rescheduling for CMT-SCTP [3], [5]. Furthermore, the congestion window of the slow subflow is halved and its slow-start threshold is set to the new congestion window size. Further improvements have been introduced in [11], where – after the congestion window is halved – the slow-start threshold remains unmodified in the slow-start phase.

There are previous studies that analyze the impact of bufferbloat on transport protocols, although the focus is on wired networks and on the Internet [1], [15]. [16] is the only one that studied MBB networks in this context. The performance of CMT-SCTP with a bufferbloating DSL path is shown in [3], [17]. However, the impact of bufferbloat – particularly in wireless networks – on MPTCP has not been considered previously.

III. THE IMPACT OF BUFFERBLOAT ON MPTCP

We have used two different 3G UMTS ($3G_1$ and $3G_2$) to analyze MPTCP. Both are commercial operators in Norway with separate infrastructure. Although they use similar technologies, they operate independently, with significant performance difference. The public WLAN at Simula is used by approximately 100 people during work hours. Figure 1 and 2 show the congestion window, RTTs and application goodput of MPTCP over $3G_1$ and $3G_2$ as well as $3G_2$ and WLAN.

First, we examine $3G_1$ and $3G_2$. This is realistic, since many buses and trains in Norway provide WLAN-based Internet. However, behind the WLAN there is just another 3G connection. In Figure 1(a), we observe gaps in goodput (around 7 s). Such gaps are undesirable for delay-sensitive applications. The main reason for the gaps is HOL blocking due to high RTTs in Figure 1(b). Note that in both networks, RTTs increase rapidly due to bufferbloat. In fact, we observe

that MPTCP *penalizes* both links to avoid high RTTs in Figure 1(c) and 1(d). Subflow $3G_2$ is *penalized* around 2.5 s, and at about 5 s, it experiences a loss (here, the slow-start threshold is set). Similarly, the $3G_1$ subflow is penalized (at 3 s) but it increases its congestion window up to a loss (at 5.75 s). The penalization is not able to reduce the congestion window, which provides poor goodput quality.

$3G_2$ and WLAN is the next case. Clearly, this is a common scenario for smart phones illustrated. Figure 2 shows the results: similar to $3G_1$ and $3G_2$, we observe goodput gaps. As presented in Figure 2(c), MPTCP *penalizes* the $3G_2$ subflow, however, the penalization is not able to keep the congestion window small. Since $3G_2$ is bufferbloating, MPTCP becomes receive-window limited and, the WLAN capacity can not be fully utilized. Here, $3G_2$ has higher capacity than WLAN, however, $3G_2$ is the *penalized* path. In other words, the penalization algorithm penalizes the higher capacity link due to bufferbloat, resulting in poor overall MPTCP performance.

Bufferbloat creates an artificially high BDP that affects TCP profoundly. Congestion control needs to get a timely notification on packet drops due to congestion (acknowledgments). These notifications are then delayed by bufferbloat, which reduces the sender's ability to quickly react to changing network conditions. In [15], the authors found out that an end-host's concurrent traffic together with high queuing delays makes interactive traffic impossible. Finally, high buffer delays make TCP unable to correctly estimate the BDP and overshoots the path in slow-start, while being defeated in congestion avoidance [1]. These consequences are more severe with multi-path. The transport protocol requires large amounts of buffer to accommodate out-of-order packets. This can starve memory, aggravating receive window limitation and HOL blocking.

IV. MULTI-PATH TRANSPORT BUFFERBLOAT MITIGATION (MPT-BM)

Motivated by Section III, we developed an algorithm that mitigates bufferbloat in MBB networks: Multi-Path Transport Bufferbloat Mitigation (MPT-BM) that caps the RTTs by limiting the amount of data to be sent on each subflow, hence, controlling bufferbloat. Particularly, our goal is to not only improve goodput, but also reduce the receiver's delay, jitter and buffer space requirements.

Note that such a mechanism can be compared to a delay-based congestion control in MPTCP, although this is currently unavailable. A bufferbloat avoidance algorithm for TCP is described in [12], where the receiver estimates the sender's congestion window dynamically to adjust its advertised window size. The authors argue that placing the algorithm at the receiver, e.g., smart phone, is less troublesome than changing server machines. We implemented their algorithm and observed that it mitigates bufferbloat at an expense of reduced goodput. This is mainly due to response times and estimation inaccuracy. Therefore, we borrow the main idea of this work and make use of it for MPT-BM. This can be also implemented in other multi-path transport protocols such as CMT-SCTP.

The main idea behind MPT-BM is to capture bufferbloat by monitoring the difference between $sRTT_{min}$ and $sRTT$ for each subflow. Whenever $sRTT$ drifts apart from $sRTT_{min}$ of the same subflow, as a result of sending data, we take it as an indication of bufferbloat. Therefore, the congestion window for each subflow is limited by an upper bound $cwnd_{limit}$:

$$cwnd_{limit} = \lambda * (sRTT_{min}/sRTT) * cwnd.$$

where λ is a configuration parameter that determines the *toler-*

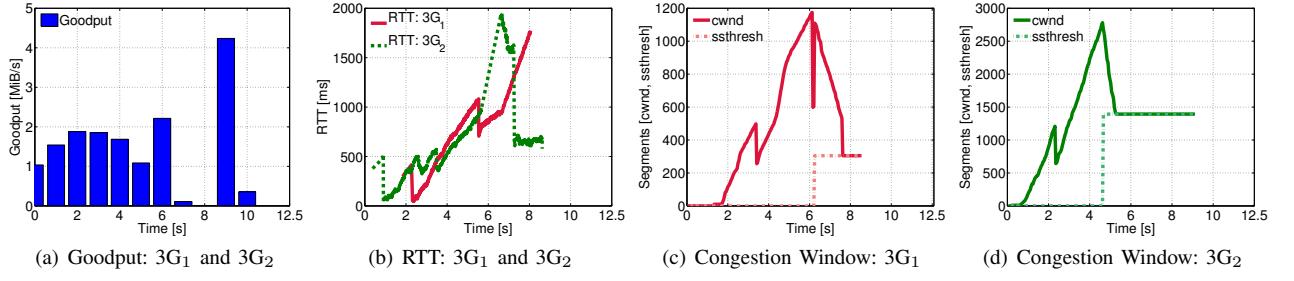


Fig. 1. An MPTCP Example: 3G₁ and 3G₂

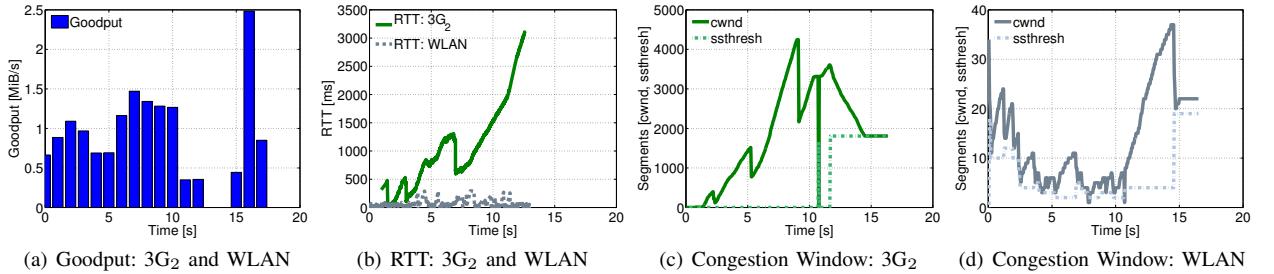


Fig. 2. An MPTCP Example: 3G₂ and WLAN

Algorithm 1 Per-Subflow Bufferbloat Mitigation by MPT-BM

Initialization:

$sRTT \leftarrow \infty$; $sRTT_{min} \leftarrow \infty$

RTT estimation:

$sRTT_{min} \leftarrow \min(sRTT_{min}, sRTT)$

How many segments can be sent?

$cwnd_{limit} \leftarrow \lambda * (sRTT_{min}/sRTT) * cwnd$
 $send \leftarrow \begin{cases} \max(0, (\min(cwnd, cwnd_{limit}) - inflight)) & (sRTT_{min} \geq \Theta) \\ \max(0, cwnd - inflight) & (sRTT_{min} < \Theta) \end{cases}$

ance between $sRTT_{min}$ and $sRTT$. Currently, it is set globally (`sysctl`). In the future, it could be configured by the application (socket option) to adapt it to application-specific QoS requirements. $sRTT_{min}$ is the minimum $sRTT$ over a pre-defined window given that the minimum expected RTT might change, e.g., on handover from 3G to 2G.

The steps of MPT-BM are summarized in Algorithm 1. First, the algorithm initializes $sRTT_{min}$ and $sRTT$. Then, $sRTT_{min}$ is recomputed after each $sRTT$. Next, $cwnd_{limit}$ is computed. The sender keeps sending packets, as long as the $cwnd_{limit}$ is smaller than $cwnd$. We also define the *RTT threshold* Θ to activate MPT-BM on particular subflow(s). Θ can be set through a `sysctl`, and if it is set to zero, MPT-BM runs for all subflows. Otherwise, MPT-BM is activated whenever $sRTT_{min}$ exceeds Θ . We find this necessary, since there will be scenarios with certain links, where bufferbloat is negligible.

V. EXPERIMENT SETUP

We perform our evaluations in a NORNET EDGE testbed [7], [8], [18] setup, as shown in Figure 3. We consider two different operational UMTS MBB networks in Oslo, Norway (3G₁ and 3G₂). The WLAN access point connects ca. 100 people during work hours in a large office complex with several other interfering WLAN networks. MPTCP was tested using bulk transfer of 16 MiB data blocks in downlink direction.

For our evaluation, we use two different scenarios. First, we consider 3G and WLAN, using the smart phone use case and providing a good example to path heterogeneity in terms of bandwidth, delay and loss. Our second scenario analyses

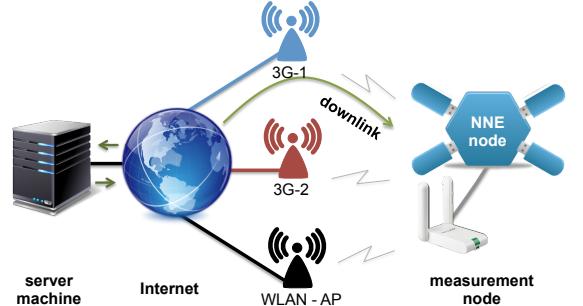


Fig. 3. Measurement Setup: MPTCP over 3G and WLAN

two 3G connections. This scenario is also realistic, especially on buses and trains where WLAN access is provided over a MBB connection. We collect around 50 measurements for each scenario. All measurements are performed in the same networks and at the same locations over a period of 4 weeks.

On the system level, we flush all TCP metrics before each experiment to avoid any dependencies. Auto-tuning was left on [11]. We consider unbounded buffers, since MPTCP performance increases with larger buffer sizes [11]. Linux MPTCP version 0.88, i.e. the current state-of-the-art version, is used. In addition, Reno loss-based congestion control was chosen for TCP as well as OLIA [19] for MPTCP [20].

VI. RESULTS

MPTCP's *penalization and opportunistic retransmission* algorithm tries to reduce receive window limitation caused by RTTs without taking RTTs into account. Thus, with bufferbloat, or any scenario without a steady packet flow, it does not solve the problem. From Section III, MPTCP halves the congestion window and inflates it again in slow-start.

MPT-BM has a different approach in which it tries to monitor the RTTs of the subflows. It works based on the drift between $sRTT$ and $sRTT_{min}$. This way, it controls the data to be sent. This is evaluated after every $sRTT$ update and, hence, is responsive to sudden delay changes. Furthermore, MPT-BM attempts to bound RTTs to trade off between *goodput volume* versus *quality*, i.e. it tries to reduce delay and jitter. This leads to lower requirements for buffer space in the end-hosts and

Network	RTT [ms]	Bandwidth [KiB/s]
3G ₁	60 up to 500	380
3G ₂	80 up to 850	1300
WLAN	20 up to 30	950

TABLE I
AVERAGE RTT AND BANDWIDTH OF THE NETWORK DOWNLINK

improves the protocol's responsiveness to network events. In the following, we show how bufferbloat mitigation improves the overall application goodput quality.

A. Heterogeneous Networks

Table I provides an overview of the path characteristics of each network in terms of average RTTs and bandwidth measured with TCP. Note that the high RTTs for both 3G₁ and 3G₂ are caused due to bufferbloat transferring a large file. In particular, 3G₂ shows larger buffers compared to 3G₁; we could observe up to 2 MiB in-flight data in this network. Since 3G₁ has smaller buffers than 3G₂, it experiences loss earlier, and thus, it keeps the congestion window and RTTs relatively smaller. On the other hand, the WLAN network has very low RTTs, but it is also lossier compared to 3G networks. Therefore, the TCP throughput is quite low.

In order to evaluate MPTCP with MPT-BM, we consider three different scenarios: (a) 3G₁+3G₂, (b) 3G₁+WLAN, and (c) 3G₂+WLAN. In our experiments, we did not run MPT-BM on WLAN, since WLAN was not bufferbated compared to 3G. We also set a threshold to sRTT_{min} ($\Theta=100$ ms), and MPT-BM is only active when a connection sRTT_{min} exceeds Θ .

B. RTT Bounding for Bufferbloat Mitigation

From Section IV, MPT-BM bounds the RTTs based on λ . As λ increases, so the *delay tolerance* between sRTT and sRTT_{min} increases, hence, the bufferbloat. For example, $\lambda = 3$, allows for $3 \times \text{sRTT}_{\min}$. We tested different values for λ , and based on our observations, we show results for $\lambda = 1.5$ and $\lambda = 3$ to evaluate the effect on MPTCP. We compare current MPTCP and MPTCP with MPT-BM in terms of the average RTTs for each subflow of the same MPTCP session in Figure 4. The results show that MPT-BM bounds the RTTs in both 3G₁ and 3G₂ according to λ , where $\lambda = 1.5$ achieves lower average RTTs compared to $\lambda = 3.0$, as expected.

C. Application Goodput Quality

Next, the effect of MPT-BM on the application is shown. We evaluate the average goodput, μ , and the goodput coefficient of variation, σ/μ , where σ is the goodput standard deviation.

Subfigure 5(a) shows that, when bufferbloat is avoided with MPT-BM, the application goodput volume increases slightly compared to current MPTCP in all scenarios. Except in the scenario (c) 3G₂+WLAN, the goodput with MPT-BM $\lambda = 3.0$ is 13% higher than for current MPTCP. Note that, in this scenario, MPT-BM $\lambda = 1.5$ shows a marginal improvement in goodput volume but the variance is higher compared to current MPTCP. We can explain that by looking at the delay values of provider 3G₂. If λ is set to an arbitrarily small value, the connection will be throttled. In (a) 3G₁+3G₂, the goodput is only 8% higher. In the (b) 3G₁+WLAN scenario, the goodput improvement is marginal. This is because the buffers in the 3G₂ network are larger compared to the 3G₁ network. However, when comparing current MPTCP and MPTCP with MPT-BM in terms of goodput variance, MPT-BM outperforms current MPTCP in all settings and scenarios. In (a) 3G₁+3G₂, MPT-BM for $\lambda = 3.0$ increases the average goodput volume by 8% and reduces its variation by closely 45%. In (c) 3G₂+WLAN, the goodput volume is increased by 13%, while the coefficient of variation is reduced by 23%.

In terms of completion time, there is on average an improve-

	3G₁ + 3G₂	Median	75%	95%
Current MPTCP	14.4	40.9	81.1	
MPT-BM, $\lambda = 1.5$	1.9	4.3	9.1	
MPT-BM, $\lambda = 3.0$	4.2	8.7	20.6	
3G₁ + WLAN				
Current MPTCP	11	29	82.3	
MPT-BM, $\lambda = 1.5$	4.49	9.1	20.9	
MPT-BM, $\lambda = 3.0$	8.7	17.5	44.5	
3G₂ + WLAN				
Current MPTCP	10.5	25	54.1	
MPT-BM, $\lambda = 1.5$	2.3	4.8	9.1	
MPT-BM, $\lambda = 3.0$	4.48	6.1	7.5	

TABLE II
MPTCP BUFFER DELAY [ms]: MEDIAN, 75% AND 95% PERCENTILES

ment of 10% for all scenarios with MPT-BM for $\lambda = 3.0$. In (a) 3G₁+3G₂, MPTCP took 13.5 s against 12.1 s for MPTCP with MPT-BM; in (b) 3G₁+WLAN, MPTCP took 15.4 s, against 14 s of MPT-BM; and in (c) 3G₂+WLAN 10.9 s for current MPTCP against 9.6 s for MPTCP with MPT-BM. The improvement with MPT-BM for $\lambda = 1.5$ was marginal. This is due to small tolerance between sRTT and sRTT_{min}, capping the amount of data to be sent. Thus, λ should not be arbitrarily low unless the RTT characteristics of the paths are known.

While the improvement in goodput volume in networks with smaller buffers is marginal, the improvement in variance is remarkable. This is beneficial for both, multi-path transport protocol and for delay-sensitive applications. In the following, we show the benefits of MPT-BM for buffer delays and jitter.

D. Buffer Delay

Mitigating bufferbloat affects RTTs. In heterogeneous networks, this will limit the RTT differences among subflows, having a direct impact on buffer delays due to re-ordering.

Figures 6 show the CDF of MPTCP level buffer delays, for MPTCP with MPT-BM ($\lambda = 1.5$ and $\lambda = 3.0$) and current MPTCP in all different scenarios: (a) 3G₁+3G₂, (b) 3G₁+WLAN, and (c) 3G₂+WLAN. Buffer delay is measured at the MPTCP level by tracking the amount of time segments stay in the MPTCP out-of-order queue. We normalized segments to 1024 bytes, since this was on average the minimum amount of bytes written in the buffer. When RTTs are bounded, there is significantly less delay, and this is aligned to our observations in Subsection VI-C regarding application goodput. The faster the application can read data, the lower the application goodput delay and jitter are. Buffer delay gains is more evident in (a) 3G₁+3G₂ and (c) 3G₂+WLAN, where 3G₂ buffers are larger compared to 3G₁. In (b) 3G₁+WLAN, the improvement compared to MPTCP is also significant: approximately 80% of the cases MPTCP with MPT-BM shows less than 20 ms buffer delay with $\lambda = 1.5$ and $\lambda = 3.0$ compared to current MPTCP. However, comparing (a) 3G₁+3G₂ and (c) 3G₂+WLAN to MPTCP shows a significant improvement: in both scenarios, buffer delays with both MPT-BM $\lambda = 1.5$ and $\lambda = 3.0$ do not exceed 20 ms, whereas current MPTCP shows considerably higher delay values.

In [21] the authors also observed high buffer delays in MPTCP with large buffer sizes. This confirms our observations in Figure 6, where current MPTCP experiences higher buffer delays in all scenarios. In MPTCP, data reordering is first done on the subflow levels, and after that, on the MPTCP level. This additional level of buffer delay could increase application goodput delay if it is taken properly into account. Looking through our dataset, scenarios (a) 3G₁+3G₂ and (c) 3G₂+WLAN with MPTCP show maximum buffer delay of 1 s up to 3.5 s, where MPT-BM maximum buffer delays were seldom over few hundreds of milliseconds.

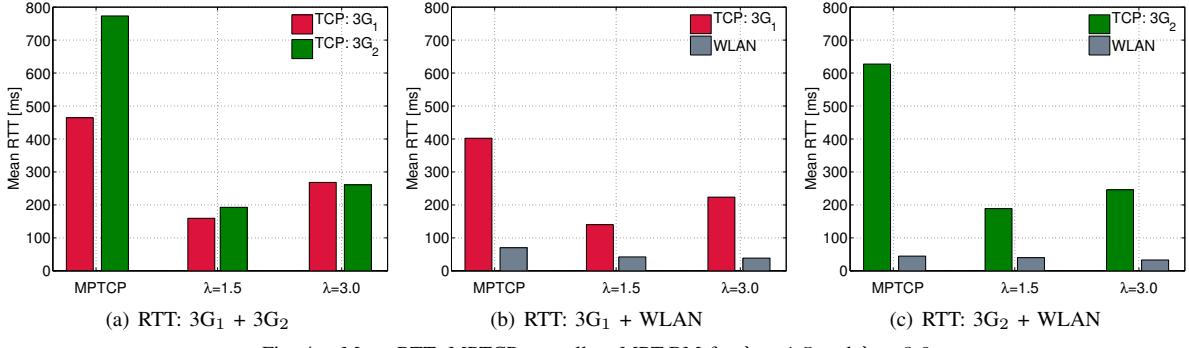


Fig. 4. Mean RTT: MPTCP as well as MPT-BM for $\lambda = 1.5$ and $\lambda = 3.0$

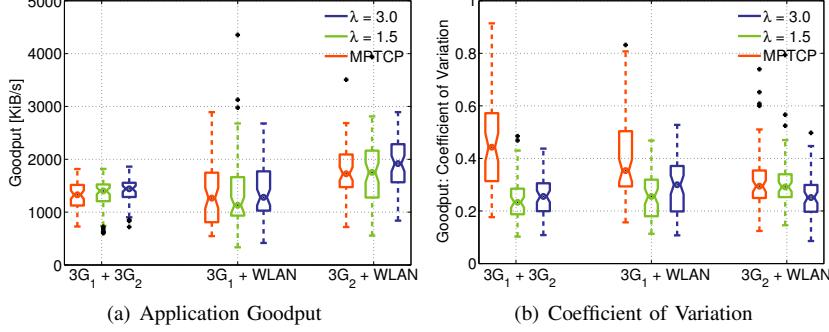


Fig. 5. Goodput and Coefficient of Variation

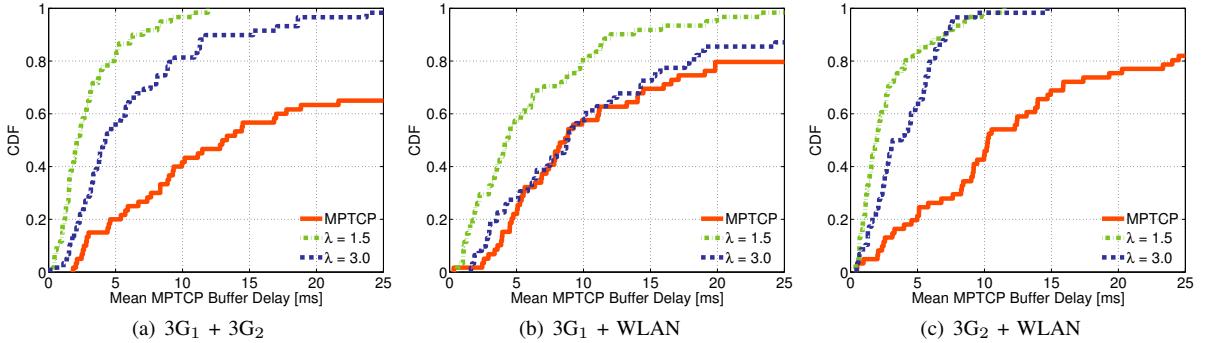


Fig. 6. MPTCP Buffer Delay

Table II shows median, 75% and 95% percentiles for buffer delay with current MPTCP and MPTCP with MPT-BM ($\lambda = 1.5$ and $\lambda = 3.0$). MPT-BM with both λ reduces the time in the MPTCP out-of-order queue. In 3G₂+WLAN, one can see a single case for 95% percentile with MPT-BM $\lambda = 1.5$ compared to $\lambda = 3.0$, where delay for $\lambda = 1.5$ is higher. We can explain it, because 3G₂ has higher delays compared to the other 3G provider, and λ being arbitrarily small could cap the congestion window, throttling the connection.

E. Buffer Size Requirements

Besides delay reductions, it is interesting to look at buffer size requirements for the end-hosts. Therefore, Figure 7 presents the CDF of the buffer sizes measured at the MPTCP level to keep track of the memory allocated for segments. As shown, MPT-BM has significantly less need for buffer space on the MPTCP level. This is also related to less reordering at the receiver, mainly caused by differences in the RTTs of the subflows. This matches our observations in Subsections VI-C and VI-D: the faster the application can read data, the lower is the application goodput jitter and the delay in MPTCP reordering. Thus, less buffer space is required.

Here, the gain in terms of buffer is evident in all scenarios.

	3G₁ + 3G₂	Median	75%	95%
Current MPTCP	5353	6269	6807	
MPT-BM, $\lambda = 1.5$	2320	2461	2660	
MPT-BM, $\lambda = 3.0$	2759	3168	5128	
3G₁ + WLAN				
Current MPTCP	4029	4667	5560	
MPT-BM, $\lambda = 1.5$	2183	3366	4368	
MPT-BM, $\lambda = 3.0$	2659	3479	4940	
3G₂ + WLAN				
Current MPTCP	4400	5017	5781	
MPT-BM, $\lambda = 1.5$	2139	2591	3216	
MPT-BM, $\lambda = 3.0$	2244	2573	3620	

TABLE III
MPTCP BUFFER SIZE [KiB]: MEDIAN, 75% AND 95% PERCENTILES

In (a) 3G₁+3G₂ there are few cases where current MPTCP needs less buffer compared to MPTCP and MPT-BM ($\lambda = 1.5$ and $\lambda = 3.0$). However, in more than 90% of the cases, more buffer space is required with MPTCP. In (c) 3G₁+WLAN, MPTCP needs almost twice the buffer space compared to MPT-BM with both $\lambda = 1.5$ and $\lambda = 3.0$. Note that our measurements consider unbounded buffers (16 MiB). For better comparability, Table III also shows median, 75% and 95% percentiles for MPTCP buffer delay for current MPTCP

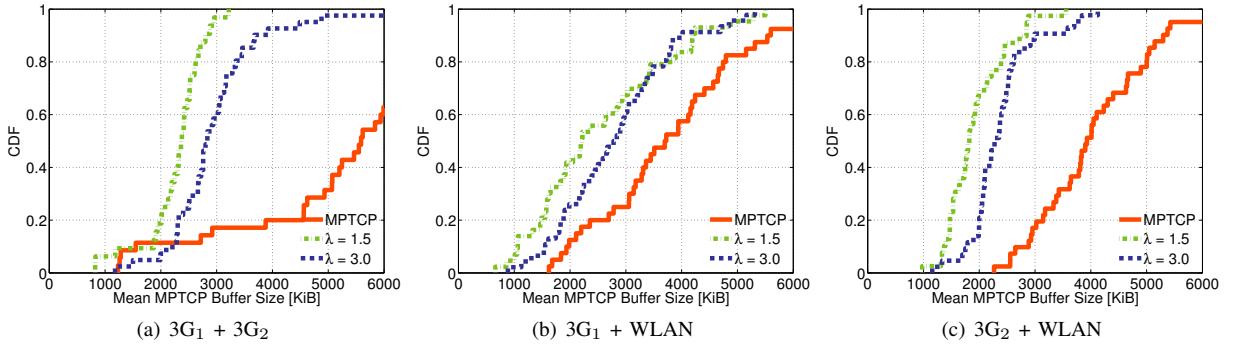


Fig. 7. MPTCP Buffer Size

and MPTCP with MPT-BM ($\lambda = 1.5$ and $\lambda = 3.0$).

While buffer delay in Subsection VI-D is relevant for delay-sensitive applications, buffer size is central for end-hosts and multi-path transport protocol implementations. Both are crucial for busy servers with many MPTCP connections and end-hosts with limited memory, e.g., smart phones.

VII. CONCLUSIONS AND FUTURE WORK

In this work, we investigated the effect of bufferbloat on MPTCP in heterogeneous wireless networks. We proposed an algorithm, Multi-Path Transport Bufferbloat Mitigation (MPT-BM), to mitigate bufferbloat for multi-path transport in heterogeneous wireless networks. We consider 3G UMTS, WLAN and different ISPs in terms of *path heterogeneity*. We show that, although 3G UMTS networks use similar technologies, the different ways they are operated profoundly affect the performance of transport protocols such as MPTCP.

We showed that MPT-BM not only improves goodput volume, but also goodput variance compared to the state-of-the-art Linux MPTCP version 0.88. While the improvement in goodput volume is marginal in networks without bufferbloat, goodput variance is remarkably lower in all tested scenarios. Furthermore, we analyzed the system requirements in terms of buffer delay and size, showing that MPT-BM requires smaller buffers, hence, resulting in shorter delays compared to MPTCP. To sum up, bounding the RTTs of each of the subflows in MPTCP with MPT-BM increases the performance and robustness of multi-path transport. And the idea implemented in MPT-BM could be also beneficial for other multi-path transport protocols providing in-order data delivery.

We leave further improvements and integration of MPT-BM for future work. One possible direction could be to use λ to improve scheduling decisions. Moreover, λ could also be set by the application as an indication of *tolerance* to delay. MPTCP could also take λ as a parameter to decide which subflows should be active within an MPTCP session. MPT-BM could be also evaluated against delay-based congestion control for MPTCP, but such congestion control for MPTCP is currently unavailable.

REFERENCES

- [1] V. Cerf, V. Jacobson, N. Weaver, and J. Gettys, "BufferBloat: What's Wrong with the Internet?" *ACM Queue*, vol. 9, no. 12, 2011.
- [2] C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchêne, O. Bonaventure, and M. Handley, "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, 2012.
- [3] T. Dreibholz, "Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol," Habilitation Treatise, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, 2012.
- [4] T. Dreibholz, I. Rüngeler, R. Seggelmann, M. Tüxen, E. P. Rathgeb, and R. R. Stewart, "Stream Control Transmission Protocol: Past, Current, and Future Standardization Activities," *IEEE Communications Magazine*, vol. 49, no. 4, 2011.
- [5] T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tüxen, "On the Use of Concurrent Multipath Transfer over Asymmetric Paths," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2010.
- [6] P. Natarajan, N. Ekiz, E. Yilmaz, P. D. Amer, and J. R. Iyengar, "Non-Renewable Selective Acknowledgments (NR-SACKS) for SCTP," in *Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP)*, 2008.
- [7] S. Ferlin-Oliveira, T. Dreibholz, Özgür Alay, and A. Kvalbein, "Measuring the QoS Characteristics of Operational 3G Mobile Broadband Networks," in *Proceedings of the 4th International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, 2014.
- [8] A. Kvalbein, D. Baltrūnas, K. R. Evensen, J. Xiang, A. Elmokashfi, and S. Ferlin-Oliveira, "The NorNet Edge Platform for Mobile Broadband Measurements," *Computer Networks, Special Issue on Future Internet Testbeds*, vol. 61, 2014.
- [9] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving Datacenter Performance and Robustness with Multipath TCP," in *Proceedings of the ACM SIGCOMM Conference*, 2011.
- [10] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, "On the Fairness of Transport Protocols in a Multi-Path Environment," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2012.
- [11] C. Paasch, R. Khalili, and O. Bonaventure, "On the Benefits of Applying Experimental Design to Improve Multipath TCP," in *Proceedings of the 9th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2013.
- [12] H. Jiang, Y. Wang, K. Lee, and I. Rhee, "Tackling Bufferbloat in 3G/4G Networks," in *Proceedings of the 2012 ACM Internet Measurement Conference (IMC)*, 2012.
- [13] Y.-C. Chen, Y. Lim, R. J. Gibbens, E. M. Nahum, and D. Towsley, "A Measurement-Based Study of MultiPath TCP Performance over Wireless Networks," in *Proceedings of the 13th ACM Internet Measurement Conference (IMC)*, 2013.
- [14] M. Scharf and S. Kiesel, "Head-of-line Blocking in TCP and SCTP: Analysis and Measurements," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2006.
- [15] S. Alfredsson, G. D. Giudice, J. Garcia, A. Brunström, L. D. Cicco, and S. Mascolo, "Impact of TCP Congestion Control on Bufferbloat in Cellular Networks," in *Proceedings of the 14th IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2013.
- [16] H. Jiang, Z. Liu, Y. Wang, K. Lee, and I. Rhee, "Understanding Bufferbloat in Cellular Networks," in *Proceedings of the ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design (CellNet)*, 2012.
- [17] H. Adhari, T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tüxen, "Evaluation of Concurrent Multipath Transfer over Dissimilar Paths," in *Proceedings of the 1st International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, 2011.
- [18] T. Dreibholz, "NorNet – An Open, Large-Scale Testbed for Multi-Homed Systems," Invited Talk at Swinburne University, Centre for Advanced Internet Architectures (CAIA), 2014.
- [19] R. Khalili, N. Gast, M. Popovic, and J.-Y. L. Boudec, "MPTCP is not Pareto-Optimal: Performance Issues and a Possible Solution," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, 2013.
- [20] ———, "MPTCP is not Pareto-Optimal: Performance Issues and a Possible Solution," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.
- [21] M. Li, A. Lukyanenko, S. Tarkoma, Y. Cui, and A. Ylä-Jääski, "Tolerating Path Heterogeneity in Multipath TCP with Bounded Receive Buffers," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2013.