The Performance Impact of Buffer Sizes for Multi-Path TCP in Internet Setups

Feng Zhou*, Thomas Dreibholz†, Xing Zhou*, Fa Fu* Yuyin Tan* and Quan Gan‡

*Hainan University, College of Information Science and Technology, Haikou, China
Email: 969493314@qq.com, {zhouxing,fufa,tanyuyin}@hainu.edu.cn

†Simula Research Laboratory, Centre for Resilient Networks and Applications, Fornebu, Norway
Email: dreibh@simula.no

‡China Unicom, Hainan Branch, Haikou, China
Email: 18608902600@wo.com.cn

***Abstract*—The Multi-Path Transmission Control Protocol (MPTCP) is the new concurrent multi-path transfer extension for the widely-deployed Transmission Control Protocol (TCP). Of course, having multiple and possibly highly dissimilar paths for transmission is a challenge for the management of the send and receive buffers, since optimal throughput is desired with a reasonable allocation of the limited memory resources in MPTCP endpoints. This is particularly important when many MPTCP connections have to be handled simultaneously.**

This paper measures out the required MPTCP buffer size in the real-world Internet testbed NORNET, comparing theoretical size and real size to analyse MPTCP performance. The experiment shows that multi-path transmission can effectively increase the application payload throughput, and greatly improve the robustness of the data transmission. As an important point of this paper, we can show that appropriate buffer size settings can increase the payload throughput, while not wasting resources. This paper has certain significance for further accurately determining the optimal buffer size settings for multi-path transmission in large-scale Internet setups.

Keywords: Multi-Path Transport, Multi-Path TCP (MPTCP), Buffer Size, Throughput, Robustness

I. INTRODUCTION

With the development of new network access technologies, many devices today provide multiple network interfaces. For example, each modern smartphone has at least one mobile broadband interface (i.e. 2G/3G/4G) and Wi-Fi. However, the Transmission Control Protocol (TCP) [1] can only use a single path. For example, if a smartphone has 4G and Wi-Fi, when 4G is broken, the connection will be interrupted even though Wi-Fi is turned on. By using the Multi-Path TCP (MPTCP) [2], [3] extension, this issue can be solved by the support of *multi-homing*, i.e. by using multiple addresses simultaneously. Particularly, this property makes MPTCP also interesting for data centre communications [4].

As shown in Figure 1, MPTCP uses multiple subflows to implement multi-homing and possibly concurrent multi-path transport. Each subflow is defined by a source/destination IP address pair, i.e. an MPTCP connection may simultaneously use IPv4 and IPv6 address pairs. On the wire, each subflow appears like a regular TCP connection, i.e. MPTCP-unaware middleboxes [5] in the network should handle subflows like TCP connections.

[1]This work has been funded by the National Natural Science Foundation of China (funding numbers 61163014 and 61363008) as well as the International Cooperation Projects of Hainan (funding number KJHZ2013-20).

[2]Xing Zhou is corresponding author.

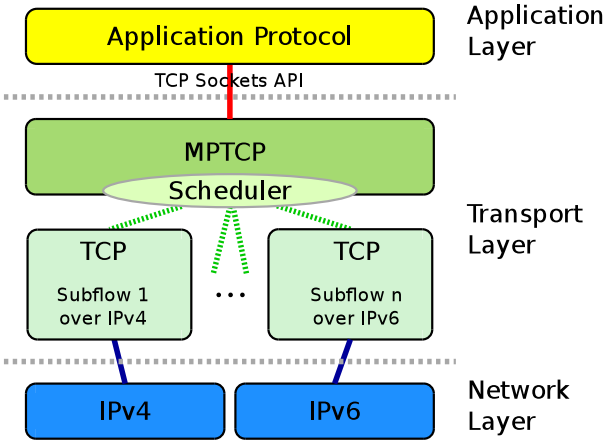


Figure 1. The Architecture of MPTCP

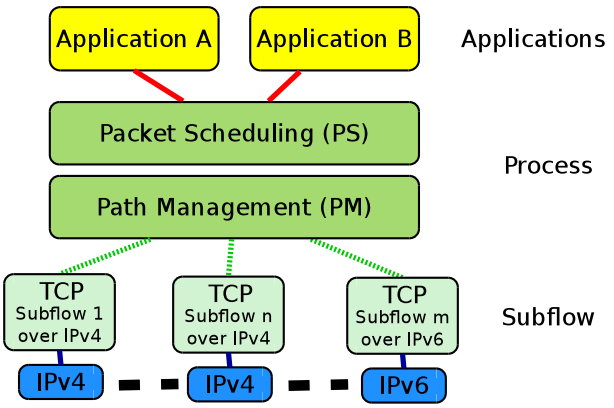
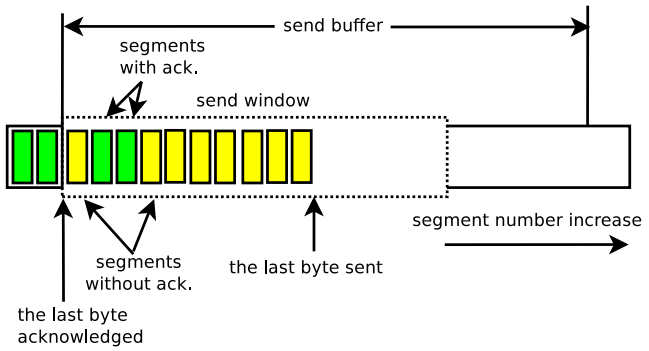
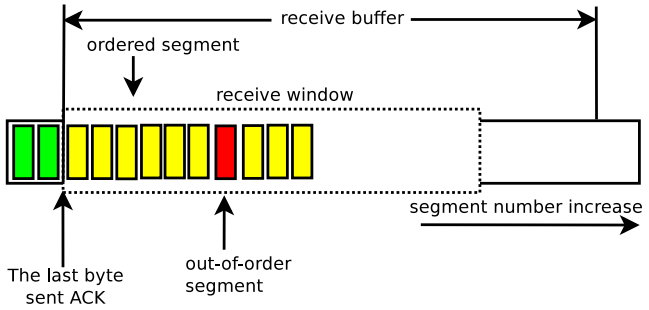


Figure 2. The Functions of MPTCP

MPTCP has mainly two functions [6], as depicted in Figure 2: the first one is Packet Scheduling (PS), which is linked with data scheduling, interfacing with the subflow, and congestion control. The second one is Path Management (PM) [7]: managing the communication paths, i.e. subflows, between the two MPTCP instances. PS is processing and sending the data from the Application Layer; it transmits them via a subflow, and adds subflow sequence numbers and confirmation numbers into data segments, before handing them to the Network



(a) Send Buffer



(b) Receive Buffer

Figure 3. The Usage of Send and Receive Buffers by TCP

Layer. When the peer side receives a subflow segment, it sends the data to the PS and hands it to the Application Layer. The PM manages the paths between the sender and the receiver. Paths can be added to, or removed from, a running MPTCP connection, allowing to dynamically adapt to changing connectivity conditions.

II. Buffer Handling for TCP

TCP uses window-based flow and congestion control, as depicted in Figure 3; Subfigure 3(a) shows the sender side (i.e. the send buffer) and Subfigure 3(b) depicts the receiver side (i.e. the receive buffer). Data that is being transmitted – also denoted as outstanding or in flight – is stored in the send buffer. The corresponding data segments have been sent, but are not yet acknowledged by the receiver side. That is, they may still be in transmission, or the segment or its acknowledgement has been lost. Since there may be a need for a retransmission (RTX), the data needs to be remained in the send buffer until it is finally acknowledged by a cumulative acknowledgement. A cumulative acknowledgement means that all data until a given segment has been received by the peer side. In this case, and only in this case, it can be removed safely from the send buffer.

The receiver side stores the received segments in the receive buffer. Obviously, when all segments have been received in their correct sequence, they can be forwarded to the Application Layer (since the data is complete and in the right order). Then, all segments until the last one can be acknowledged by a cumulative acknowledgement. However, sometimes, reordering occurs in the network. That is, there may e.g. be gaps in the segment sequence due to packet losses. Then, segments denoted as "out-of-order segments" can also be found in the receive buffer, waiting for their preceding segments to arrive. Modern TCP implementations [8] acknowledge such segments by so-called selective acknowledgements (SACKs). Once the missing segments have arrived, and all data is in the right order, a cumulative acknowledgement for the whole segment range can be generated, and the data is passed to the Application Layer (i.e. leaving the receive buffer).

Modern TCP implementations [8] apply fast retransmission, i.e. once a segment is seen as missing (by analysing the incoming acknowledgements) for 3 times, it is immediately scheduled for retransmission. Further retransmissions are scheduled by the retransmission timer set to a dynamically configured retransmission timeout (RTO; usually at least 1 s). While fast retransmissions are frequent (due to the network's feedback on congestion by packet losses), timer-based retransmissions should be rare (they are usually a sign of severe network congestion). Clearly, in the ideal case, a segment is successfully transmitted and acknowledged. In this case, it takes only one round-trip time (RTT) to receive the acknowledgement. If there is a fast retransmission involved, this increases to 2*RTT. Therefore, in order to utilise a network path, the send/receive window size constraint B is:

$$B \geq \mathrm{RTT} * \mathrm{Bandwidth}$$

But, in order to cover a fast retransmission (since they occur frequently), the constraint increases to:

$$B \geq 2 * \mathrm{RTT} * \mathrm{Bandwidth}$$

Further, covering even a timer-based retransmission, the constraint is:

$$B \geq (3 * \mathrm{RTT} + \mathrm{RTO}) * \mathrm{Bandwidth}$$

Congestion control dynamically limits the configured window size, in order to avoid network overload. A detailed introduction on window-based flow and congestion control can be found in [9, Chapter 2]. Particularly, [9, Subsubsection 2.9.2.3] provides details on the window size constraints.

III. Buffer Handling for MPTCP

Clearly, when using multi-path transport – here with MPTCP – the buffer handling becomes challenging. Note, that these challenges are generic and apply to other multi-path transport protocols, particularly to Concurrent Multipath Transfer for the Stream Control Transmission Protocol (CMT-SCTP) [9]–[11] as well.

For MPTCP (and also for CMT-SCTP), send and receive buffers are shared among all subflows. When path characteristics (i.e. bandwidth, delay, loss rate and error rate) become dissimilar – which is very likely when using the Internet – blocking issues can occur. That is, as shown by [9], [12], some low-performance subflows may occupy a major share of the buffers, leaving no room to fully utilise other subflows. Mechanisms like buffer splitting [13], non-renegable selective acknowledgements [14], [15], chunk rescheduling [9], [13], opportunistic retransmission [16], buffer bloat mitigation [17] and smart scheduling decisions [18]–[21] are necessary to avoid these issues. However, in any case, the buffers must be large enough to cope with the *maximum* RTT of any of the subflows. That is, the constraints from Section II extend to

$$B \geq \max_{1 \leq i \leq n} \{\mathrm{RTT}_i\} * \sum_{i=1}^{n} \mathrm{Bandwidth}_i \qquad (1)$$

for RTT_i the RTT and $\mathrm{Bandwidth}_i$ the bandwidth of subflow i. And, considering a fast retransmission, it adapts to:

$$B \geq 2 * \left(\max_{1 \leq i \leq n} \{\mathrm{RTT}_i\} * \sum_{i=1}^{n} \mathrm{Bandwidth}_i \right). \quad (2)$$

Considering even a timer-based retransmission, the buffer size requirement is:

$$B \geq \left(3 * \max_{1 \leq i \leq n} \{\mathrm{RTT}_i\} + \max_{1 \leq i \leq n} \{\mathrm{RTO}_i\} \right) * \sum_{i=1}^{n} \mathrm{Bandwidth}_i. \quad (3)$$

That is, in the worst case, it takes three times the highest subflow RTT (first transmission, fast retransmission, timer-based retransmission) plus the highest subflow RTO.

Here, it is useful to visualise the values with a real-world example of two dissimilar Internet paths:

1) High-speed fiber, 100 Mbit/s, 10 ms RTT, 1 s RTO.
2) Asymmetric Digital Subscriber Line, 1 Mbit/s uplink, 200 ms RTT (i.e. light buffer bloat [12], [22]), 1 s RTO.

With these settings, the buffer sizes are:

- 2466 KiB (only first transmission, Equation 1)
- 4932 KiB (including fast retransmission, Equation 2)
- 19727 KiB (including timer-based retransmission, Equation 3)

That is, once timer-based retransmissions are included in the calculation, the buffer size requirements become inconveniently large. In this example, it is ca. 20 MiB *per connection*. Considering a server with hundreds, thousands or even more simultaneous connections, this becomes costly and inefficient.

Obviously, the question is: How much buffer space is needed in realistic Internet setups? In the following, we will analyse such real-world setups.

IV. Measurement Setup

A. The NorNet Core Testbed

The NorNet [23]–[25] testbed[1] is the world's first, open, large-scale Internet testbed for multi-homed systems and applications. Its wired network part is denoted as NorNet Core [26]–[28]. A unique characteristic of NorNet Core is that each site is multi-homed to several ISPs. Particularly, it is currently used for research on topics like multi-path transport and resilience. Researchers can run experiments on distributed, programmable nodes which spread over four continents (Europe, Asia, Australia, America) and are connected to multiple different ISPs with different access technologies. Clearly, a key feature of NorNet Core is to work in the real-world Internet.

The information for the NorNet Core sites [23], [29], [30] can be found in Table I. High-speed ISP connections are shown in green colour, while slow-speed connections (up to 16 Mbit/s, in many cases ADSL connections – marked with "A") are shown in yellow colour. An illustration of the sites, as well as their connectivity based on Traceroute observations [27], [31] over four weeks (May 16 to June 13, 2016), is presented in Figure 4. Different autonomous systems (AS) of the links' routers are represented by divergent colours. Obviously, there is a significant variation of paths, motivating the usage of multi-path transport to utilise this property for throughput improvements.

[1] NorNet: https://www.nntb.no.

B. Measurement Tools

For our experiments, we used Ping and NetPerfMeter.

1) Ping: Ping is the well-known Unix command to test Internet connectivity by Internet Control Message Protocol (ICMP) Echo Requests and Echo Replies [32]. We used this tool to record the RTTs of the paths during the bandwidth measurements.

2) NetPerfMeter: The bandwidth measurements have been performed by applying the NetPerfMeter [9], [33], [34] tool. It provides the performance comparison of multiple transport connections and protocols. Particularly, MPTCP is supported by NetPerfMeter as well [26], [33], [35]. Furthermore, it supports configuring the send and receive buffer sizes for a connection by using the SO_SNDBUF and SO_RCVBUF socket options. Note, that kernels also need to take buffer management overhead into account. While FreeBSD treats the SO_SNDBUF/SO_RCVBUF values just "as hints", Linux simply doubles the given values[2].

All results have been processed with GNU R [36]. Results plots show the average application payload throughput for a saturated NetPerfMeter flow running 60 s, together with the corresponding 95% confidence intervals.

C. Scenarios Parameters

In all measurement scenarios, we have used the following Linux kernel setup:

(a) Linux kernel version 4.1.27,
(b) Linux MPTCP [16] version 0.91[3] using the "fullmesh" path manager (to use all possible paths [37]), and
(c) Cubic [38] congestion control (the Linux default; since the ISPs are independent) with Explicit Congestion Notification (ECN) support [39], [40] enabled.
(d) The TCP (and MPTCP) buffer size limit (also for SO_SNDBUF and SO_RCVBUF settings) is 16 MiB[4].

V. Results Analyses

For this paper, we have selected four different scenarios.

A. Challenging Inter-Continental Multi-Homing Scenario

In the first scenario, we analyse the communication between the Universität Duisburg-Essen (UDE) site in Germany and the Hainan University (HU) site in China. As shown in Table I, both sites are connected to two ISPs each, with one of the ISPs in Germany (Versatel) being an ADSL provider and one of the ISPs in China a consumer-grade fibre connection (CnUnicom). The other ISPs (DFN and CERNET) are the national research network ISPs. This dissimilar scenario – with four different paths – is therefore quite challenging for multi-path transport [37]. Figure 5 (UDE→HU) and Figure 6 (HU→UDE) present the average application payload throughput (over 20 runs) for varying send/receive buffer sizes (same value for both buffers). TCP results are presented in the left-hand subfigure, MPTCP results in the right-hand subfigure. Each plot shows the results for each ISP combination used for establishing the TCP connection (TCP) or for establishing the first subflow (MPTCP). For better readability, different destination ISPs use different colours (red and blue), while different source ISPs use different line styles (solid and

[2] In `net/core/sock.c` of the Linux kernel sources.
[3] Available from http://www.multipath-tcp.org.
[4] `sysctl`: net.ipv4.tcp_rmem, net.ipv4.tcp_wmem, net.ipv4.tcp_mem.

Index	Site	Abbreviation	Location (Province, Country)	ISP 1	ISP 2	ISP 3
3	Høgskolen i Gjøvik	HiG	Oppland, Norway	Uninett	PowerTech[A]	
5	Universitetet i Stavanger	UiS	Rogaland, Norway	Uninett	Altibox	PowerTech[A]
6	Universitetet i Bergen	UiB	Hordaland, Norway	Uninett	BKK	
9	Universitetet i Trondheim	NTNU	Sør-Trøndelag, Norway	Uninett	PowerTech[A]	
10	Høgskolen i Narvik	HiN	Nordland, Norway	Uninett	Broadnet[A]	PowerTech[A]
42	Universität Duisburg-Essen	UDE	Nordrhein-Westfalen, Germany	DFN	Versatel[A]	
88	Hainan University	HU	Hainan, China	CERNET	China Unicom	
100	The University of Kansas	KU	Kansas, United States	KanREN		

Table I
THE NORNET CORE TESTBED SITES USED FOR THE MEASUREMENTS

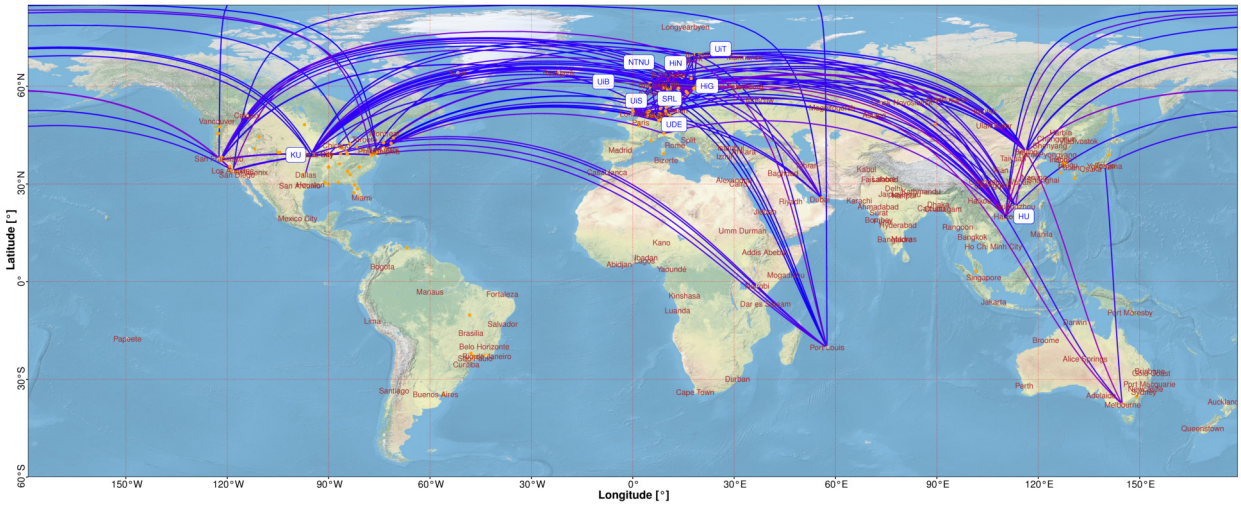
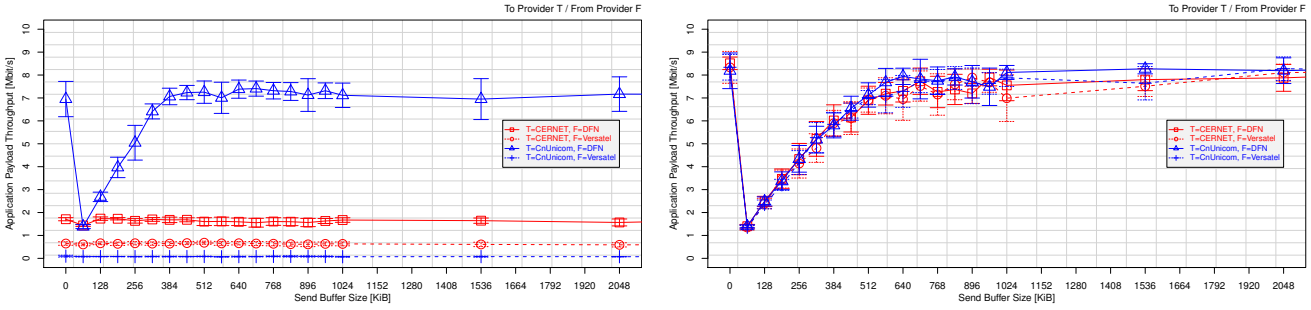
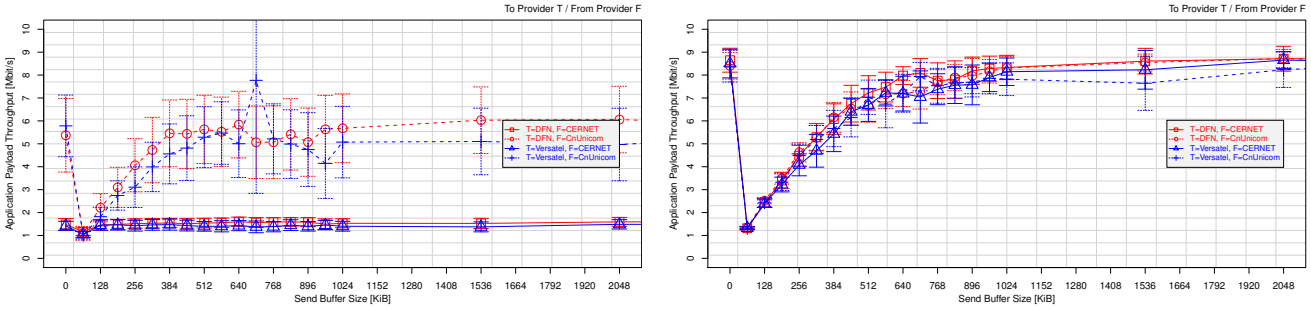


Figure 4. The NORNET CORE Sites and their Connectivity in the Internet



(a) TCP (b) MPTCP

Figure 5. Universität Duisburg-Essen (UDE) → Hainan University (HU) – Average over 20 Runs



(a) TCP (b) MPTCP

Figure 6. Hainan University (HU) → Universität Duisburg-Essen (UDE) – Average over 20 Runs

dashed). The buffer size of 0 KiB has a special meaning: the size is not explicitly set by SO_SNDBUF/SO_RCVBUF, and the kernel automatically adapts the value up to the limit of 16 MiB (see Subsection IV-C).

Having a look at the TCP results (Subfigure 5(a) and 6(a)), it is obvious that the application payload throughput performance highly depends on the choice of ISP combination. For example, using DFN→CERNET (for UDE→HU in Subfigure 5(a)) achieves about 7.34 Mbit/s, while any other combination not even reaches 2.1 Mbit/s. In the worst case, i.e. slow ADSL uplink (Versatel→CnUnicom), it even reduces to just a couple of Kbit/s. The reverse direction is similar (Subfigure 6(a)), although the faster ADSL downlink performs better. Nevertheless, in any case, it is crucial for the application – or even the user – to choose the right ISP combination for TCP connections, in order to experience a good performance. From the buffer size perspective, 512 KiB are enough to utilise the paths (detailed analysis follows in Subsection V-C).

Clearly, the setup is interesting for multi-path transport. First of all, the application – or even the user – should not need to know about path details. As shown by the MPTCP results (Subfigure 5(b) and 6(b)), this is clearly the case. Regardless of the chosen initial path for establishing the first subflow, the resulting application payload throughput is almost similar. Furthermore, it is even better than for the best TCP path (i.e. one of the goals of multi-path transport). It is just slightly lower for initial paths over ADSL, due to the fact that the higher RTT of this path (detailed analysis in Subsection V-C) results in a slightly longer time to establish all subflows, inflate the congestion windows, and finally utilise all paths.

Furthermore, from the buffer size perspective, moderate settings between 640 KiB and 1024 KiB are already sufficient to get the full performance. We will analyse the size in detail in Subsection V-C. That is, MPTCP can achieve a good performance, in a challenging scenario, with reasonably small buffer size settings. But what about other, somewhat extreme, Internet scenarios?

B. Some Extreme Scenarios

In the second scenario, we analyse the paths from The University of Kansas (KU) in the United States to Høgskolen i Gjøvik (HiG) in Norway (see Table I). The KU site is just single-homed, using the local research network ISP (KanREN). On the other side, HiG is connected to the Norwegian research network ISP (UNINETT) and an ADSL provider (PowerTech). Clearly, as shown in the results in Figure 7, the TCP application payload throughput (Subfigure 7(a)) extremely depends on the chosen path: high-speed (KanREN→UNINETT with ca. 58.9 Mbit/s) or slow-speed (KanREN→PowerTech with about 1.87 Mbit/s). Due to the high dissimilarity in the path bandwidths, and the quite slow ADSL path, MPTCP is not able to provide a better throughput than TCP over the best path. Instead, it is smaller (about 53.5 Mbit/s). However, it is again almost independent of the chosen path for the initial subflow. That is, the application – or the user – does not need knowledge about the paths. Also, with a buffer size of about 2560 KiB, the full performance is already reached. This is similar to the size needed for TCP.

To examine the effect of additional slower-speed ISPs in more detail, the third scenario examines the throughput between NTNU Trondheim (NTNU) and Høgskolen i Narvik (HiN). Both sites, being located in Norway, are connected to the research network ISP (UNINETT). Furthermore (see Table I), NTNU has one additional ADSL ISP (PowerTech), while HiN even has two (PowerTech and Broadnet). As shown by the results in Figure 8, the optimal TCP path choice is UNINETT→UNINETT with about 91.69 Mbit/s application payload throughput (Subfigure 8(a)). Uninett→Broadnet reaches about 14.05 Mbit/s, while the other 4 choices perform significantly worse. So, an application – or user – without path knowledge has just a 1-in-6 chance to select the right path. MPTCP, on the other hand, solves this issue (Subfigure 8(a)): while its about 78.12 Mbit/s do not fully reach the 91.69 Mbit/s of the best TCP path, it is *significantly* better than all 5 other choices. Note, that due to the low RTT between the sites in Norway (detailed analysis in Subsection V-C), the difference among different MPTCP choices for the initial subflow becomes very small. With respect to buffer size, the reasonable MPTCP performance and convenience comes with a price: the 6 highly dissimilar paths cause packed reordering, requiring about 4096 KiB of buffer space for MPTCP, while TCP on the best path just needs about 256 KiB.

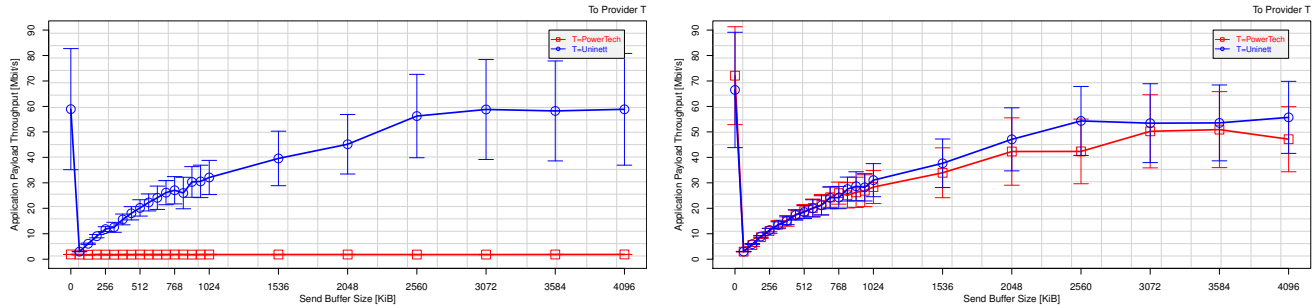
Finally, for our last scenario, we examine the effect of multiple high-speed ISP connections, by observing the throughput between Universitetet i Bergen (UiB) and Universitetet i Stavanger (UiS). Again, both of these sites in Norway are connected to the research network ISP (UNINETT). In addition (see Table I), the UiB site has an additional, business-grade fibre connection (BKK, symmetric 100 Mbit/s). The UiS site is connected to a consumer fibre ISP (Altibox) and a DSL provider (PowerTech). Clearly, the UNINETT research network between UiB and UiS is very fast: about 230.91 Mbit/s application payload throughput can be achieved with TCP (Subfigure 9(a)). Using the somewhat slower BKK, TCP can reach about 90.60 Mbit/s. All combinations with the consumer-grade ISP are much slower. MPTCP (Subfigure 9(b)) cannot reach the 230.91 Mbit/s. However, with any ISP combination used for the initial subflow, it still achieves a throughput of about 150.47 Mbit/s – which is much better than the 5 other ISP combinations for TCP. Furthermore, due to the lower fibre RTTs, there is not much difference between the choices of the initial subflow path. The low fibre RTTs (just PowerTech at UiS is ADSL) also lead to a low buffer space requirement: about 2048 KiB are sufficient in this scenario.

In summary, MPTCP performs reasonably well in challenging Internet setups – even in quite extreme scenarios. While MPTCP not always achieves the performance of the optimal TCP path, it prevents really bad performance by inappropriate path choices for TCP. As part of further work, a more in-depth analysis of the observed performance, the PS decisions, as well as possibilities for improvements, are necessary. Particularly, applications and users do not need to care for paths. Furthermore, the observed buffer sizes needed for MPTCP are not overly large. So, the remaining question is: how much buffer space is really needed for certain scenarios?

C. How Much Buffer Space is Needed?

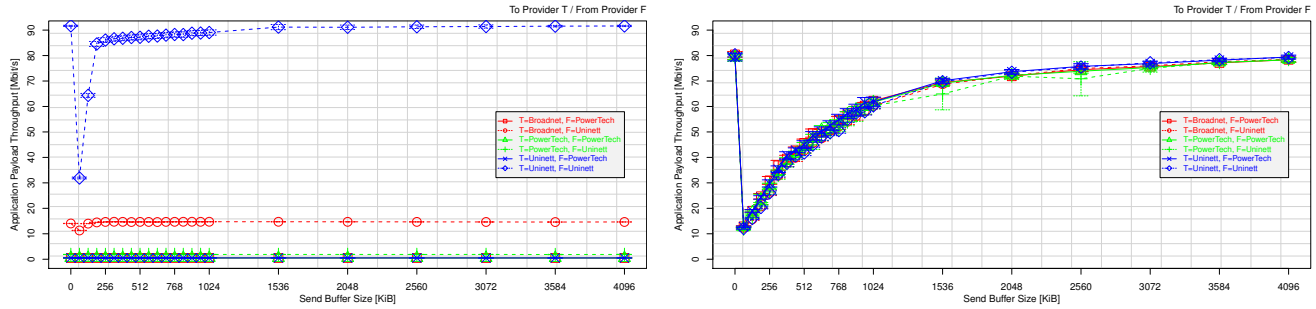
Clearly, as explained in Section III, the buffer size requirements strongly depend on the RTTs of paths. Therefore, Table II presents the RTTs (minimum, average, median, maximum; in ms) and the average application payload throughput (TP; in Mbit/s; at buffer size "0") for our four scenarios.

For the inter-continental setup between UDE (Germany) and HU (China), the RTTs are – as expected – large: at



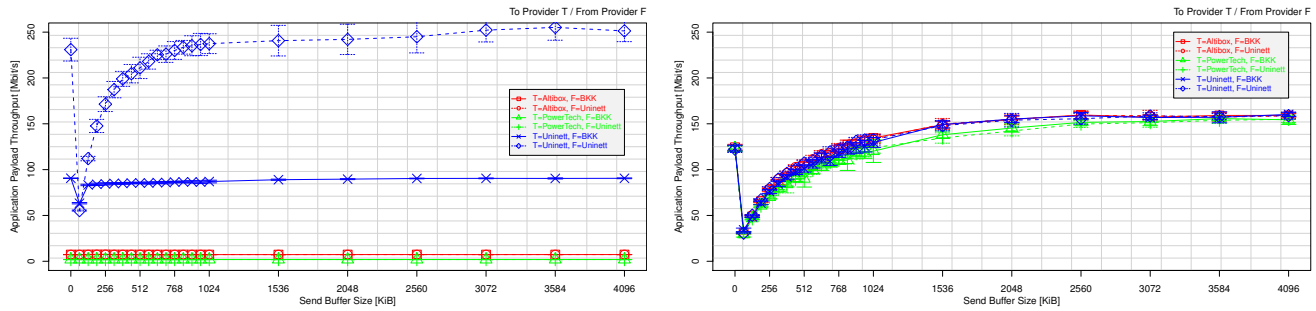
(a) TCP (b) MPTCP

Figure 7. The University of Kansas (KU) → Høgskolen i Gjøvik (HiG) – Average over 20 Runs



(a) TCP (b) MPTCP

Figure 8. NTNU Trondheim (NTNU) → Høgskolen i Narvik (HiN) – Average over 20 Runs



(a) TCP (b) MPTCP

Figure 9. Universitetet i Bergen (UiB) → Universitetet i Stavanger (UiS) – Average over 20 Runs

least 279 ms (absolute minimum) and between 351 ms and 469 ms on average. Interesting are the absolute maximum values. While some buffer bloat due to the ADSL connection (Versatel) of e.g. 1 s or 2 s can be expected [12], some peaks with more than 7 s have been observed. Particularly, the absolute maximum of the Versatel→CERNET path has been 8786 ms. While the routing of this setup is challenging (see Figure 4 – sometimes westwards via the United States, sometimes eastwards via Russia), a probable explanation of these peaks is spurious delay by deep packet inspection in firewalls. To filter out these outliers, the median RTT value is therefore a more appropriate metric for RTT comparison than the average value.

The results for the other scenarios – with smaller geographical distances – are less extreme. As expected, the RTTs are highest when ADSL connections are involved (PowerTech, Broadnet). The effect of buffer bloat is easily observable by the high absolute maximum in the range of about 1 s, and the corresponding difference between average and median value.

Based on the measured *median* RTTs, and assuming an RTO of 1 s (which is small, particularly with buffer bloat), we computed the theoretical buffer size values according to the equations in Section III: for the initial transmission (Equation 1), with a fast retransmission (Equation 2), and with a timer-based retransmission (Equation 3). Table III presents the results, together with the practically useful values from our measurements in Subsection V-A and Subsection V-B.

First, it is clearly observable that the practical buffer size does not have to cover a timer-based RTX. This is particularly important in scenarios with a mix of high-speed/low-delay and low-speed/high-delay paths (like e.g. fibre and ADSL). That is,

(a) UDE → HU

Path	Min	Avg	Med	Max	TP
DFN → CERNET	335	351	340	7726	1.71
DFN → CnUnicom	279	469	**386**	2557	6.95
Versatel → CERNET	334	355	340	8786	0.65
Versatel → CnUnicom	291	463	**386**	3010	0.10

(b) HU → UDE

Path	Min	Avg	Med	Max	TP
CERNET → DFN	335	384	340	8553	1.48
CERNET → Versatel	368	448	**412**	8505	1.42
CnUnicom → DFN	329	380	340	8791	5.37
CnUnicom → Versatel	368	468	411	8857	5.78

(c) KU → HiG

Path	Min	Avg	Med	Max	TP
KanREN → PowerTech	160	231	**177**	2435	1.87
KanREN → Uninett	142	151	145	1959	58.94

(d) NTNU → HiN

Path	Min	Avg	Med	Max	TP
PowerTech → Broadnet	32	69	49	399	0.50
PowerTech → PowerTech	38	110	**75**	1153	0.48
PowerTech → Uninett	13	45	24	273	0.52
Uninett → Broadnet	32	68	47	1025	14.05
Uninett → PowerTech	38	113	74	1148	1.89
Uninett → Uninett	13	44	22	853	91.69

(e) UiB → UiS

Path	Min	Avg	Med	Max	TP
BKK → Altibox	18	32	22	869	7.31
BKK → PowerTech	26	77	**54**	1006	1.89
BKK → Uninett	5	20	11	646	90.60
Uninett → Altibox	18	32	22	1266	7.30
Uninett → PowerTech	26	77	**54**	1036	1.89
Uninett → Uninett	5	20	11	551	230.91

Table II
MEASURED ROUND-TRIP TIMES (MS) AND AVERAGE PAYLOAD THROUGHPUT (MBIT/S) OF SINGLE-PATH TCP FLOWS

Scenario	Theoretical Size			Practical Size
	Initial Only	Fast RTX	Timer-Based RTX	
UDE → HU	443 KiB	887 KiB	2479 KiB	640 KiB
HU → UDE	707 KiB	1413 KiB	3835 KiB	1024 KiB
KU → HiG	1314 KiB	2628 KiB	11365 KiB	2536 KiB
NTNU → HiN	999 KiB	1998 KiB	16319 KiB	4096 KiB
UiB → UiS	2241 KiB	4481 KiB	48213 KiB	1024 KiB

Table III
THEORETICAL AND PRACTICAL MPTCP SEND/RECEIVE BUFFER SIZE SETTINGS (KIB)

instead of requiring tens of MiB (e.g. 48213 KiB is more than 47 MiB for UiB→UiS), values of 1024 KiB to 4096 KiB have already been sufficient for our challenging Internet setups. So, multi-path transport is in practice not overly expensive in terms of buffer space.

Furthermore, as a rough estimate, covering a fast RTX with the buffer size seems to be useful. Fast RTX occur frequently, so it is useful to assume their occurrence. For four of the five scenarios in Table III, approximating the buffer size with Equation 2 would have been sufficient. Note, that due to the implementation of the SO_SNDBUF/SO_RCVBUF socket option in Linux (as explained in Subsubsection IV-B2), the actually allocated buffer size may be twice the given size. Subtracting the necessary management overhead, the buffer size for *payload data* is somewhere between 1 and 2 times the given size.

With 4096 KiB, only the NTNU→HiN setup requires a larger buffer space than approximated with Equation 2 (1998 KiB), due to its challenging setup with 1 high-speed/low-delay fibre path plus 5 low-speed/high-delay ADSL paths. Here, particularly buffer bloat on the ADSL paths increases the RTO (which we approximated with 1 s) to sometimes 2 s and more. This leads to a performance reduction if there is insufficient buffer space to fully utilise all paths. However, such extreme cases are likely to be rare in practise.

In summary, taking a fast RTX into account when estimating the buffer size requirements (Equation 2) seems to be a reasonable approach for many scenarios. However, it cannot provide a very exact model for computing accurate sizes. Part of future work is therefore the development of a more fine-granular model for such estimations, and its validation in NORNET-based Internet setups.

VI. CONCLUSIONS

Multi-path transport in today's heterogeneous networks is challenging. Therefore, we have evaluated the performance of MPTCP in four interesting wide-area Internet scenarios in the NORNET CORE testbed. Particularly, we have made three key observations:

1) Even when facing challenges, MPTCP can achieve performance advantages over TCP.
2) MPTCP offers robust performance. Independent of the path chosen for the initial subflow, the long-term performance is similar. That is, applications (or even users) do not need knowledge about underlying paths.
3) Although there may be fear of an overly large buffer space need, due to packet reordering over very dissimilar paths, we have shown that buffer size requirements remain reasonably small. This is particularly important for systems having to manage many simultaneous connections.

As part of future work, a more fine-granular estimation of the buffer size limits vs. resulting performance is necessary. Therefore, we intend to run further Internet measurements to get more detailed, long-term data. Also, in some cases it may be useful to *not* use certain paths that may reduce the overall multi-path transport performance (e.g. ADSL path vs. high-speed fibre) for payload transport, like in [21]. This also needs more detailed examination in Internet setups.

REFERENCES

[1] J. B. Postel, "Transmission Control Protocol," IETF, Standards Track RFC 793, Sep. 1981, ISSN 2070-1721.

[2] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF, RFC 6824, Jan. 2013, ISSN 2070-1721.

[3] A. Ford, C. Raiciu, M. Handley, S. Barré, and J. R. Iyengar, "Architectural Guidelines for Multipath TCP Development," IETF, Informational RFC 6182, Mar. 2011, ISSN 2070-1721.

[4] C. Raiciu, C. Pluntke, S. Barré, A. Greenhalgh, D. Wischik, and M. Handley, "Data Center Networking with Multipath TCP," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Monterey, California/U.S.A., Oct. 2010, pp. 1–6, ISBN 978-1-4503-0409-2.

[5] E. Lopez, "Multipath TCP Middlebox Behavior," IETF, Individual Submission, Internet Draft draft-lopez-mptcp-middlebox-00, Nov. 2014.

[6] F. Fu, Z. Xing, Y. Xiong, H. Adhari, and E. P. Rathgeb, "Performance Analysis of MPTCP and CMT-SCTP Multi-Path Transport Protocols," *Computer Engineering and Applications*, vol. 49, no. 21, pp. 79–82, Oct. 2013.

[7] K. Wang, T. Dreibholz, X. Zhou, F. Fu, Y. Tan, X. Cheng, and Q. Tan, "On the Path Management of Multi-Path TCP in Internet Scenarios based on the NorNet Testbed," in *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Taipei, Taiwan/People's Republic of China, Mar. 2017.

[8] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," IETF, Standards Track RFC 5681, Sep. 2009, ISSN 2070-1721.

[9] 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.

[10] P. D. Amer, M. Becke, T. Dreibholz, N. Ekiz, J. R. Iyengar, P. Natarajan, R. R. Stewart, and M. Tüxen, "Load Sharing for the Stream Control Transmission Protocol (SCTP)," IETF, Individual Submission, Internet Draft draft-tuexen-tsvwg-sctp-multipath-13, Dec. 2016.

[11] 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, pp. 82–88, Apr. 2011, ISSN 0163-6804.

[12] 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)*, Singapore, Mar. 2011, pp. 708–714, ISBN 978-0-7695-4338-3.

[13] 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, ISBN 978-1-4244-5637-6.

[14] P. Natarajan, N. Ekiz, E. Yilmaz, P. D. Amer, and J. R. Iyengar, "Non-Renegable Selective Acknowledgments (NR-SACKs) for SCTP," in *Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP)*, Orlando, Florida/U.S.A., Oct. 2008, pp. 187–196, ISBN 978-1-4244-2506-8.

[15] Z. Deng, "Non-Renegable Selective Acknowledgements (NR-SACKs) for MPTCP," IETF, Individual Submission, Internet Draft draft-deng-mptcp-nrsack-00, Dec. 2013.

[16] 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, pp. 1–14.

[17] S. Ferlin, 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, pp. 123–128, ISBN 978-1-4799-4852-9.

[18] Q. Hu, R. Zhou, and L. Zhou, "Forward Prediction Data Scheduling Mechanism for MPTCP," *Application Research of Computers*, vol. 30, no. 2, pp. 560–561, Feb. 2013, ISSN 1001-3695.

[19] Qing Hu and Zou Ran and Liu Peng, "Dynamic Reservation Data Scheduling Mechanism for MPTCP," *Journal of Chongqing University of Posts and Telecommunications*, vol. 25, no. 6, 2013.

[20] M. Becke, T. Dreibholz, A. Bayer, M. Packeiser, and E. P. Rathgeb, "Alternative Transmission Strategies for Multipath Transport of Multimedia Streams over Wireless Networks," in *Proceedings of the 12th IEEE International Conference on Telecommunications (ConTEL)*, Zagreb, Središnja Hrvatska/Croatia, Jun. 2013, pp. 147–153, ISBN 978-953-184-175-7.

[21] S. Ferlin, T. Dreibholz, and Özgü Alay, "Multi-Path Transport over Heterogeneous Wireless Networks: Does it really pay off?" in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Austin, Texas/U.S.A., Dec. 2014, pp. 4807–4813, ISBN 978-1-4799-3512-3.

[22] V. Cerf, V. Jacobson, N. Weaver, and J. Gettys, "BufferBloat: What's Wrong with the Internet?" *ACM Queue*, vol. 9, no. 12, pp. 10–20, Dec. 2011, ISSN 1542-7730.

[23] T. Dreibholz, "NorNet – Building an Inter-Continental Internet Testbed based on Open Source Software," in *Proceedings of the LinuxCon Europe*, Berlin/Germany, Oct. 2016.

[24] E. G. Gran, T. Dreibholz, and A. Kvalbein, "NorNet Core – A Multi-Homed Research Testbed," *Computer Networks, Special Issue on Future Internet Testbeds*, vol. 61, pp. 75–87, Mar. 2014, ISSN 1389-1286.

[25] A. Kvalbein, D. Baltrūnas, K. R. Evensen, J. Xiang, A. M. Elmokashfi, and S. Ferlin, "The NorNet Edge Platform for Mobile Broadband Measurements," *Computer Networks, Special Issue on Future Internet Testbeds*, vol. 61, pp. 88–101, Mar. 2014, ISSN 1389-1286.

[26] T. Dreibholz, X. Zhou, and F. Fu, "Multi-Path TCP in Real-World Setups – An Evaluation in the NorNet Core Testbed," in *5th International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Gwangju/South Korea, Mar. 2015, pp. 617–622, ISBN 978-1-4799-1775-4.

[27] T. Dreibholz, J. Bjørgeengen, and J. Werme, "Monitoring and Maintaining the Infrastructure of the NorNet Testbed for Multi-Homed Systems," in *5th International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Gwangju/South Korea, Mar. 2015, pp. 611–616, ISBN 978-1-4799-1775-4.

[28] T. Dreibholz and E. G. Gran, "Design and Implementation of the NorNet Core Research Testbed for Multi-Homed Systems," in *Proceedings of the 3nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Barcelona, Catalonia/Spain, Mar. 2013, pp. 1094–1100, ISBN 978-0-7695-4952-1.

[29] T. Dreibholz, "NorNet – The Internet Testbed for Multi-Homed Systems," in *Proceedings of the Multi-Service Networks Conference (MSN, Coseners)*, Abingdon, Oxfordshire/United Kingdom, Jul. 2016.

[30] ——, "NorNet at NICTA – An Introduction to the NorNet Testbed," Invited Talk at National Information Communications Technology Australia (NICTA), Sydney, New South Wales/Australia, Jan. 2016.

[31] F. Golkar, T. Dreibholz, and A. Kvalbein, "Measuring and Comparing Internet Path Stability in IPv4 and IPv6," in *Proceedings of the 5th IEEE International Conference on the Network of the Future (NoF)*, Paris/France, Dec. 2014, pp. 1–5, ISBN 978-1-4799-7531-0.

[32] D. L. Mills, "Internet Delay Experiments," IETF, RFC 889, Dec. 1983, ISSN 2070-1721.

[33] T. Dreibholz, "NetPerfMeter: A Network Performance Metering Tool," *Multipath TCP Blog*, Sep. 2015.

[34] T. Dreibholz, M. Becke, H. Adhari, and E. P. Rathgeb, "Evaluation of A New Multipath Congestion Control Scheme using the NetPerfMeter Tool-Chain," in *Proceedings of the 19th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Hvar, Dalmacija/Croatia, Sep. 2011, pp. 1–6, ISBN 978-953-290-027-9.

[35] F. Fu, X. Zhou, T. Dreibholz, K. Wang, F. Zhou, and Q. Gan, "Performance Comparison of Congestion Control Strategies for Multi-Path TCP in the NorNet Testbed," in *Proceedings of the 4th IEEE/CIC International Conference on Communications in China (ICCC)*, Shenzhen, Guangdong/People's Republic of China, Nov. 2015, pp. 607–612, ISBN 978-1-5090-0243-6.

[36] R Development Core Team, *R: A Language and Environment for Statistical Computing*, Vienna/Austria, Mar. 2014.

[37] M. Becke, H. Adhari, E. P. Rathgeb, F. Fu, X. Yang, and X. Zhou, "Comparison of Multipath TCP and CMT-SCTP based on Intercontinental Measurements," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Atlanta, Georgia/U.S.A., Dec. 2013.

[38] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-friendly High-Speed TCP Variant," *ACM Operating Systems Review (SIGOPS)*, vol. 42, no. 5, pp. 64–74, Jul. 2008, ISSN 0163-5980.

[39] G. Fairhurst and M. Welzl, "The Benefits of using Explicit Congestion Notification (ECN)," IETF, Internet Draft draft-ietf-aqm-ecn-benefits-08, Nov. 2015.

[40] B. Briscoe, "Tunnelling of Explicit Congestion Notification," IETF, RFC 6040, Nov. 2010, ISSN 2070-1721.