Multi-Path Transport over Heterogeneous Wireless Networks: Does it really pay off?

Simone Ferlin, Thomas Dreibholz, Özgü Alay
Simula Research Laboratory
Centre for Resilient Networks and Applications
Martin Linges vei 17, 1364 Fornebu, Norway
{ferlin,dreibh,ozgu}@simula.no

***Abstract*—Multi-path transfer protocols such as Concurrent Multi-Path Transfer for SCTP and Multi-Path TCP (MPTCP), are becoming increasingly popular, due to widespread deployment of smartphones with multi-homing support. Although the idea of using multiple interfaces simultaneously to improve application throughput is tempting, does transmission over multiple interfaces always provide benefits especially in realistic setup?**

In this paper, we first show that multi-path transfer might actually have a negative impact in real-world scenarios with mobile broadband and wireless LAN networks. We then introduce our Dynamic Relative Path Scoring (DRePaS) algorithm that continuously evaluates the contribution of paths to the overall performance and dynamically influences the scheduling decisions to make best use of the paths for the overall system performance. We show that DRePaS outperforms the current MPTCP implementation in terms of throughput and application delay, especially when the links are heterogeneous.

Keywords: Multi-Path Transport, Multi-Path TCP, Heterogeneous Networks, Mobile Broadband, Performance

I. Introduction

Back in 1981, TCP has been standardized for reliable, ordered, connection-oriented byte transfer over a packet-oriented network. Together with later enhancements and extensions, it has to cope with heterogeneous networks and varying Quality of Service (QoS) – in terms of bandwidth, delay and loss – in the network. Particularly, these QoS conditions affect flow- and congestion control; and problems still remain [1], [2]. Recently, Multi-Path TCP (MPTCP) [3] has been defined to make simultaneous use of multiple paths in the network. Clearly, with multiple heterogeneous paths, it is more challenging to provide "good" service in terms of achieved goodput and latency. This issue is generic and applies e.g. to Concurrent Multi-Path Transfer for SCTP (CMT-SCTP) [4] as well.

Particularly for multi-path transfer, issues on one path affect the whole transmission: since data at a receiver instance is delivered to the application in-order and reliably, a missing segment on a problematic path delays the delivery of already-received data until a successful reception of the missing segment. This problem, denoted as Head-of-Line (HOL) blocking [2], is particularly problematic when the receive buffer size is limited (which is the case for almost all realistic application scenarios). Then, as soon as the amount of in-flight data becomes smaller than the Bandwidth Delay Product (BDP) of a path, the network capacity remains underutilized and the overall throughput of the multi-path connection suffers.

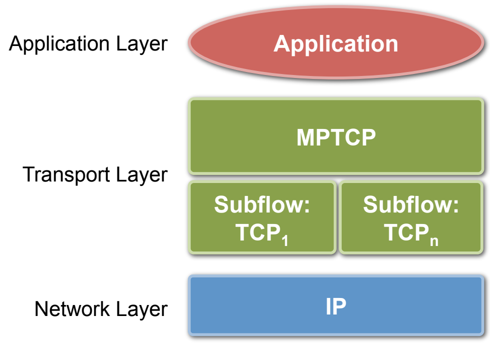


Fig. 1. The MPTCP Architecture

In this paper, we examine a realistic smart-phone multi-path transfer scenario: the combination of a Wireless LAN (WLAN) with a Mobile Broadband (MBB) path. Clearly, when using multipath transfer, the goal is to:

1) achieve path redundancy,
2) reduce latency and
3) improve the overall throughput.

However, we show that in certain scenarios, the current MPTCP implementation performs worse than the best available path in terms of throughput, but only provides redundancy. To overcome this problem, we propose an advanced path management algorithm, denoted as *dynamic relative path scoring* (DRePaS), that continuously evaluates the relative path performance and adapts the scheduling of payload data accordingly. Its performance is evaluated in real-world setups.

II. Background

A. Multi-Path TCP

Multi-path transport has shown to provide benefits, from bandwidth aggregation [4], [5] to increased robustness [6]–[8]. Although many devices support multi-homing, the most prominent transport protocol today, i.e. TCP, still only supports single-path transmission. The MPTCP extension adds multi-path transmission to TCP. Its design is motivated by the need to be compatible to network middleboxes and hence, it is backwards-compatible to TCP [3]. From the application perspective, MPTCP utilizes a single standard TCP socket [9] on the MPTCP level, whereas lower in the stack, several subflows that are conventional TCP connections may be opened. This way, an application only has to deal with a single MPTCP connection and therefore may remain unmodified. MPTCP provides two levels of congestion control, as illustrated in Figure 1: at the subflow level, each subflow is in charge of its own congestion control. At the MPTCP level, coupled congestion control [10] is provided to fairly share the network links among the subflows [11].

B. Heterogeneous Networks

A common scenario for MPTCP has been data center networks, where paths have similar characteristics [3]. However, MPTCP can also be used in other scenarios, where paths can be heterogeneous and especially challenging to effectively utilize them [4], [5], [12], [13]. A smartphone is a common example, since it usually provides two different network interfaces: WLAN and MBB. Both technologies have distinct characteristics in terms of bandwidth, delay and packet loss.

In this paper, we consider the smartphone use case to evaluate the MPTCP performance and study real operational MBB networks and WLAN. Although MBB networks are nowadays technologically closer to 4G and beyond, many operational MBB networks still provide 2G or 3G coverage only. Therefore, in this paper, we focus on 2G and 3G networks, specifically, the Universal Mobile Telecommunications System (UMTS). Theoretically, 3G and 3.5G networks offer peak throughputs from 5 Mbit/s up to several hundreds of Mbit/s and 40 ms delay with very low loss rates [14].

Another very common access technology is WLAN. In this paper, we specifically consider IEEE 802.11a/g that offers a peak throughput of 54 Mbit/s. WLAN has shown to have comparable delays to 3G UMTS (with HSPA+) networks [15], [16]. However, it is lossier compared to 3G UMTS [17].

C. Implications of Path Heterogeneity

A multi-path transport protocol must handle path heterogeneity, i.e., paths having different QoS properties [4], [5].

1) Receive Window Limitation: In order to fully utilize the capacity of a path, a sender has to keep at least the amount of data given by the Bandwidth-Delay Product (BDP) in flight. The BDP for a path i can be expressed as:

$$\mathrm{BDP}_i\ [\mathrm{B}] = \rho_i\ [\mathrm{B/s}] * \delta_i\ [\mathrm{s}]$$

where ρ_i is the bandwidth and δ_i is the delay of path i.

However, in order to avoid overloading the receiver, MPTCP applies window-based flow control: the 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.

Algorithm 1 Penalization and Retransmission

$$\text{cwnd} \leftarrow \text{cwnd} / 2$$
$$\text{ssthresh} \leftarrow \begin{cases} \text{cwnd} & \text{if ssthresh} == \infty \\ \text{ssthresh} & \text{otherwise} \end{cases}$$

A detailed introduction to the underlying transport mechanisms is provided in [4, Chapter 2]. The advertised receiver windows depend on the receive buffer sizes; for multi-path transport, it is particularly necessary to take care of having a sufficient size for *all* paths to mitigate blocking [4], [5].

2) Head-of-Line Blocking: Similar to TCP, MPTCP also provides in-order delivery. In case of a packet loss, all subsequent segments are held in the receive buffer until the lost packet is retransmitted and successfully received. Since all preceding segments are held in the buffer, the application does not obtain any new data until the missing segment is recovered. This effect, which is denoted as HOL blocking [18], may reduce goodput and increases delay as well as jitter. Heterogeneous paths aggravate this problem, since segments arrive out-of-order through different paths.

MPTCP applies two levels of receive buffering: subflow level and MPTCP level. First, each subflow reorders the segments before delivering them to the MPTCP buffer. Second, at the MPTCP level, reordered segments from each subflow are put together and again reordered before they are delivered to the application. Clearly, HOL blocking on one path (i.e. on the subflow level) also affects the MPTCP-level performance.

D. Dealing with Path Heterogeneity

The MPTCP Linux implementation [3][1] is the reference and used by the IETF MPTCP working group. It can therefore be considered as state of the art in MPTCP research and development. Its current version 0.88.8 realizes a mechanism called *opportunistic retransmission and penalization* that has first been described in [3]. It tries to compensate for the receive window limitation caused by Round Trip Time (RTT) differences in the subflows by resending unacknowledged segment(s) on another subflow, similar to chunk rescheduling for CMT-SCTP [4], [5]. Furthermore, the congestion window of the subflow holding up the connection is halved and its slow-start threshold is set to the new congestion window size (see Algorithm 1). Some further improvements have been introduced in [12], but the original idea of the algorithm is kept similar. The main difference is that after the congestion window is halved, the slow-start threshold is only set if not being in the initial show-start phase.

While *opportunistic retransmission and penalization* to some extent prevents massive out-of-order reception and, consequently, HOL blocking, it does not reduce the effect of extreme RTT heterogeneity, mostly aggravated by *bufferbloat* [19]. As we will show in this paper, this brings severe consequences at the receiver in terms of application delay and jitter, and consequently, affecting the application goodput. In the following, we therefore discuss these effects.

[1] MPTCP Linux Kernel Implementation: http://www.multipath-tcp.org.

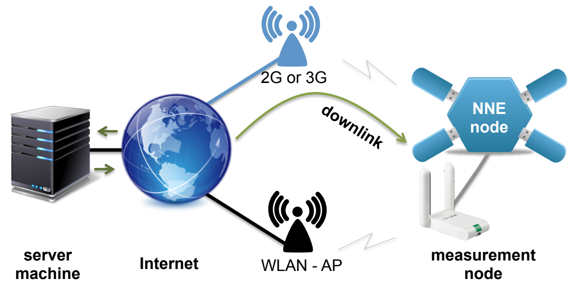


Fig. 2. Measurement Setup: MPTCP over 2G or 3G and WLAN

III. Experiment Setup

Our evaluations are performed in the NorNet Edge[2] testbed [14], [20], as illustrated in Figure 2. We consider one operational UMTS MBB network in Oslo, Norway. It is labelled as "2G or 3G". The WLAN access point is a public WLAN hotspot, connecting ca. 100 people during work hours in a large office complex with several interfering WLAN networks. In this real-world environment, MPTCP was tested with bulk transfer downloading 32 MiB data blocks.

For our evaluation, we use three different scenarios with two subflows: 3G and WLAN, 2G and WLAN, and 2G and 3G (e.g. a WLAN access point with MBB in the backhaul). These scenarios basically utilize the smartphone use case and provide a good example to path heterogeneity in terms of bandwidth, delay and loss. Especially, when considering WLAN and MBB networks, the RTT differences can be significant, since WLAN commonly has lower RTTs and it seldom has excessive buffers compared to MBB networks. We collect around 50 measurements for each scenario in the same networks and at the same locations over 4 weeks.

On the system level, we flush all TCP metrics caches to avoid any dependency between experiments. TCP buffer sizes were set to the values of Android for WLAN connections [21], i.e., 1 MiB for send and 2 MiB for receiver buffer, with auto-tuning turned on [12]. Linux MPTCP version 0.88.8, i.e. the current state-of-the-art version, is used for all of our measurements. In addition, Reno [22] loss-based congestion control was chosen for TCP and OLIA for MPTCP [10].

IV. Real-World MPTCP Bandwidth Aggregation

A common metric used to evaluate the bandwidth aggregation of MPTCP is *bandwidth aggregation benefit* B [23], and it indicates on a scale of -1 to 1 how beneficial multi-path (here: MPTCP) is compared to single-path transfer (here: TCP):

$$B = \begin{cases} [-1;0) & \text{if MPTCP} < \text{TCP} \\ 0 & \text{if MPTCP} == \text{TCP} \\ (0;1] & \text{if MPTCP} > \text{TCP} \end{cases}$$

Figure 3 illustrates the bandwidth aggregation metric and how MPTCP should act based on the overall multi-path performance B. We distinguish two cases: first, any value on the left side of 0 should be an indication that an additional subflow hurts the overall performance, hence it should not be utilized. Second, any value larger than 0 means that the subflow improves the resulting aggregated bandwidth and should therefore be *active*.

[2] NorNet testbed: https://www.nntb.no.

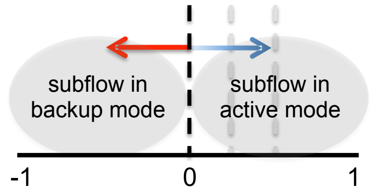


Fig. 3. Bandwidth Aggregation Benefit and Possible Reactions

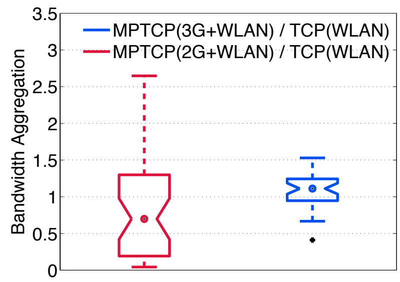


Fig. 4. MPTCP Performance Relative to Single-Path TCP

In general, we would expect that an additional subflow sending data will increase the aggregated bandwidth. However, we identified cases in our measurements, where an additional subflow has considerably negative impact on the aggregation benefit. For example, in the 2G+WLAN and 2G+3G scenarios, we observed that the 2G path actually hurts the overall performance whereas there was a positive aggregation benefit in the 3G+WLAN case. We will discuss our measurements and results in detail in Section VI.

We would like to also mention that there is usually a *cost* (e.g. subscription budget or CPU and consequently battery usage) to keep a subflow *active* (right side of Figure 3), however it is not trivial to define it. There might be cases where the contribution of a subflow is limited, but its cost is high. Then, the aggregation benefit should be evaluated and optimized together with this cost. In this paper, we only focus on the aggregation benefit and develop an algorithm to avoid a negative aggregation benefit. The algorithm can also be used for the positive aggregation benefit region, considering the aggregation benefit together with the cost of additional subflows. However this is beyond the scope of this paper and we leave this discussion to future work.

Figure 4 shows the aggregation of MPTCP over TCP with heterogeneous wireless paths. The scenarios with MPTCP over 2G, 3G and WLAN give some level of heterogeneity: 2G can be seen as low bandwidth and high RTT, 3G has high bandwidth and lower RTT, whereas WLAN has high bandwidth, low RTT, and a higher loss profile (compared to both, 2G and 3G).

It can be observed that 3G contributes on average by up

Algorithm 2 Per-Subflow Path Scoring with DRePaS

```
Initialization for all subflow i:
   sRTT_i ← ∞ ; sinflight_i ← 0 ; state_i ← active
   min_contribution ← 0.1 ; interval ← 3 s

for all subflow i do
   throughput_i ← sinflight_i/srtt_i
end for

for all subflow i do
   factor_i ← throughput_i/max(throughput_i)
   score_i ← { 1  if factor_i >= min_contribution
             { 0  otherwise
end for

for all subflow i do
   if state_i == active then
      if avg(interval, score_i) == 0 then
         state_i ← probing
      end if
   else
      if score_i == 1 then
         state_i ← active
      end if
   end if
end for
```

to +30% to the WLAN path, while 2G hurts the WLAN performance up to -25%. Thus, we would like to address the question of keeping a second subflow active in Section V, based on our observations from the measurements.[3]

For 2G+WLAN, we give an example that illustrates the performance with and without the 2G path in Figure 5. In this example, for the first 60 seconds, we keep both 2G and WLAN active, and then deactivate the 2G for the remaining of the experiment and observe the results. We observe that the 2G path has extreme large buffers, which causes *bufferbloat* with bulky flows. MPTCP attempts to bring the rate down on the 2G path with Algorithm 1 because of the RTT difference between 2G and WLAN. It does not succeed, since the 2G path is still in slow-start phase and the WLAN is limited by its congestion window due to losses. Then, the 2G path increases its congestion window again very rapidly. Figure 5(a) shows the 2G path congestion window and Figure 5(c) shows the resulting goodput of the 120 s transfer. One can clearly see that the 2G path is hurting the WLAN's performance in terms of goodput volume and jitter (0 s to 60 s). This example motivates our work in Section V to continuously evaluate individual subflows in the presence of path heterogeneity.

V. DRePaS – Dynamic Relative Path Scoring

In order to overcome the multi-path transport limitations observed in Section IV, we propose our Dynamic Relative Path Scoring algorithm (DRePaS) that dynamically scores the paths relative to the best path as presented in Algorithm 2. DRePaS first calculates the throughput, throughput_i, that is defined as the amount of inflight data divided by the smoothed RTT calculated every 100 ms for each subflow i. The smoothed inflight data, sinflight_i, is averaged over the past 0.5 s. We can justify the choice of smoothing inflight because of its fluctuation when segments are cumulatively acknowledged. Also, sinflight_i reflects the behavior of the connection more dynamically in contrast to the congestion window: For example, when application-limited, the congestion window does not represent the network state, e.g., cwnd $>$ 0 whereas inflight = 0 [24].

DRePaS then computes the factor_i, which is the ratio of the throughput to the maximal throughput. factor_i is a value between 0 and 1 and it is compared with a predefined threshold to determine the score; it is either 1 or 0. In the last loop, subflows are either put in the probing state or activated *conservatively*. In probing state, no payload is scheduled. However, the subflow remains established for redundancy purposes and the probing traffic is sent to evaluate the subflow, e.g., if its QoS characteristics improve. The probe data can be dynamically adapted via `sysctl`. In our experiments, we used 10 packets of 1 KiB each. Moreover, whenever the *best* subflow's performance decreases, the *probing* subflow may get resumed (due to its now relatively beneficial contribution) to ensure resilience. This complies with default MPTCP.

DRePaS has two parameters: interval and $\text{min}_{\text{contribution}}$. We set interval=3 s in order to avoid wrong decisions on short performance fluctuations[4]. Based on our observations, $\text{min}_{\text{contribution}}$ is set to 0.1. It remains open whether $\text{min}_{\text{contribution}}$ can be inferred dynamically during the transfer.

In this work, our goal is to show the potential of inferring the status of multiple connections to the same client from the server, without client support and high deployment costs. Moreover, if the client has support to identify its interfaces (as it is the case for smartphones), this information can be integrated to the algorithm. However, this might only help if the connection is not behind a router.

VI. Evaluation

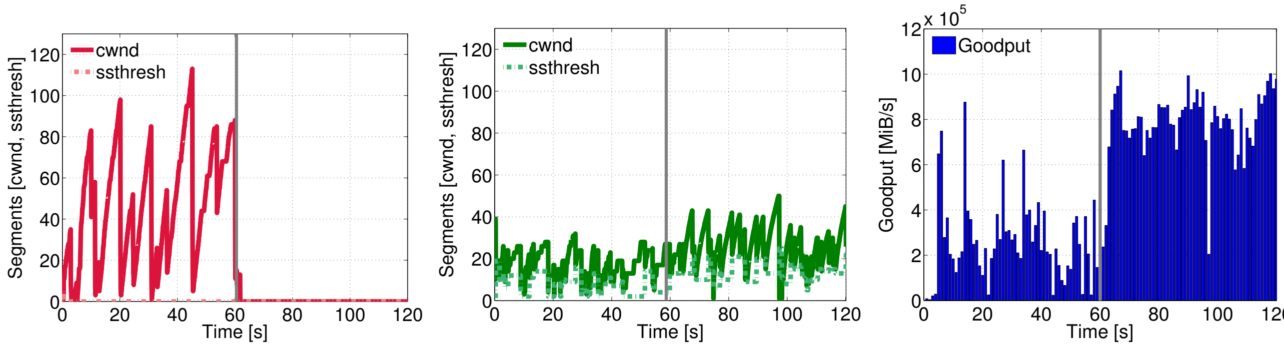
We measure MPTCP and DRePaS downloading a large file (32 MiB) with 3G+WLAN, 2G+WLAN, and 2G+3G. In our setup, MPTCP opens the first subflow on WLAN for 3G+WLAN and 2G+WLAN, and first on 3G for 2G+3G.

A. Goodput

Figure 6 shows the goodput in KiB/s for 2G+WLAN, 3G+WLAN and 2G+3G scenarios. One can see that MPTCP and DRePaS perform similarly with 3G+WLAN. This is because the 3G contribution relative to WLAN is high. On the other hand, in the scenarios including 2G, DRePaS outperforms MPTCP. This reconfirms the example given in Figure 5, where 2G has shown to hurt the performance combined with WLAN. The 2G path has limited capacity and increasingly high RTT's (due to *bufferbloat*). Thus, 2G usage results in

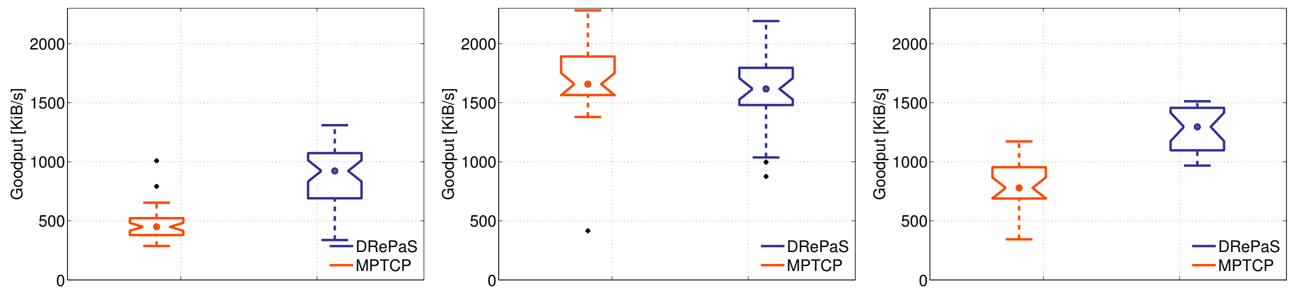
[3] Figure 4 only illustrates the aggregated goodput of MPTCP relative to single-path TCP with WLAN, and it is not the *bandwidth aggregation benefit* metric itself defined in [23].

[4] For the given paths' RTTs, 3 s seems to be a conservative setting. However, we target long flows and want to avoid that subflows are switched on and off due to short performance changes (channel variation or congestion). The 3 s interval is not fixed, but a multiple of factor_i's interval (every 100 ms in our setup). Then, 3 s is robust to capture trends in the paths (given by factor_i).



(a) Congestion Window of the 2G Subflow (b) Congestion Window of the WLAN Subflow (c) Resulting MPTCP Goodput

Fig. 5. MPTCP over 2G+WLAN (2G used for payload from 0 s to 60 s only)



(a) 2G + WLAN (b) 3G + WLAN (c) 2G + 3G

Fig. 6. Average Goodput with default MPTCP and DRePaS

receive-window limitation, and MPTCP tries to reduce its rate by cutting down the congestion window. Moreover, the 2G path increases its congestion window in slow-start rapidly due to the bad performance of the WLAN path, and thus receive-window limitation and HOL blocking will reoccur.

B. *Completion Time and Application Delay*

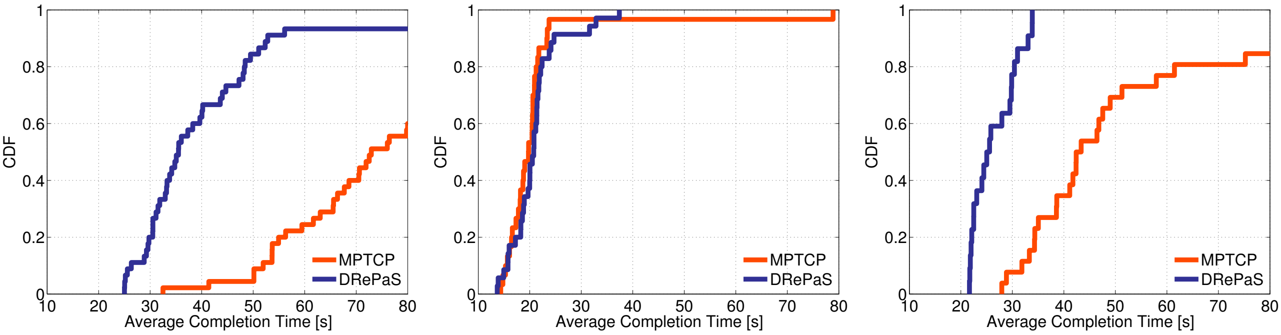
Figure 7 shows the average completion time CDF in seconds for the 2G+WLAN 7(a), 3G+WLAN 7(b) and 2G+3G 7(c) scenarios. In 3G+WLAN (i.e. 7(b)) MPTCP and DRePaS perform similarly. However, in both scenarios with the 2G path, MPTCP needs considerably more time. For example, 2G+WLAN (i.e. 7(a)) shows additional 40 s compared to DRePaS for 60% of the cases, whereas 2G+3G (i.e. 7(c)) shows 20 s for 80% of the cases. This is due to HOL blocking, i.e., the WLAN path waiting for segments on the 2G path.

Related to the application, Figure 8 shows the application delay mean (solid lines) and max (dashed lines) for 2G+WLAN 8(a), 3G+WLAN 8(b), and 2G+3G 8(c) scenarios. We define application delay as the time difference between the application sending data and the peer application reading it. One can see that the mean application delay with MPTCP in 8(a) is approximately 0.8 s higher for over 85% of the cases, whereas it is slightly higher (approximately 0.3 s) for over 80% of the cases in 8(c). Here, it is visible that the interaction between a path with low RTTs (average of 50 ms in WLAN) and a path with high RTTs (average of 1.4 s due to *bufferbloat* in 2G) considerably affects application performance. The impact is less visible in 2G+3G, since 3G has also shown high RTTs (caused by *bufferbloat*). The application delay is mainly due to in-order delivery.

Also, in 2G+WLAN, the minimum buffer size (for a retransmission timeout of 1 s) is 1052 KiB, whereas for 2G+3G the minimum is 1023 KiB. However, the measurements returned an average buffer size of more than 1900 KiB in both scenarios. Thus, delay differences, aggravated by *bufferbloat*, stress the requirements for the end systems.

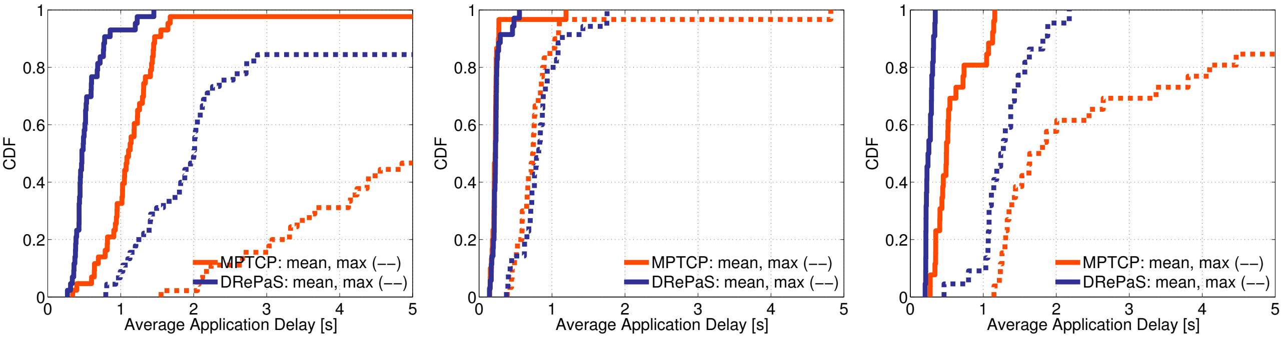
C. *The Contribution of a Subflow*

Figure 9 shows the subflow's relative contribution for the 2G/WLAN 9(a), 3G/WLAN 9(b) and 2G/3G 9(c) scenarios. Here, we present the average contribution of a subflow relative to the best subflow. Figure 9 shows that DRePaS does not completely suspend the 2G path. Whenever the WLAN's performance drops (at least for 3 s consecutively), see Algorithm 2, the 2G path will be used for payload data again. On the other hand, MPTCP continuously uses the 2G path, regardless of its behaviour, increasing HOL at the receiver, see Figures 7(a) and 8(a). In 9(b), MPTCP and DRePaS performed similarly, although we observed cases where the WLAN path was probing. Finally, 9(c) shows that the 2G path contributes little to 3G, confirming our observations in Figures 7(c) and 8(c), where the effect of HOL blocking was less visible.



(a) 2G + WLAN (b) 3G + WLAN (c) 2G + 3G

Fig. 7. Average Completion Time with Default MPTCP and DRePaS



(a) 2G + WLAN (b) 3G + WLAN (c) 2G + 3G

Fig. 8. Average Application Delay with Default MPTCP and DRePaS

	Contribution			Completion Time		
	2G/WLAN	2G/3G	3G/WLAN	2G+WLAN	2G+3G	3G/WLAN
DRePaS	0.06	0.021	0.900	40.4 s	26.4 s	20.9
MPTCP	0.27	0.074	0.857	74.8 s	50.2 s	21.1

TABLE I
MEAN RELATIVE PATH CONTRIBUTION AND COMPLETION TIME

	2G + WLAN	2G + 3G
DRePaS	159 KiB	76 KiB

TABLE II
PROBE AMOUNT

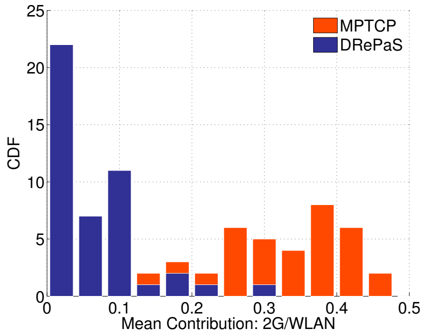
Finally, Table I shows the average contribution and completion times for all scenarios. One can see that in the scenarios including the 2G path, the DRePaS completion time was considerably lower compared to MPTCP. While MPTCP always keeps the 2G path active, DRePaS continuously evaluates it relatively to WLAN and 3G, and it keeps 2G active only when its throughput exceeds 10% of the other subflow, see Algorithm 2.

D. Probe Data relative to Payload Data

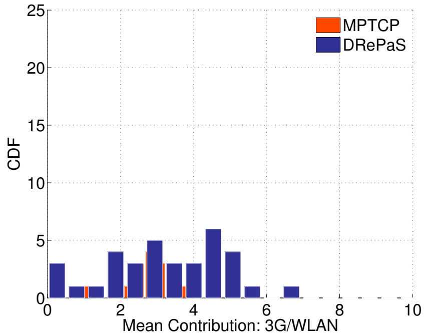
Probes do not represent actual payload data, but they are necessary to evaluate suspended subflows for resilience. Thus, they represent additional overhead. Table II shows the average amount of probe data sent by the 2G path throughout the transfer in 2G+WLAN and 2G+3G scenarios. The probes are static and set to 10 KiB data every 3 s. Note that the 3G+WLAN scenario is not mentioned because the amount of probe data is relatively small (much less than 1 KiB on average). Although we observed the WLAN temporarily probing in very fews cases, the WLAN path was mostly actively sending payload data. That is, compared to the total payload size of 32 MiB, the average overhead for DRePaS (159 KiB for 2G+WAN or 76 KiB for 2G+3G) is quite small, but the achieved performance improvement is significant.

VII. CONCLUSIONS AND FUTURE WORK

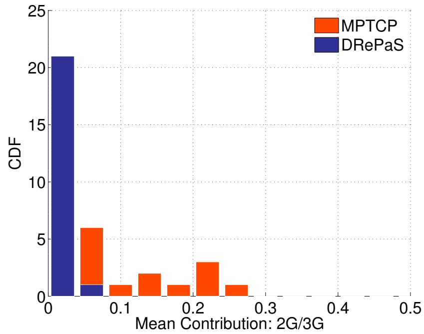
Multi-path transport is becoming increasingly important and there are many devices – like e.g. smartphones – that actually already have multiple network connections. However, as we have shown, the usage of multi-path transport is not always beneficial under realistic conditions and parameter settings, e.g. 2G and WLAN. Therefore, we have developed the Dynamic Relative Path Scoring (DRePaS) to continuously evaluate the contribution of each path to the overall performance and dynamically adapt the scheduling accordingly. In our evaluations, we have shown that DRePaS outperforms the current MPTCP implementation especially when the paths are



(a) 2G / WLAN



(b) 3G / WLAN



(c) 2G / 3G

Fig. 9. Average Contribution with default MPTCP and DRePaS

heterogeneous.

As part of our future work, we are going to evaluate DRePaS in large-scale measurements using the NorNet Edge testbed, covering many more locations. Furthermore, we are going to combine DRePaS with bufferbloat mitigation ideas to even make better use of the available, limited buffer spaces at sender and receiver. Also, we intend evaluations in mobility scenarios, with NorNet Edge nodes on buses and trains. Our results will be contributed to the ongoing IETF standardization processes of CMT-SCTP and MPTCP as well, in order to finally make multi-path transport "just work" for the users, regardless of the underlying networks' characteristics.

References

[1] M. C. Chan and R. Ramjee, "TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation," in *Proceedings of the 8th International Conference on Mobile Computing and Networking (MobiCom)*, Atlanta, Georgia/U.S.A., Sept. 2002.

[2] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 32, Oct. 2002.

[3] 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)*, San Jose, California/U.S.A., Apr. 2012.

[4] 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, Mar. 2012.

[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)*, Miami, Florida/U.S.A., Dec. 2010.

[6] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A Transport Layer Approach for Improving End-to-End Performance and Robustness using Redundant Paths," in *In Proceedings of the USENIX Annual Technical Conference*, Boston, Massachusetts/U.S.A., June 2004.

[7] S. Barré, C. Paasch, and O. Bonaventure, "MultiPath TCP: From Theory to Practice," in *Proceedings of the 10th International IFIP Networking Conference*, Valencia/Spain, May 2011.

[8] 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*, Toronto, Ontario/Canada, Aug. 2011.

[9] M. Scharf and A. Ford, "Multipath TCP (MPTCP) Application Interface Considerations," IETF, Informational RFC 6897, Mar. 2013.

[10] 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, Oct. 2013.

[11] 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)*, Ottawa, Ontario/Canada, June 2012.

[12] 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)*, Santa Barbara, California/U.S.A., Dec. 2013.

[13] S. Ferlin-Oliveira, T. Dreibholz, and Özgü Alay, "Tackling the Challenge of Bufferbloat in Multi-Path Transport over Heterogeneous Wireless Networks," in *Proceedings of the IEEE/ACM International Symposium on Quality of Service (IWQoS)*, Hong Kong/People's Republic of China, May 2014.

[14] S. Ferlin-Oliveira, T. Dreibholz, Özgü 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)*, Victoria, British Columbia/Canada, May 2014.

[15] Y.-C. Chen, Y. Lim, E. M. Nahum, R. J. Gibbens, and D. Towsley, "Characterizing 4G and 3G Networks: Supporting Mobility with Multi-Path TCP," University of Massachusetts, Amherst, Massachusetts/U.S.A., Tech. Rep. UM-CS-2012-022, Sept. 2012.

[16] Y.-C. Chen, E. M. Nahum, R. J. Gibbens, and D. Towsley, "Measuring Cellular Networks: Characterizing 3G, 4G, and Path Diversity," International Technology Alliance, Tech. Rep., June 2012.

[17] 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)*, Barcelona/Spain, Oct. 2013.

[18] 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)*, San Francisco, California/U.S.A., Nov. 2006.

[19] 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)*, Aug. 2012.

[20] 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, Mar. 2014.

[21] 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)*, Boston, Massachusetts/U.S.A., Nov. 2012.

[22] J. Mo, R. J. La, V. Anantharam, and J. Walrand, "Analysis and comparison of TCP Reno and Vegas," in *Proceedings of the 18th IEEE INFOCOM*, vol. 3, Mar. 1999.

[23] D. Kaspar, "Multipath Aggregation of Heterogeneous Access Networks," Ph.D. dissertation, University of Oslo, Dec. 2011.

[24] A. Sathiaseelan, J. Crowcroft, M. Goulden, C. Greiffenhagen, R. Mortier, G. Fairhurst, and D. McAuley, "PAWS: Public Access WiFi Service," in *The Third Digital Economy All-hands Meeting: Digital Engagement (DE)*, Aberdeen, Scotland/United Kingdom, Oct. 2012.