On the Path Management of Multi-Path TCP in Internet Scenarios based on the NorNet Testbed

Kun Wang*, Thomas Dreibholz†, Xing Zhou*, Fu Fa*, Yuyin Tan*, Xi Cheng*, Qining Tan*

*Hainan University, College of Information Science and Technology, Haikou, China
Email: 780952372@qq.com, {zhouxing,fufa,tanyuyin}@hainu.edu.cn, {391204334,447542994}@qq.com

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

***Abstract*—With the rapid development of Internet communications, there is a growing demand to support devices being connected to multiple Internet service providers simultaneously. For example, every modern smartphone already provides at least mobile broadband (UMTS, LTE) as well as Wi-Fi interfaces. This multi-homing property can be used for resilience, but there is also an increasing interest in making use of concurrent multi-path transport. That is, multiple network paths can be utilised simultaneously, in order to improve the payload throughput for applications like big data or cloud computing.**

In this paper, we examine the performance of multi-path transport in real-world Internet setups, based on Multi-Path TCP (MPTCP) in the NorNet testbed for multi-homed systems. However, systems in such challenging setups need proper configuration. Therefore, we particularly would like to highlight the performance impact of different path management and congestion control settings in such realistic scenarios.

Keywords: Multi-Path Transport, Multi-Path TCP (MPTCP), Resilience, Path Management, Congestion Control, Configuration

I. Introduction

With the popularity of mobile Internet, more and more people are using mobile devices for communication. Most of these devices are able to transport data over at least two different network access infrastructures: mobile broadband (i.e. UMTS, LTE, etc.) and Wi-Fi (i.e. IEEE 802.11). And today, most major organisation sites make use of at least two or more different ISPs for backup reason. Each infrastructure usually uses standard Internet protocols to transmit the data over the Internet. However, the Transmission Control Protocol (TCP) [1] – as the most widespread Transport Layer protocol of the Internet – only selects one path (given by source and destination IP addresses, e.g. directly mapping to an underlying mobile broadband or Wi-Fi interface) for transmission. The other path, and therefore the other network interface, remains inactive. When losing connectivity on the interface chosen for a TCP connection, the connection is broken and needs to be reestablished over the other interface (with the other interface's IP address).

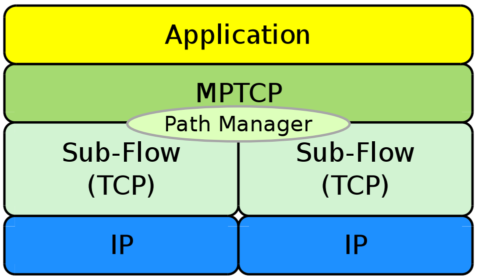


Figure 1. The Architecture of MPTCP

To overcome such problems, modern transport protocols like the Stream Control Transmission Protocol (SCTP) [2], [3] support multi-homing [4]: endpoints may have multiple IP addresses, e.g. due to connection to multiple Internet service providers (ISP). As long as there is connectivity between some of the addresses of two endpoints, the connection can remain operational. This path redundancy can be used for availability-critical applications [5], [6]. Furthermore, the Concurrent Multipath Transfer for SCTP extension – denoted as CMT-SCTP [7] – even allows to utilise paths simultaneously, in order to improve e.g. payload throughput [8], [9] or latency [5], [10]. This is denoted as multi-path transport. Multi-Path TCP (MPTCP) [11], [12] is an extension that adds multi-homing and multi-path transport to TCP. It brings many of the features already existing for SCTP to TCP. However, its most important advantage is backwards compatibility with the existing TCP (see e.g. [13]–[15] for more details), proving a smooth operation over MPTCP-unaware middlebox devices like firewalls and network/port address translation routers.

Regardless of the protocol chosen for multi-path transport, the general problems are the same:

- How to schedule payload data onto paths with highly dissimilar properties (bandwidth, delay, loss rate) appropriately [8], [9], [16]?
- How to perform proper congestion control handling in multi-path environments [17]–[21]?

In this paper, we examine above two topics – on the example of MPTCP – in real-world Internet setups, by using the NorNet Core testbed. Particularly, we want to highlight the challenges of such setups for the current state-of-the-art in multi-path transport.

II. Path Management

MPTCP [11], [12] implements multi-homing and multi-path transport as shown in Figure 1: each MPTCP connection consists of one or more TCP subflows. Each subflow is defined by a source/destination IP address (IPv4 or IPv6) pair.

---

[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] Parts of this work have been funded by the Research Council of Norway (Forskingsrådet), prosjektnummer 208798/F50.

[3] Xing Zhou is corresponding author.

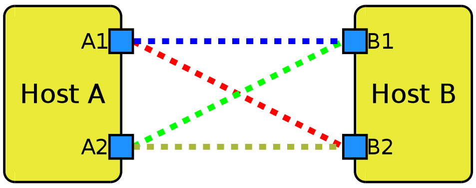


Figure 2. MPTCP Connection with Full-Mesh Paths

MPTCP even allows to simultaneously have IPv4 and IPv6 subflows. On the wire, each subflow appears like a regular TCP connection. Then, MPTCP-unaware middleboxes [14] in the network can handle a subflow like a TCP connection. The core idea of MPTCP is to share network resources by distributing payload data transport onto multiple subflows. Then, multiple paths in the underlying network can be utilised to maximise the overall connection throughput. The subflow model of the MPTCP protocol is explained in detail by [13], [16], [22].

An MPTCP connection is established as follows: given two systems, Host A and Host B, a regular TCP connection is established first. The MP_CAPABLE TCP option [11] set by each host signals the presence of MPTCP support. Once the initial connection (and therefore the first subflow) is established, the ADD_ADDR [11] TCP option is applied to add further subflows to the existing MPTCP connection. In the example given in Figure 2, with two network interfaces for each host (with IP addresses A1 and A2 for Host A, and IP-addresses B1 and B2 for Host B), finally four subflows can be established: A1-B1, A1-B2, A2-B1, A2-B2. If all possible subflows are established (which is useful for Internet setups [16], [22]), there is a full mesh of paths.

The actual decisions about path establishment (e.g. using a full mesh, or just a subset?) are implementation-specific. In Linux MPTCP [15], the most state-of-the-art implementation of MPTCP, this decision is based on the configured path manager. Four path managers are currently provided by Linux MPTCP:

1) "default": This path manager actually does not do anything. It will neither announce different IP addresses nor initiate the creation of new subflows. However, it will accept the passive creation of new subflows.
2) "fullmesh": As the name already says, this path manager establishes the full mesh [16] of subflows.
3) "ndiffports": Instead of using multiple IP addresses, this path manager always uses the same IP-address pair for its paths. However, each path uses different source and destination TCP ports. This path manager is intended to circumvent bandwidth-limiting middleboxes by mimicking different TCP connections.
4) "binder": This path manager [23] is using Loose Source Routing [24] to distribute the packets of subflows. Using packet relays, it allows for applications on end-user devices to benefit from gateway aggregation without requiring any modifications.

III. CONGESTION CONTROL

Besides path management, congestion control is a further important protocol mechanism. It not only has to take care for a reasonable allocation of network resources, but also has to ensure reasonable fairness in multi-path setups [17], [19]. For MPTCP (and multi-path transport in general), congestion control is used to adjust the congestion windows of subflows, in order to control each subflow's transmission rate. In order to achieve TCP-friendly Internet deployment, the following three rules [25], [26] of practical multi-path congestion control should be met:

- *Rule 1 ("Improve Throughput")*: A multi-path flow should perform at least as well as a single path flow would on the best of the paths available to it.
- *Rule 2 ("Do no Harm")*: A multi-path flow should not take up more capacity from any of the resources shared by its different paths than if it were a single flow using only one of these paths. This guarantees it will not unduly harm other flows.
- *Rule 3 ("Balance Congestion")*: A multi-path flow should move as much traffic as possible off its most congested paths, subject to meeting the first two goals.

The first two rules ensure fairness at a shared bottleneck, while the last rule makes use of the concept of resource pooling [27]: if each multi-path flow sends more data through its least-congested path, the traffic in the network will move away from congested areas. This improves robustness and overall throughput.

The way to achieve resource pooling is to effectively "couple" the congestion control loops of the different subflows. Congestion control algorithms can therefore be divided into two categories: uncoupled algorithms (which handle each subflow independently, assuming independent paths) and coupled algorithms (which apply resource pooling). For some more details, see e.g. [8], [17], [18], [28]. Relevant for this paper are two algorithms:

- Cubic [29] is the uncoupled default algorithm used by Linux for TCP and MPTCP, i.e. widely deployed. However, it gives the multi-path flow an unfair share when the paths taken by its different subflows share a common bottleneck [18].
- OLIA (Opportunistic LIA [30]) is a further development of LIA (Linked Increases Algorithm [25]). OLIA (and LIA) are based on New Reno congestion control [31], and just add the path coupling. OLIA is now the main coupled algorithm of Linux MPTCP. The basic idea is to solve the unfairness issue of uncoupled congestion control on shared bottlenecks [18].

IV. MEASUREMENT SCENARIO DESIGN

A. *The* NORNET CORE *Testbed*

The NORNET [32] 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 [33]–[36]. 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 are spread over currently 21 sites [32] on four continents, with sites connected to multiple different ISPs with different access technologies. Obviously, a key feature of NORNET CORE is to work in the real-world Internet. The information for the NORNET CORE sites used for this paper 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 some cases ADSL connections – marked with "A") are shown in yellow colour.

---

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

Table I. THE NORNET CORE TESTBED SITES USED FOR THE MEASUREMENTS

Index	Site	Abbreviation	Location (City, Province, Country)	ISP 1	ISP 2	ISP 3
3	Høgskolen i Gjøvik	HiG	Gjøvik, Oppland, Norway	Uninett	PowerTech[A]	–
6	Universitetet i Bergen	UiB	Bergen, Hordaland, Norway	Uninett	BKK	–
9	NTNU Trondheim	NTNU	Trondheim, Sør-Trøndelag, Norway	Uninett	PowerTech[A]	–
10	Høgskolen i Narvik	HiN	Narvik, Nordland, Norway	Uninett	Broadnet[A]	PowerTech[A]
42	Universität Duisburg-Essen	UDE	Essen, Nordrhein-Westfalen, Germany	DFN	Versatel[A]	–
88	Hainan University	HU	Haikou, Hainan, China	CERNET	China Unicom	–
89	Haikou College of Economics	HKC	Guilinyang, Hainan, China	China Telecom	CERNET	–
100	The University of Kansas	KU	Lawrence, Kansas, United States	KanREN	–	–
160	Korea University	KRU	Seoul, South Korea	KREONET	–	–

B. Measurement Tools

The bandwidth measurements have been performed by applying the NETPERFMETER [8], [37], [38] tool. It provides the performance comparison of multiple transport connections and protocols, including MPTCP support [18], [34], [37]. All results have been processed with GNU R [39]. All measurement scenarios we have run 20 times, and got the average value by NETPERFMETER. Results plots show the average application payload throughput of a saturated NETPERFMETER flow running 300 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 [15] version 0.91[2],
(c) ndiffports=2 (only relevant for "ndiffports" path manager),
(d) Explicit Congestion Notification (ECN) support [40], [41] enabled, and
(e) TCP (and MPTCP) buffer size limit [42] set to 16 MiB[3] (in order to avoid effects caused by buffer space scarcity).

V. RESULTS ANALYSIS

In order to show the effects of different settings for path manager and congestion control, we have selected five different scenarios for this paper.

A. Challenging Inter-Continental Multi-Homing Scenario

In the first scenario, we analyse the transport performance 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. Obviously, the 4 paths in this scenario have significantly differing QoS characteristics (i.e. bandwidth, delay, loss rate), making the setup challenging for multi-path transport [16].

In providing a baseline performance overview, Figure 3 presents the received payload data (in Mbit/s) within the first 60 s, for single-path TCP using each of the four paths. Note, that this is just a snapshot of one single measurement, in order to illustrate the short-term behaviour. We will examine the long-term behaviour later. Also note that the received data is counted on the Application Layer, i.e. data can only be forwarded to the Application Layer when all previous data has been delivered (since TCP provides ordered delivery, see [8, Section 2.8]). For better readability, different destination ISPs use different colours (red and blue), while different source ISPs use different line styles (solid and dashed). The results for Cubic congestion control are presented in Subfigure 3(a) (left-hand side), while the OLIA results can be found in Subfigure 3(b) (right-hand side).

Clearly, when using the ADSL connection (i.e. Versatel) for output, the performance is lowest. The uplink speed is just about 1 Mbit/s, which is then the bottleneck for the connection. When using the high-speed network (i.e. DFN) for output instead, the bottleneck is nearby the destination ISP (i.e. CnUnicom or CERNET). In this case, the combination of DFN→CnUnicom offers the best performance, which is usually an average payload throughput of about 7 Mbit/s. Comparing Cubic and OLIA (which acts like New Reno [31] here, since there is only one path for TCP), it is observable that there is some more variation for Cubic in the received payload rate.

When using MPTCP instead of TCP, the path manager used for MPTCP becomes relevant. Therefore, the results for MPTCP in Figure 4 are separated by path manager: "default", "ndiffports", "binder", and "fullmesh" (see Section II). Obviously, "default" (Subfigure 4(a)/4(b)) does not significantly differ from TCP (see Figure 3), since only a single path is used for data transport: the performance depends on the choice of this single path. If it is DFN→CnUnicom (the best path for single-path TCP), the best performance is reached. All other 3 choices perform worse. So, the application (or even the user) must make a good choice to get a good performance.

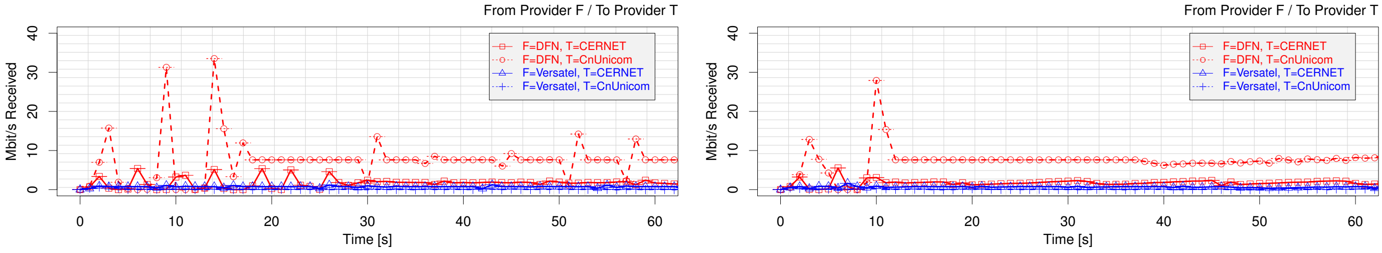
Using "ndiffports" instead (Subfigure 4(c)/4(d)) does not change the situation. Instead of using one path, the same path is used twice (by two subflows, since ndiffports=2), just with different TCP port numbers for the subflows. Since our setup does not contain load balancers distributing TCP traffic (for load balancers, MPTCP traffic looks like TCP traffic) based on port numbers, the "ndiffports" path manager has no significant effect here. Instead, all disadvantages of "default" also apply here.

"binder" applies the Loose Source and Record Route (LSRR) IP option [24], in order to try distributing packets of different subflows to paths. It requires the local network to support LSRR, which is – in most networks – turned off for security reasons. So, while "binder" is useful in appropriately-configured community networks [23], it is not useful for general multi-homed Internet setups. The LSRR options are simply ignored here, leading to having all packets routed over the same path (i.e. the initially chosen path). The result (Subfigure 4(e)/4(f)) is therefore similar to "default".

---

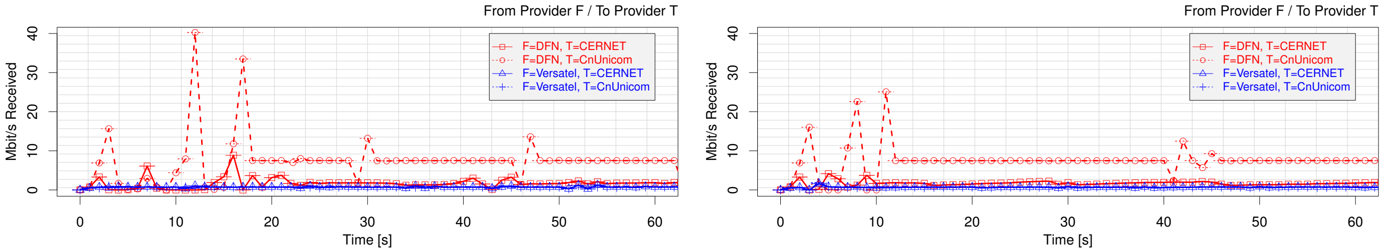
[2] Available from http://www.multipath-tcp.org.

[3] `sysctl`: net.ipv4.tcp_rmem, net.ipv4.tcp_wmem, net.ipv4.tcp_mem.

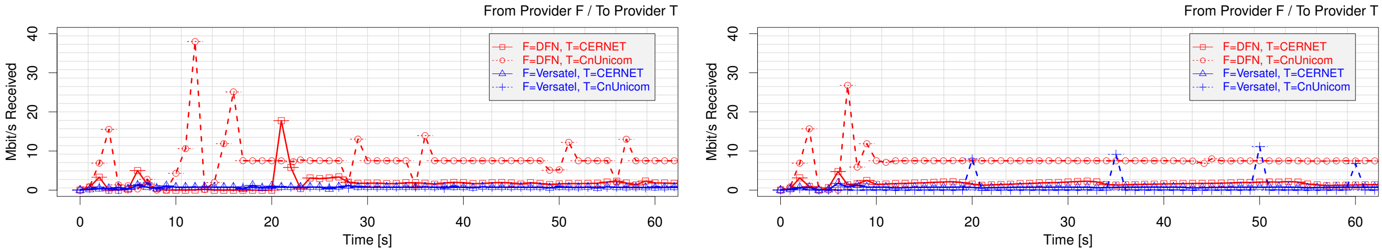


(a) Cubic Congestion Control (b) OLIA Congestion Control

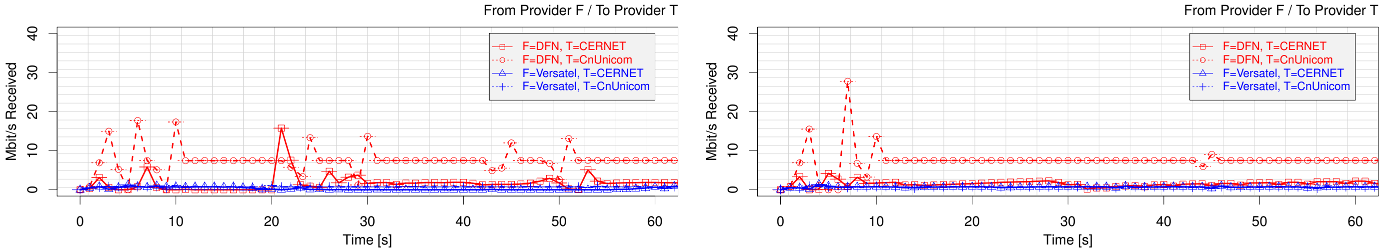
Figure 3. Short-Term TCP Results for the Universität Duisburg-Essen (UDE) → Hainan University (HU) Scenario



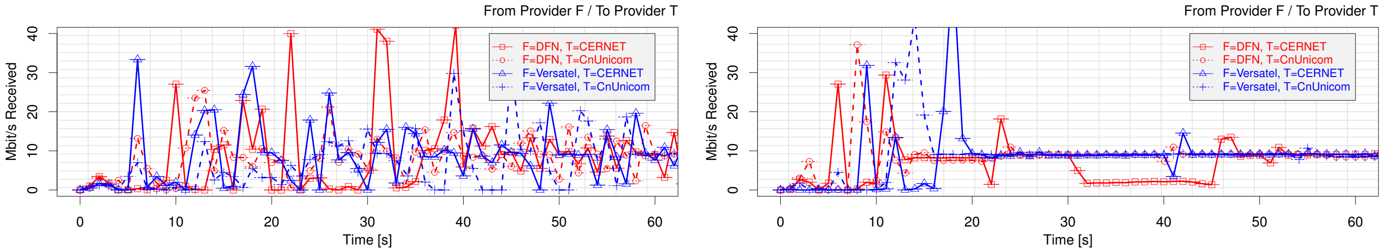
(a) "default" Path Manager, Cubic Congestion Control (b) "default" Path Manager, OLIA Congestion Control



(c) "ndiffports" Path Manager, Cubic Congestion Control (d) "ndiffports" Path Manager, OLIA Congestion Control

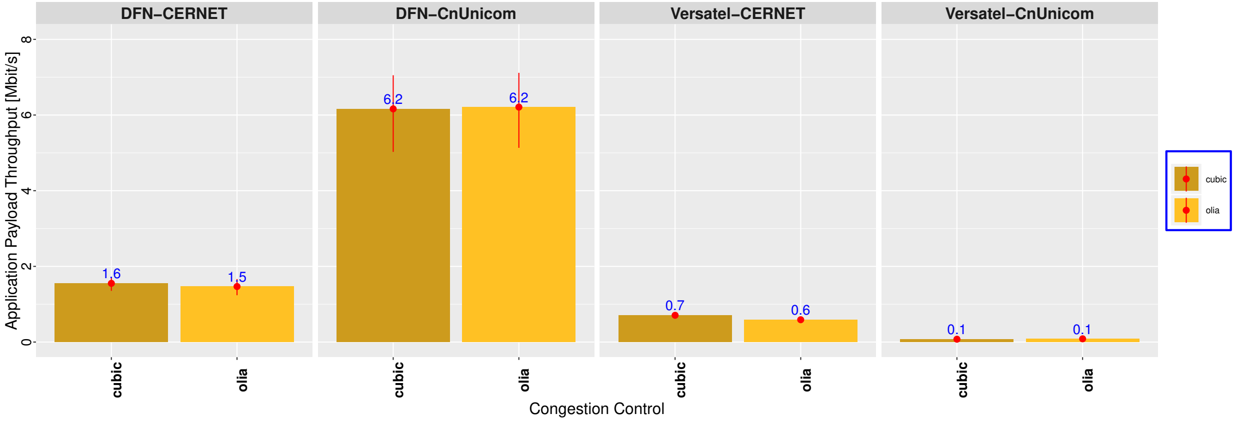


(e) "binder" Path Manager, Cubic Congestion Control (f) "binder" Path Manager, OLIA Congestion Control

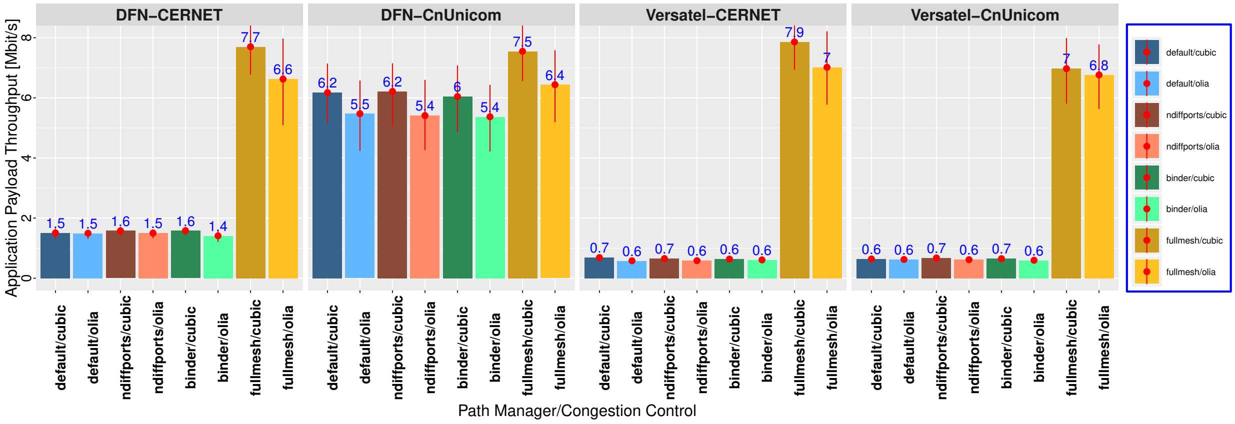


(g) "fullmesh" Path Manager, Cubic Congestion Control (h) "fullmesh" Path Manager, OLIA Congestion Control

Figure 4. Short-Term MPTCP Results for the Universität Duisburg-Essen (UDE) → Hainan University (HU) Scenario



(a) TCP



(b) MPTCP

Figure 5. Long-Term Results for the Universität Duisburg-Essen (UDE) → Hainan University (HU) Scenario

Only the "fullmesh" path manager (Subfigure 4(g)/4(h)) really makes use of the different paths provided by the different ISP combinations in our setup. In this case, all 4 paths get utilised. After some initialisation phase, the received data rate stabilises at similar values, regardless of the choice for the initial path. That is, after some time, it does not matter whether the best, the worst, or any other path has initially been used to establish the MPTCP connection. So, the application (or the user) does not have to take care of the path choice to get a decent performance. As expected from the TCP results in Figure 4, the difference between Cubic and OLIA now becomes clearer. While Cubic [29] has a higher variation (and, as uncoupled congestion control, behaves in the same way on each path), OLIA takes care of stability as well. After the first 15 s, there is only small variation. Of course, the results presented here are only a 60 s snapshot of a single measurement. So, the obvious question is: How is the long-term behaviour?

In demonstrating the long-term behaviour, Figure 5 presents the average application payload throughput over 20 runs, with the TCP results in Subfigure 5(a) and the MPTCP results in Subfigure 5(b). As expected from the short-term results for TCP, the performance mostly depends on the choice of the path. Furthermore, using Cubic instead of OLIA (which just behaves like standard New Reno in this single-path TCP scenario) has a slightly better performance (e.g. 1.6 Mbit/s vs. 1.5 Mbit/s for DFN→CERNET). Cubic [29] is a newer algorithm than New Reno [31], trying to improve TCP performance on high-speed connections. This is also the case in our inter-continental Internet setup (so Linux's default – i.e. Cubic – is reasonable).

The long-term MPTCP results (see Subfigure 5(b)) confirm the observations already made with the corresponding short-term results: the performance of the "default", "ndiffports" and "binder" path managers depends mostly on the choice of the initial path. So, only when using the best path (DFN→CnUnicom), the performance of TCP on this path can be reached. The payload throughput is only a fraction when using any other path. So, without a scenario that can benefit from "ndiffports" (load balancer) or "binder" (networks supporting LSRR), the "fullmesh" path manager is the only useful choice here to achieve a good throughput. Resilience – i.e. handling broken paths – however would also be provided by the other path managers as well.

Furthermore, the MPTCP performance with "fullmesh" is even better than for TCP over the best path (see Subfigure 5(a), due to the utilisation of multi-path transport), satisfying Rule 1 ("Improve Throughput", see Section III). Obviously Cubic provides a better performance than OLIA, since Cubic can handle paths independently, while OLIA must always assume that paths may have shared bottlenecks. So, Cubic more aggressively claims bandwidth resources. In our setup with different ISPs, it may be justified to assume this to be acceptable (see [17], [19] for a general discussion). However,

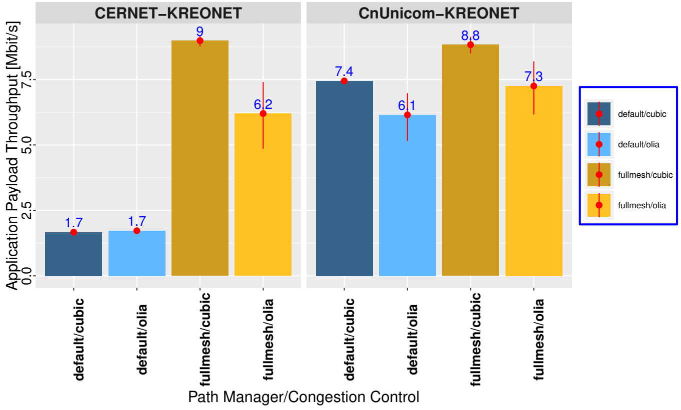


Figure 6. MPTCP for Hainan University (HU) → Korea University (KRU)

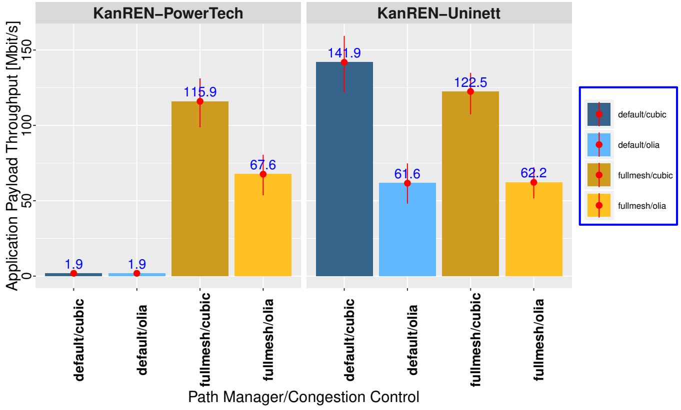


Figure 7. MPTCP: The Univ. of Kansas (KU) → Høgskolen i Gjøvik (HiG)

even when using the less aggressive OLIA, the performance is usually better than with TCP over the best path. So far, we have only observed a single – but challenging – multi-homed setup. So, what about other scenarios?

B. Further Scenarios

In examining further setups, Figure 6 shows the average MPTCP payload throughput results (over 20 runs) for the Hainan University (HU) → Korea University (KRU) scenario, and Figure 7 presents the results for The University of Kansas (KU) → Høgskolen i Gjøvik (HiG). We omit "ndiffports" and "binder" results here, since these path managers are not useful for our setups. Both scenarios have in common that one side is single-homed (KRU and KU; see Table I) using the local research network ISP (KREONET, KanREN), while the peer side is dual-homed. Therefore, each "fullmesh" MPTCP connection consists of two subflows.

As already seen from the results in Subsection V-A, the "fullmesh" MPTCP usage makes the performance independent of the choice for the initial subflow. While the MPTCP performance clearly exceeds the performance of the best path for HU→KRU (about 8.8 Mbit/s vs. 7.4 Mbit/s for Cubic), the throughput for KU→HiG is slightly lower. Here, e.g. about 122.5 Mbit/s are achieved instead of 141.9 Mbit/s for Cubic. The reason here is the very high difference in ISP speeds: high-performance research network (Uninett) vs. slow-speed ADSL connection (PowerTech). Nevertheless, the MPTCP performance is two orders of magnitude better than for TCP over the bad path (i.e. just 1.8 Mbit/s). Again, in both scenarios, Cubic performs better than OLIA – since both paths are handled independently.

To further examine the performance, particularly also Cubic vs. OLIA, Figure 8 presents the average MPTCP payload throughput results (over 20 runs) for Universitetet i Bergen (UiB) → Haikou College of Economics (HKC). In this case, the UiB site is connected to two high-speed ISPs (with at least 100 Mbit/s); the ISPs at HKC are slower but usually provide speeds in the range of 5 Mbit/s to 20 Mbit/s. The throughput limitation here is mostly the inter-continental transport between Norway and China, not the ISP connectivity.

While MPTCP makes the performance – mostly – independent of the path choice for the initial subflow, a closer look at the bandwidth aggregation performance becomes interesting here: by using OLIA, at least the performance of TCP over the best path (5 Mbit/s for Uninett→CERNET) is reached. However, in this setup, the performance of OLIA is significantly lower than for Cubic: it is just about one fifth (e.g. 6.8 Mbit/s vs. 29.1 Mbit/s)! The reason is a higher level of background congestion (e.g. compared to the UDE→HU scenario in Subsection V-A), causing a higher amount of packet losses on the CnTelecom paths. Due to coupling of the paths in congestion control, this leads to an overly reduced size of *all* congestion windows – and therefore to a significantly reduced throughput. Cubic circumvents this problem – uncoupled congestion control only leads to a throughput reduction on the problematic paths (here: to CnTelecom), while the unproblematic paths (here: to CERNET) can still utilise their full capacity.

So, does OLIA always perform (much) worse than Cubic? To show a counterexample, Figure 9 presents the average MPTCP payload throughput results (over 20 runs) for the NTNU Trondheim (NTNU) → Høgskolen i Narvik (HiN) Scenario. As stated in Table I, the NTNU site has 2 ISPs, while the HiN site even has 3 ISPs. But except for the research network ISP (Uninett) at each site, all other ISP connections are low-speed ADSL. In total, there are 6 paths. Obviously, for TCP, only the Uninett→Uninett path provides a good performance: about 86.6 Mbit/s. All other paths, are much worse – with only Uninett→Broadnet reaching 13.8 Mbit/s, while the four others not even achieve 2 Mbit/s. MPTCP almost reaches the performance of TCP on the best TCP, but obviously it is not able to exceed it. Here, using multi-path transport is in fact not really beneficial for throughput improvement. A solution like [43] could e.g. just avoid using them for payload, while keeping them for redundancy. Interestingly, OLIA performs slightly better than Cubic here: the loss rates on the ADSL paths are higher, reducing the congestion windows on these paths. While coupled congestion control also reduces the congestion window on the good Uninett→Uninett path, its small delay (bee-line distance between Trondheim and Narvik is less than 700 km) however leads to a quick recovery again. So, the high-speed path can still perform well. Therefore, OLIA achieves (with about 86.7 Mbit/s to 86.9 Mbit/s) more throughput than Cubic (with 82.3 Mbit/s to 86.6 Mbit/s) here. Note, however, that this scenario with multiple consumer-grade ADSL connections in addition to a high-speed ISP is somewhat unrealistic for real production setups.

VI. CONCLUSIONS

Multi-homed system setups have become increasingly popular and the need for multi-path transport – with e.g. MPTCP

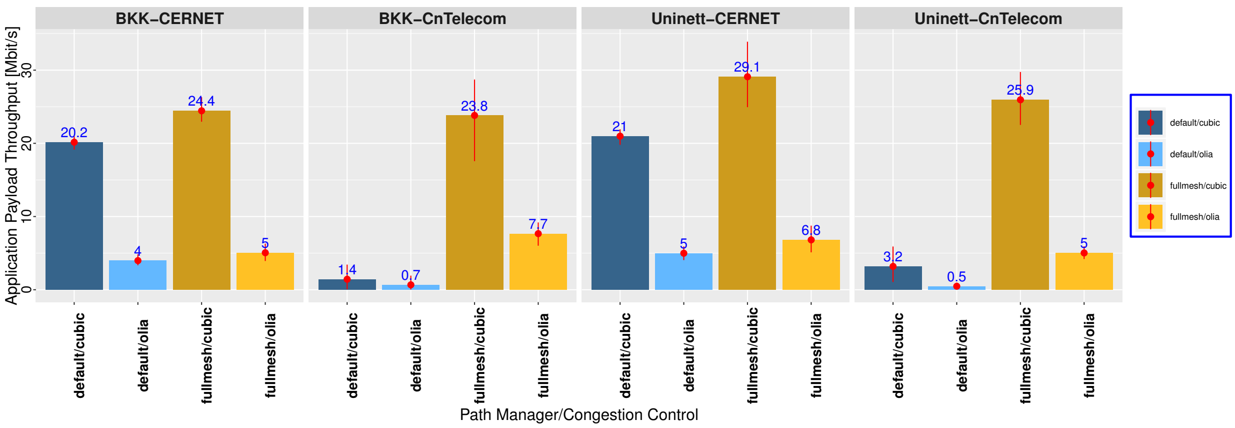


Figure 8. Long-Term MPTCP Results for the Universitetet i Bergen (UiB) → Haikou College of Economics (HKC) Scenario

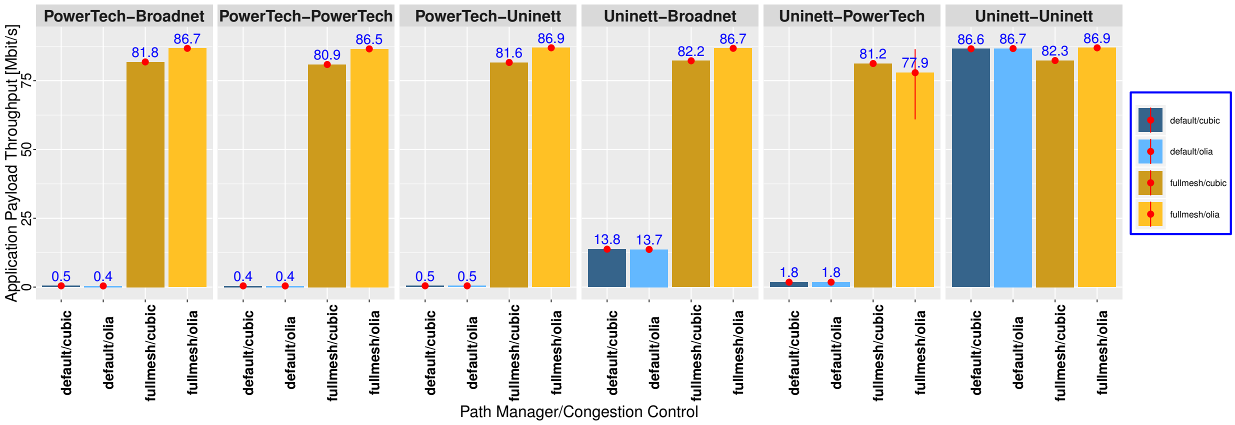


Figure 9. Long-Term MPTCP Results for the NTNU Trondheim (NTNU) → Høgskolen i Narvik (HiN) Scenario

as transport protocol – is increasing as well. However, to really utilise the features of MPTCP, some configuration is necessary. Therefore, in this paper, we have examined the performance impact of path manager and congestion control in real-world Internet setups using the NORNET CORE testbed. As a general recommendation, we have shown that:

- Single-path TCP in multi-homed setups requires the application (or even the user) to carefully choose the right path to achieve a good performance. Without careful selection, the performance is likely to be bad.
- MPTCP with its “default” path manager does not improve the performance in comparison to TCP. It only adds path redundancy for resilience.
- “ndiffports” and “binder” path managers are only useful in special scenarios (load balancing, gateways), i.e. not generally useful in real multi-homed Internet setups.
- Only the “fullmesh” path manager makes the long-term performance – mostly – independent of the chosen initial path. Furthermore, it can provide bandwidth aggregation benefits.
- Uncoupled Cubic congestion control performs better than coupled OLIA. Particularly, in high-delay, intercontinental setups with multi-homing, OLIA performs significantly worse than Cubic. Since uncoupled congestion control can be justified in scenarios with multiple ISPs (since the user pays for each ISP!), it is the recommended choice for such setups.

As part of future work, a more fine-granular analysis of the different subflows is necessary. As we have shown in this paper, it is possible that inappropriate behaviour of the congestion control (e.g. OLIA on congested long-distance paths), or multi-path transport with very diverse paths (e.g. high-speed fibre plus low-speed ADSL) can impair the overall performance. Therefore, our goal is to develop a more flexible path manager that takes such cases into account when making scheduling decisions. Also, it is useful to take a closer look at resilience by path redundancy as well. Of course, we are going to evaluate new approaches again in realistic, larger-scale Internet scenarios in the NORNET testbed.

REFERENCES

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

[2] 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.

[3] R. R. Stewart, “Stream Control Transmission Protocol,” IETF, RFC 4960, Sep. 2007, ISSN 2070-1721.

[4] M. Handley, “Why the Interent Only Just Works,” *BT Techology Journal*, vol. 24, no. 3, pp. 119–129, Jul. 2006, ISSN 1358-3948.

[5] J. Eklund, “Latency Reduction for Soft Real-Time Traffic using SCTP Multihoming,” Ph.D. dissertation, Karlstads Universitet, Faculty of Health, Science and Technology, Department of Mathematics and Computer Science, Jun. 2016, ISBN 978-91-7063-693-6.

[6] T. Dreibholz, “Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture,” Ph.D. dissertation, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, Mar. 2007.

[7] 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.

[8] 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.

[9] 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.

[10] K. V. Yedugundla, S. Ferlin, T. Dreibholz, Özgü Alay, N. Kuhn, P. Hurtig, and A. Brunström, “Is Multi-Path Transport Suitable for Latency Sensitive Traffic?” *Computer Networks*, vol. 105, pp. 1–21, Aug. 2016, ISSN 1389-1286.

[11] 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.

[12] 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.

[13] M. Becke, “Revisiting the IETF Multipath Extensions on Transport Layer,” Ph.D. dissertation, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, Nov. 2014.

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

[15] 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.

[16] 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.

[17] 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, Jun. 2012, pp. 2666–2672, ISBN 978-1-4577-2052-9.

[18] 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.

[19] T. Dreibholz, M. Becke, H. Adhari, and E. P. Rathgeb, “On the Impact of Congestion Control for Concurrent Multipath Transfer on the Transport Layer,” in *Proceedings of the 11th IEEE International Conference on Telecommunications (ConTEL)*, Graz, Steiermark/Austria, Jun. 2011, pp. 397–404, ISBN 978-953-184-152-8.

[20] F. Fu, X. Zhou, Y. Tan, T. Dreibholz, H. Adhari, and E. P. Rathgeb, “Performance Analysis of MPTCP Protocol in Multiple Scenarios,” *Computer Engineering and Applications*, vol. 52, no. 5, pp. 89–93, Mar. 2016, ISSN 1002-8331.

[21] J. Liu, Y. Tan, F. Fu, T. Dreibholz, X. Zhou, Y. Bai, X. Yang, and W. Du, “Study on MPTCP and CMT-SCTP Congestion Control Mechanism,” *Computer Engineering*, vol. 41, no. 4, pp. 117–124, Apr. 2015, ISSN 1000-3428.

[22] F. Fu, X. Zhou, X. Yang, 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, Nov. 2013, ISSN 1002-8331.

[23] L. Boccassi, M. M. Fayed, and M. K. Marina, “Binder: A System to Aggregate Multiple Internet Gateways in Community Networks,” in *Proceedings of the ACM MobiCom Workshop on Lowest Cost Denominator Networking for Universal Access (LCDNet)*, Miami, Florida/U.S.A, Sep. 2013, pp. 3–8, ISBN 978-1-4503-2365-9.

[24] J. B. Postel, “Internet Protocol,” IETF, Standards Track RFC 791, Sep. 1981, ISSN 2070-1721.

[25] C. Raiciu, M. Handley, and D. Wischik, “Coupled Congestion Control for Multipath Transport Protocols,” IETF, RFC 6356, Oct. 2011, ISSN 2070-1721.

[26] C. Raiciu, D. Wischik, and M. Handley, “Practical Congestion Control for Multipath Transport Protocols,” University College London, London/United Kingdom, Tech. Rep., Nov. 2009.

[27] D. Wischik, M. Handley, and M. B. Braun, “The Resource Pooling Principle,” *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 38, no. 5, pp. 47–52, Oct. 2008, ISSN 0146-4833.

[28] M. Welzl, *Network Congestion Control: Managing Internet Traffic*. Chichester, West Sussex/United Kingdom: John Wiley & Sons, 2005, ISBN 978-0-470-02528-4.

[29] 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.

[30] R. Khalili, N. G. Gast, M. Popović, and J.-Y. L. Boudec, “Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP,” IETF, Individual Submission, Internet Draft draft-khalili-mptcp-congestion-control-05, Jul. 2014.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] T. Dreibholz, “NetPerfMeter: A Network Performance Metering Tool,” *Multipath TCP Blog*, Sep. 2015.

[38] 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.

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

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

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

[42] F. Zhou, T. Dreibholz, X. Zhou, F. Fu, Y. Tan, and Q. Gan, “The Performance Impact of Buffer Sizes for Multi-Path TCP in Internet Setups,” in *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Taipei, Taiwan/People’s Republic of China, Mar. 2017.

[43] 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.