Improvement and Implementation of a Multi-Path Management Algorithm based on MPTCP

Min Chen*, Thomas Dreibholz‡, Xing Zhou*, Xuelei Yang*
*Hainan University
Renmin Avenue 58, 570228 Haikou, Hainan, China
chenmin@hainanu.edu.cn, zhouxing@hainanu.edu.cn, 969557139@qq.com
‡Simula Metropolitan Centre for Digital Engineering, Centre for Resilient Networks and Applications
Pilestredet 52, N-0167 Oslo, Norway
dreibh@simula.no

***Abstract*—The core idea of the Multi-Path Transmission Control Protocol (MPTCP) is to utilize multiple network connections by distributing payload data transmission among several subflows. Then, multiple paths in the underlying networks can be used to maximize the overall connection throughput. However, the concurrent transmission on only a subset of all possible subflows' aggregation can improve network performance, because of performance differences between the subflow. In this paper, we propose a new FullMesh algorithm based on Path Characteristic and Data Characteristic (PCDC), in which a Subflow Impact Factor (IF) is used as a subflow characteristic to predict the impact of a subflow on the overall throughput. Then, different path sets are adopted for different sizes of traffic. The PCDC algorithm is evaluated in the NorNet Core testbed, comparing it to the FullMesh algorithm. Our research results show that the PCDC algorithm can improve the network throughput and reduce the overall completion time of small data streams.**[1][2][3]

Keywords: MPTCP, Multi-path Management, PCDC, Subflow Impact Factor, Data Stream Classification

I. Introduction

With Internet-connected devices becoming increasingly ubiquitous in our daily life, there is also a growing amount of different access technologies. These technologies range from low-bandwidth, high-delay 2G connections over fast but unreliable Wi-Fi networks to high-speed, low-delay 5G as well as satellite Internet links. Furthermore, many devices may in fact be connected to multiple underlying networks *simultaneously*, e.g. a smartphone being connected to Wi-Fi and 4G. Effectively utilizing the redundant network resources and improving the network performance – despite the very dissimilar characteristics of the underlying networks – have become a hot issue in recent years. Therefore, the Internet Engineering Task Force (IETF) puts forward the Multi-Path Transmission Control Protocol (MPTCP) [1]–[4], which can aggregate the bandwidths, improve the throughput, as well as enhance the robustness and fast recovery by using the existing network infrastructure in the way of software.

[1]This work has been funded by the NSFC of China (No. 61662020), CERNET NGI Technology Innovation Project (No. NGII20160110), and the Research Council of Norway (Forskingsrådet, No. 208798/F50).

[2]Xing Zhou is corresponding author.

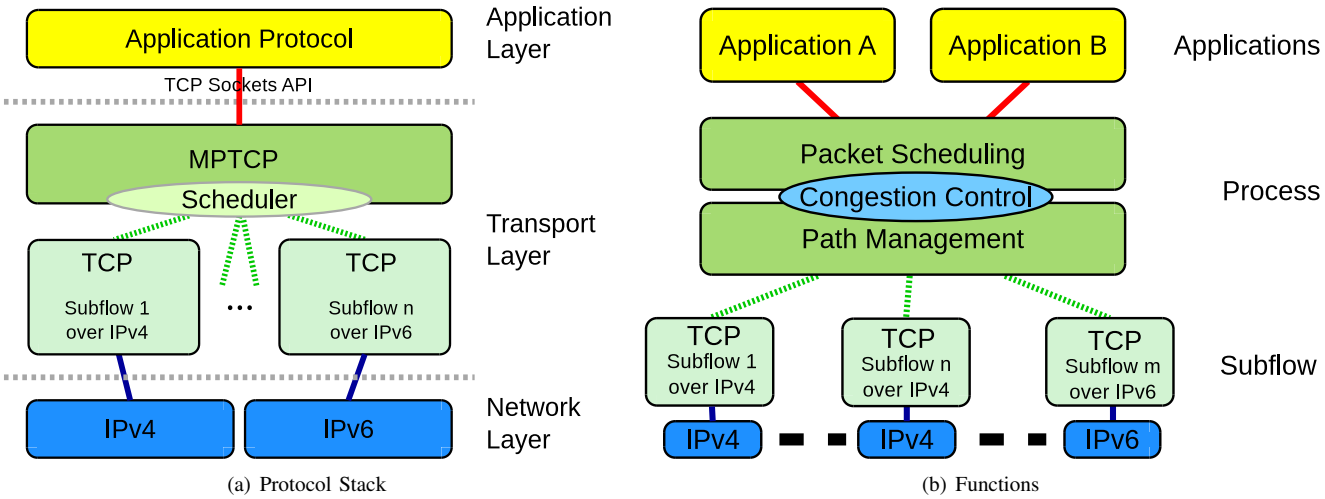
[3]The authors would like to thank Ann Edith Wulff Armitstead and Ted Zimmerman for their comments.

MPTCP, which is an extension of the well-known Transmission Control Protocol (TCP) [5], realizes multi-path transmission on the Transport Layer. While TCP only establishes a single-path connection for the communication between two hosts, MPTCP can dynamically conduct a multi-path connection consisting of multiple subflows. Subfigure 1(a) illustrates the MPTCP protocol stack [4]. It can be seen that MPTCP provides transparent end-to-end concurrent data transmission services to achieve the purpose of aggregating bandwidth and improving transport performance. MPTCP is compatible with the TCP protocol. That is, most of the current network applications are based on TCP, and TCP has a mature and extensive application ecosystem. Therefore, MPTCP can be applied without any change of the current applications. This differentiates its deployment possibility from the Stream Control Transmission protocol (SCTP) [6]–[8], with its Concurrent Multipath Transfer (CMT-SCTP) [9] extension for multi-path transport. Subfigure 1(b) shows the details of the MPTCP protocol functions. Mainly, it consists of two parts:

1) Path Management (PM) denotes the establishment, teardown and management of subflows.
2) Packet Scheduling (PS) is the scheduling of payload data onto the existing subflows.

PM and PS are closely related to Congestion Control (CC) [10], [8, Section 2.11], [11] algorithms, while the main task of Path Management is to organize and manage those subflows that can participate in the end-to-end data transmission and make contributions. Through the PM algorithm, it is possible to dynamically add or delete subflows to participate in the concurrent transmission.

MPTCP is still in the research stage, and the performance of each aspect needs to be fully verified and improved in network practice, and so does PM. The existing PM algorithms do not consider whether they can really improve the overall transmission performance when adding subflows to join the transmission as long as one of the subflows is available. It has been shown for real-world setups that by only using a subset of the subflows for concurrent multi-path transfer, the performance may be improved [10], [12]. The current PM



(a) Protocol Stack (b) Functions

Figure 1. The Architecture of MPTCP

algorithms do not consider the performance of subflows and the characteristics of the underlying network paths. They are either all joined or passively accepted, which reduces the performance in scenarios where path characteristics become highly dissimilar.

According to [13], [14], it is found that in the real network data transmission, 99% of the single transmission traffic is less than 100 MiB, and most of it is within 1 MiB. From the perspective of data size, when the data size is small, users are more sensitive to the completion time of their transmission. That is, when transmitting such data, the delay at both ends is crucial. [15] shows that, if the delay in the network increases by 100 ms, AMAZON's sales revenue will decrease by 1.6%, and if the delay increases by 500 ms, BING's search revenue will decrease by 1.2%. With the development of interactive network applications, small traffic data exchange in the Internet constitutes the majority of the communications flows. But most of the data volume transmitted in the network is from large data streams. This kind of transmission is common in file download and upload, live video broadcasts and so on. When transmitting large data streams, users have higher requirements for throughput. At this time, all subflows that are conducive to high throughput should be allowed to participate in the concurrent multi-path transmission over the network. In [16], it is found that MPTCP can effectively improve the network throughput for large data stream transmissions, but when transmitting small data streams, it significantly prolongs the completion time, particularly in heterogeneous networks with dissimilar path characteristics.

In this paper, we evaluate the influence degree of paths on the whole network transmission and analyze the reason why the completion time of small data stream transmission increases when using MPTCP. Based on the subflow characteristics and the data characteristics of the transmission traffic, we propose an improved FullMesh algorithm, named Path Characteristic and Data Characteristic based FullMesh (PCDC). This algorithm can improve the network throughput, and reduce the completion time of small data streams.

The rest of this paper is organized as follows: In Section II, we briefly introduce the path management of MPTCP and discuss the path selection issue. The main factors leading to an increase of the completion time for small data stream transmissions in MPTCP are discussed in Section III. Section IV presents the proposed PCDC algorithm. In Section V, we describe our measurement setup. Section VI demonstrates the performance of the proposed algorithm through experimental evaluation in the NORNET CORE testbed. Finally, Section VII concludes this paper.

II. PATH MANAGEMENT AND IMPACT FACTOR

Multi-path management algorithms can use multi-interface Internet Service Providers (ISP) to add and delete subflows between two hosts using MPTCP. Its working principle is to tell the other host upon connection establishment that multi-path transmission can be used and supply the available address of the local machine, and then establish a new subflow and add it to the MPTCP connection. At present, the official four path management algorithms provided by the Linux MPTCP [17] reference implementation are:

1) Default,
2) NDiffPorts,
3) Binder,
4) FullMesh.

The Default algorithm does not actively do anything, but passively accepts a new subflow creation. It can seamlessly switch to multiple other paths for transmission. The NDiffPorts algorithm uses multiple port numbers to achieve parallel transmission, but the IP address will not change. In that way multiple transmission paths are created on the same IP address, enabling simulation of different TCP connections through port numbers to avoid bandwidth restrictions. The Binder [18] algorithm uses Loose Source Routing [19] to distribute the

packets of subflows. Using packet relays, endpoints can benefit from gateway aggregation without requiring any modifications. Finally, the FullMesh algorithm can establish the full mesh of subflows [20]. All available paths are used for transmission, and all subflows are used for transmission concurrently.

The previous studies of our research group have shown that in heterogeneous multi-network integration scenarios, the default algorithm does not necessarily have a significant performance improvement in comparison to TCP; the NDiffPorts and Binder algorithms are only useful in special scenarios, while the FullMesh algorithm can provide bandwidth aggregation benefits [12]. The research shows that the Impact Factor (IF) of the subflow can be used to quantitatively explain the contribution rate of the current subflow to the overall transmission performance of the network communication. The IF of a subflow is defined by the importance of each subflow to the whole transmission in the concurrent transmission of multiple subflows. In other words, it refers to the degree of influence on the throughput of the whole flow, which can be divided into positive or negative effects. A subflow plays a negative role in the overall transmission performance when the overall throughput *increases* while the subflow is *not* participating in the transmission. On the contrary, a subflow plays a positive role in the overall transmission performance when the overall throughput *decreases* without usage of this subflow. Therefore, for all subflows, we proposed a quantitative description of the IF of the i-th subflow on the overall transmission, defined as Ω_i:

$$\begin{aligned}\Omega_i &= \frac{\sum_{t=0}^{T}\mathrm{TP}(t)-\sum_{t=T+1}^{2T}\mathrm{TP}_i(t)}{\sum_{t=0}^{T}\mathrm{TP}(t)}\\ &= 1-\frac{\sum_{t=T+1}^{2T}\mathrm{TP}_i(t)}{\sum_{t=0}^{T}\mathrm{TP}(t)},\end{aligned} \tag{1}$$

where $\mathrm{TP}(t)$ is the throughput at time t, $\mathrm{TP}_i(t)$ is the throughput at time when the i-th subflow does not participate in transmission, and T is the transmission time. Ω_i represents the contribution rate of the subflow to the overall transmission. For the subflow whose IF value is less than 0, it is not recommended to add transmission to avoid performance degradation. All $\Omega_i, i \in \{1,\ldots,N\}$ can be defined as a set X.

III. Small Data Stream Transmission

There are two main reasons for the delay of MPTCP small data stream transmissions: retransmission timeouts and different characteristics of subflows.

In TCP, the missing packets will be resent through the retransmission mechanism. There are two retransmission mechanisms for TCP [21], namely fast retransmission and timeout retransmission. [8, Section 2.9] provides a detailed description of these protocol mechanisms. The retransmission mechanisms for MPTCP are similar to those of TCP. If there is a packet loss in a certain subflow and the data transmitted is small, the sending window maintained by each subflow is very small. When there is a packet loss, a Retransmission Timeout (RTO)

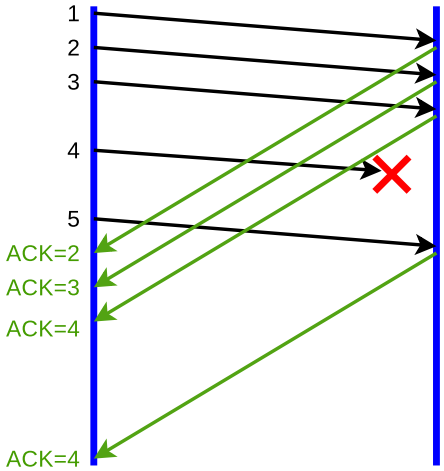


Figure 2. Retransmission Timeout leading to increased Completion Time

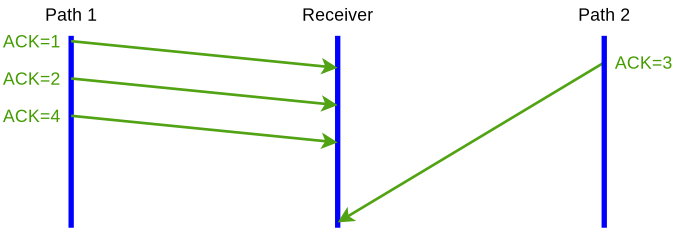


Figure 3. Data Waiting caused by Path Dissimilarity

will occur due to the receiver's failure to send an Acknowledgement (ACK) as reception confirmation. If at this time the sender cannot start the fast retransmission mechanism, it can only wait for the timeout retransmission. However, the data retransmission through the timeout retransmission mechanism will increase the completion time of this transmission. This is shown in Figure 2: when packet 4 is lost, fast retransmission cannot be started because the receiver fails to send three repeated ACK=4, which can only trigger the timeout retransmission to cause the RTO phenomenon, leading to the increase of the small data stream transmission time. For the transmission of a small data stream, if MPTCP starts all the subflows to transmit concurrently, it will increase the probability of packet loss and cause the RTO phenomenon. Therefore, for the transmission of small data streams without high throughput requirements, increasing the transmission completion time should be avoided.

On the other hand, when the characteristics of the subflows (i.e. bandwidth, delay, loss rate) are greatly different, the arrival times of the packets transmitted by each subflow will also be significantly different, resulting in the packets having arrived earlier to wait for the packets that arrive later. This also causes an increase of the completion time for small data streams. As shown in Figure 3, packets 1, 2 and 4 on path 1 have arrived, and the round-trip time (RTT) for packet 3 on path 2 is much longer than on path 1. Therefore, packets 1, 2 and 4 need to wait for packet 3 to complete

the data transmission. For small data stream transmission, the performance loss can be large, due to long waiting times.

IV. The PCDC Algorithm

The PCDC algorithm is an improvement of the original FullMesh algorithm, based on the subflow characteristics and the transmitted data size characteristics, which involves four key steps:

1) Compute the IFs Ω_i using Equation 1;
2) Create set X and classify the available subflow set into optional subset P and standby subflow subset, according to their IF values, and then sort the set P according to the RTT;
3) Classify the payload data according to the required transmitted data size;
4) Select the transmission subflow(s) in set P according to the data category.

According to the IF value, we can predict the contribution of each subflow to the overall network performance. When the IF is greater than 0, which means that the subflow plays a positive role in the overall transmission, it is classified as optional subflow. When the IF is less than 0, the subflow is classified as standby subflow. After the subflow classification, all optional subflows are sorted according to the RTT of each subflow, i.e., the elements in set P are arranged from small to large, according to their corresponding RTT values.

The transmitted data streams are classified into three classes, writing it in terms of a set Y, according to the transmitted data size.

Once the class of the current data stream to be transmitted is obtained, the appropriate subflow for transmission is selected according to the class: It belongs to class L_1 when the data flow size S_i range $0 < S_i \leq 512$ KiB, and the first subflow is selected for transmission, i.e., the first element of set P. It belongs to class L_2 when the data flow size S_i range 512 KiB $< S_i \leq$ 1024 KiB, and the first two subflows are selected for transmission, i.e., the first two elements of set P. Otherwise, the data stream belongs to class L_3, and all optional paths are selected for transmission.

Mathematically, the PCDC algorithm can be described as a piecewise function:

$$\begin{aligned} F &: X \times Y \to P \\ X &:= \{\Omega_i\} \\ Y &:= \{L_1, L_2, L_3\} \end{aligned}$$

$$F(x,y) := \begin{cases} P[a] & \text{when } x > 0, y = L_1 \\ P[a] \cup P[b] & \text{when } x > 0, y = L_2 \\ \cup_{m \in A} P[m] & \text{when } x > 0, y = L_3 \end{cases} \quad (2)$$

where

$$\begin{aligned} A &:= \{1, 2, \ldots, N\} \setminus \{i | \Omega_i < 0\}, \\ a &:= \min\{m \in A | \text{sort RTT}[m]\}, \\ b &:= \min\{m \in A | \text{sort RTT}[m] \setminus \{a\}\}. \end{aligned}$$

We define F as a mapping from $X \times Y$ to P, where $X \times Y$ is a Cartesian product of X and Y. X denotes a set of three classes $\{L_1, L_2, L_3\}$, Y is a set of IFs, and the image P is the output subflows. $x \in X$ and $y \in Y$ are two independent variables. $P[a]$ and $P[b]$ are the subflows with first and second minimum RTT, respectively. A presents a set of subflow indices, ranging from 1 to N, without those whose IFs are negative. sort RTT$[m]$ defines an ordered sequence of $\{\text{RTT}[m] | m \in A\}$ from small to large.

V. Measurement Scenario Design

In the following, we study the performance of the proposed PCDC algorithm by experiments in the NorNet Core testbed [22]–[28]. Therefore, we run measurements on the distributed and programmable nodes of NorNet Core[4], which are spread over 23 sites on four continents. These sites are connected to multiple different ISPs with different access technologies, and most of them deploy IPv4 and IPv6. Figure 4 shows the structure of the NorNet Core testbed. In order to show the effectiveness of the PCDC algorithm, considering geographical distribution and heterogeneity of networks [22], [29], we choose four sites; their detailed information is listed in Table I. In this setup, we design three measurement scenarios.

The bandwidth measurements have been performed by applying the NetPerfMeter[5] [30], [31], [8, Section 6.3] tool, which provides the performance comparison of multiple transport connections and protocols, including MPTCP support. In all measurement scenarios, the following Linux kernel setups are used:

- Ubuntu Linux 16.04 "Xenial Xerus" LTS with Linux kernel version 4.19.128,
- Linux MPTCP version 0.95,
- Buffer size limit set to 16 MiB, to prevent throughput limitations by lack of buffer space [4].

VI. Results Analysis

In this paper, we have chosen three site relations from the NorNet Core sites in Table I. The first two design scenarios (in Subsection VI-A and Subsection VI-B) are both located in the same country in different cities. For the third scenario (in Subsection VI-C), we use an inter-continental relation. The performance evaluation metrics in the test include the effective throughput of the network transmitting a large data stream and the completion time for transmitting small data streams. The transmission of the large data stream does not limit the size of the transmitted data. The effective throughput of the network is measured within a certain time (120 s), while the small data stream is transmitted according to different sizes of data, and the transmission completion time is evaluated. In order to avoid experimental error, the average value of at least 10 measurement runs is taken. Two CC algorithms, Cubic [32] and OLIA [33] (Opportunistic Linked Increases Algorithm), are

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

[5]NetPerfMeter: https://www.uni-due.de/~be0001/netperfmeter/.

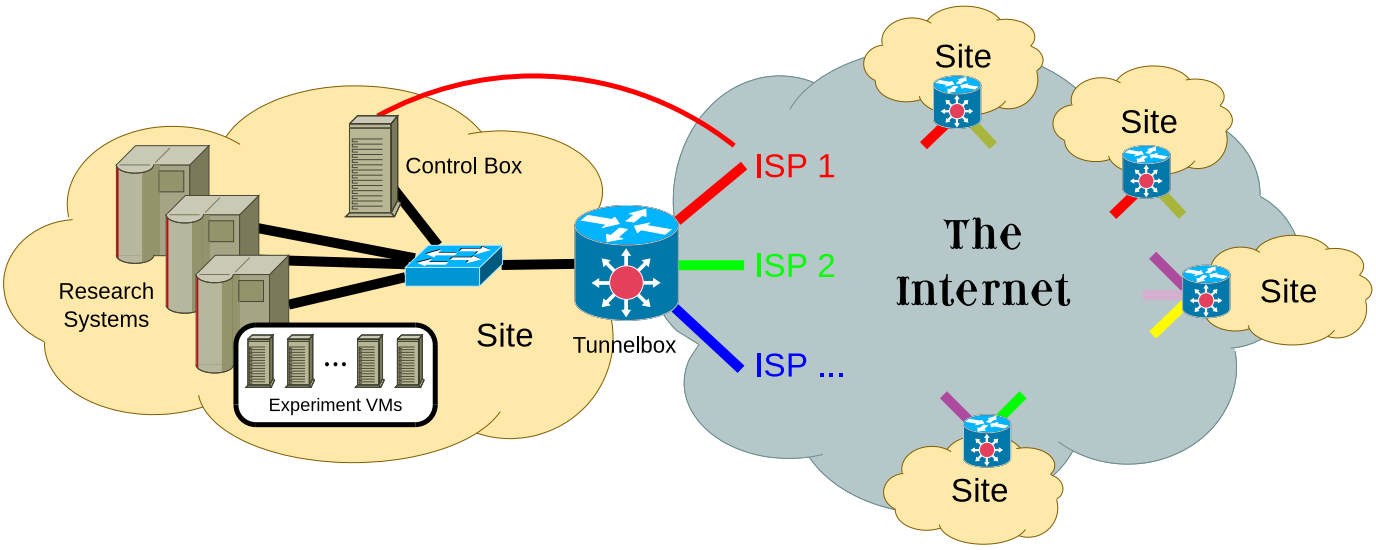


Figure 4. The Structure of NorNet Core Testbed

Table I
THE NORNET CORE SITES USED FOR THE MEASUREMENTS IN THIS PAPER

Site	Abbreviation	Location (City, Province, Country)	ISP	Bandwidth (Down/Up) Kbit/s
Hainan University	HU	Haikou, Hainan, China	CERNET China Unicom	10000 / 10000 10000 / 10000
Høgskolen i Narvik	HiN	Narvik, Nordland, Norway	Uninett PowerTech Broadnet	1000000 / 1000000 2000 / 128 2000 / 512
Universitetet i Bergen	UiB	Bergen, Vestland, Norway	Uninett BKK	1000000 / 1000000 100000 / 100000
Universitetet på Svalbard	UNIS	Longyearbyen, Svalbard, Norway	Uninett Telenor	100000 / 100000 10000 / 10000

considered for comparison. Cubic is the default CC algorithm used by Linux for TCP and MPTCP, while OLIA is the main coupled CC algorithm of Linux MPTCP.

A. UiB to UNIS

The distance between the site of UiB and the site of UNIS (see Table I) is around 2000 km. Particular to mention is that UNIS is located in Longyerbyen, on the remote island of Spitsbergen, just around 1200 km from the North Pole. Uninett is the Norwegian research network ISP (fibre), while BKK is a commercial ISP providing a business-grade fibre connection. Telenor provides a consumer-grade fibre connection. For FullMesh and PCDC, the total number of subflows is 4, and the names of subflows are: Uninett-Uninett, Uninett-Telenor, BKK-Uninett, and BKK-Telenor. The measurement runtime is 120 s.

First, we measured the IF Ω of each subflow using Equation 1. The results are shown in Table II, for each CC and combination of ISPs. Each measurement has been performed 10 times, i.e. the table presents the mean Ω_{Mean}, the median Ω_{Median}, absolute minimum and maximum (Ω_{Min}, Ω_{Max}), as well as 10% and 90% quantiles (Ω_{Q10}, Ω_{Q90}). Furthermore, it contains the mean RTT. As shown, mostly the research network Uninett-Uninett subflow has the highest mean impact factor Ω_{Mean}. It also has the lowest mean RTT. The contribution of all other subflows is considerably lower and in many cases slightly negative.

Based on the results from Table II, PCDC chooses only the Uninett-Uninett subflow for Cubic, as well as the Uninett-Telenor subflow in addition for OLIA. To give a brief overview (just *one* example run), Figure 5 presents the application payload throughput for Cubic (left-hand side) and OLIA (right-hand side) with both, the regular FullMesh and our PCDC path managers. Clearly, due to the better choice of subflows, PCDC performs better in *this* 120 s example for both CC algorithms. Particularly, OLIA with PCDC performs significantly better here. But what about the more general case?

In order to provide further insights into the behaviour of PCDC vs. regular FullMesh, Figure 6 presents the average results of at least 10 measurement runs for both path managers and both CC algorithms. In addition to the average (main bars), the thin error bars present the range from absolute minimum to maximum, and the thick error bars show the range from 10% quantile to 90% quantile. As it can be seen, the performance of PCDC for Cubic is slightly better (242.4 Mbit/s vs. 217.1 Mbit/s, i.e. 11.65% improvement),

Table II
IMPACT FACTOR Ω FOR UNIVERSITETET I BERGEN (UIB) TO UNIVERSITETET PÅ SVALBARD (UNIS)

CC	From ISP	To ISP	Samples	Ω_{Mean}	Ω_{Median}	Ω_{Min}	Ω_{Max}	$\Omega_{10\%}$	$\Omega_{90\%}$	RTT [ms]
Cubic	BKK	Telenor	10	-0.04	-0.02	-0.09	-0.00	-0.06	-0.01	104.2
Cubic	BKK	Uninett	10	-0.06	-0.09	-0.27	0.25	-0.17	0.11	50.1
Cubic	Uninett	Telenor	10	-0.01	0.01	-0.14	0.05	-0.08	0.04	61.3
Cubic	Uninett	Uninett	10	0.68	0.66	0.63	0.89	0.64	0.73	39.6
OLIA	BKK	Telenor	10	-0.02	-0.00	-0.14	0.03	-0.09	0.03	82.7
OLIA	BKK	Uninett	10	-0.02	-0.01	-0.12	0.06	-0.08	0.04	42.8
OLIA	Uninett	Telenor	10	0.03	0.04	-0.06	0.12	-0.04	0.11	73.1
OLIA	Uninett	Uninett	10	0.44	0.40	0.28	0.71	0.30	0.70	37.4

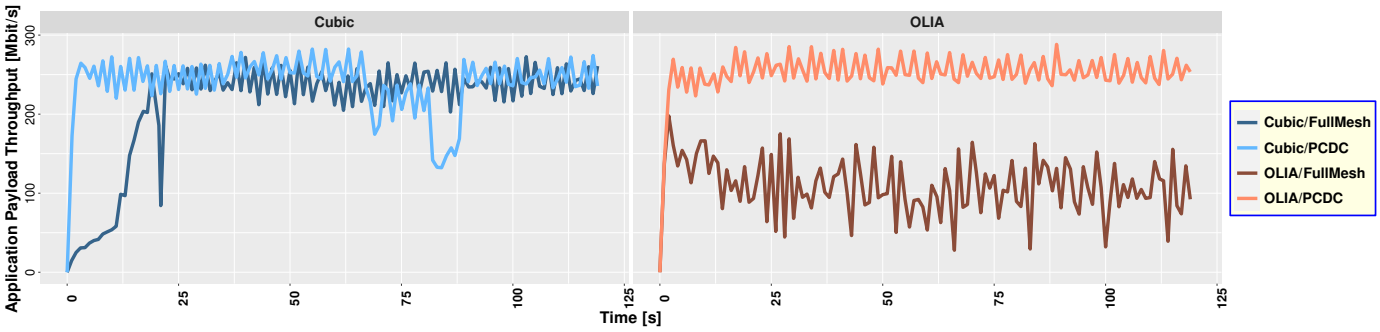


Figure 5. Vector Example for UiB to UNIS

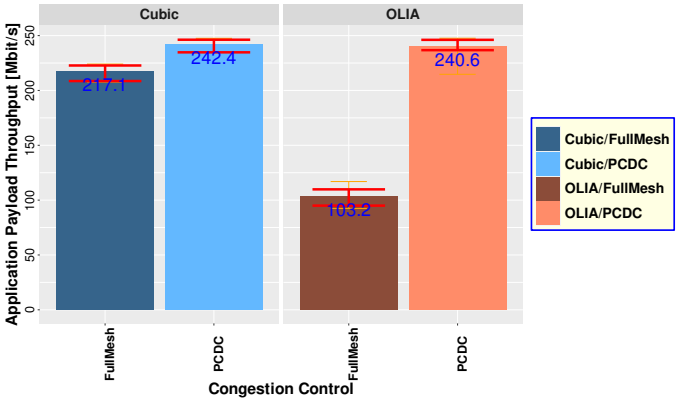


Figure 6. Throughput Comparison for UiB to UNIS

while the performance for OLIA is significantly increased (240.6 Mbit/s vs. 103.2 Mbit/s, i.e. 133.14% improvement). So, why is OLIA so significantly improved by PCDC?

OLIA, as a coupled congestion control algorithm [10], has to assume that paths may not be disjoint and instead have shared bottlenecks [8][Chapter 8], [34]. To ensure fairness on shared bottlenecks, losses on one subflow may lead to an overall rate reduction. Therefore, "bad" subflows lead to a reduced overall performance. By not using the subflows with a negative impact factor Ω, PCDC avoids such subflows. They can therefore not negatively affect the congestion control behaviour of the coupled OLIA CC.

So, while PCDC can clearly improve the performance for large data streams, e.g. like video streaming, huge package downloads, etc., what about small data streams of up to 1 MiB? Typical data of such streams are e.g. web page parts (like HTML pages, graphics images, JavaScript files, etc.). Equation 2 of PCDC therefore makes a distinction for small data streams. Figure 7 shows the average completion time for small streams of given sizes (in KiB on the x-axis) for using Cubic (left-hand side) and OLIA (right-hand side) with PCDC and regular FullMesh path managers. Besides the average completion time, the plot also displays absolute minimum and maximum (as thin error bars in orange colour) as well as 10% and 90% quantiles (as thick error bars in red colour).

For Cubic, it can clearly be observed that PCDC achieves a significant improvement over regular FullMesh. But in particular, it is also visible that the completion times are more stable. While there is a significant variation, from e.g. around 200 ms to 500 ms for 768 KiB with regular FullMesh path manager, the interval is just slightly around 206 ms for PCDC. Of course, PCDC only uses the Uninett-Uninett subflow (due to the results for Ω in Table II).

For OLIA, there is not much difference between the PCDC and regular FullMesh path managers. Since OLIA has to assume shared bottlenecks on the subflows, it increases the congestion windows more carefully, leading to a lower throughput. That is, for small data streams, the difference between the two path managers with coupled OLIA CC remains small.

Furthermore interesting is also the comparison between Cubic and OLIA, independently of the path manager: for small data streams, OLIA performs slightly better than Cubic. The reason is that Cubic performs a Slow Start procedure (see [8, Section 2.11] for detailed description) on all sub-

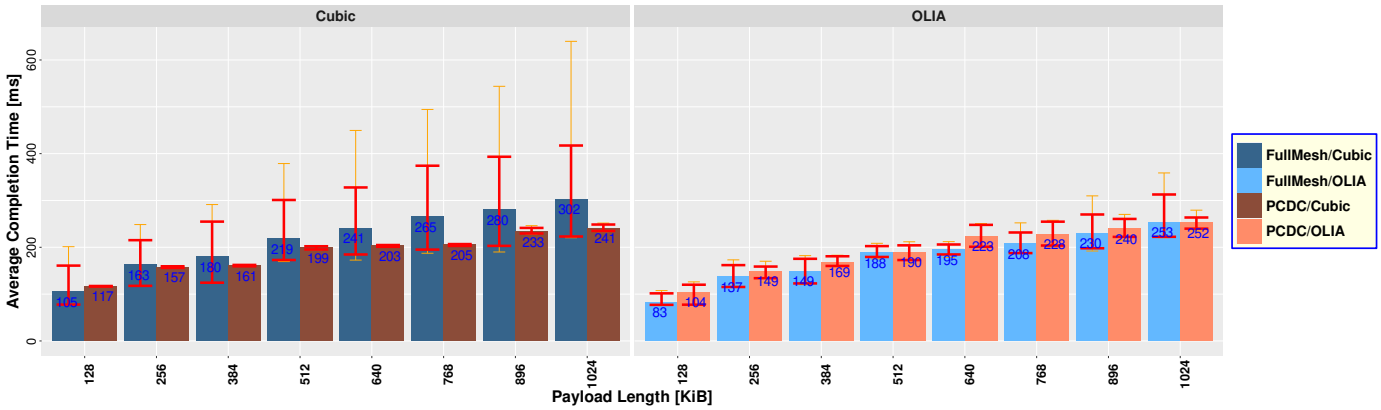


Figure 7. Average Completion Time for UiB to UNIS

flows in parallel, while OLIA takes shared bottlenecks into consideration when starting the transmission. Therefore, Cubic quickly runs into the expected packet losses on different subflows, due to reaching a path's capacity, before going into Congestion Avoidance mode (see [8, Section 2.11] for detailed description). OLIA can likely avoid some of these losses, decreasing the overall completion time for the very short-lived flows (between around 80 ms and 300 ms). In general [10], i.e. for large data streams, OLIA achieves a lower throughput than Cubic due to its less-aggressive CC behaviour.

B. UiB to HiN

In the next scenario, we examine the performance between UiB and HiN (see Table I). The distance between the two sites is approximately 1050 km. Both are located on the mainland of Norway. Particular property of the HiN site is that, in addition to its connection to the research network ISP Uninett, it is equipped with consumer-grade ADSL connections from PowerTech and Broadnet. Together with the research network (Uninett) and business-grade (BKK) connections at UiB, this scenario provides a very heterogeneous access technology set. Total number of subflows is 6, and the subflows are named as: Uninett-Uninett, Uninett-PowerTech, Uninett-Broadnet, BKK-Uninett, BKK-PowerTech, and BKK-Broadnet.

Table III presents the measured values of the IF Ω, as well as the mean RTT for Cubic and OLIA CCs with PCDC and regular FullMesh path managers. Also note particularly the heterogeneous mean RTTs, ranging from approximately 22 ms to over 60 ms. For Cubic, PCDC has a positive Ω_{Mean} for BKK-PowerTech, Uninett-Uninett, Uninett-PowerTech and Uninett-Broadnet. For OLIA, this is just the case for Uninett-Uninett.

Figure 8 shows the comparison between PCDC and regular FullMesh path managers for large data streams (120 s). As expected from the results for the first scenario in Subsection VI-A, PCDC improves the average application payload throughput over regular FullMesh: about 236.9 Mbit/s vs. 220.5 Mbit/s for Cubic, and even 240.3 Mbit/s vs. 136.5 Mbit/s

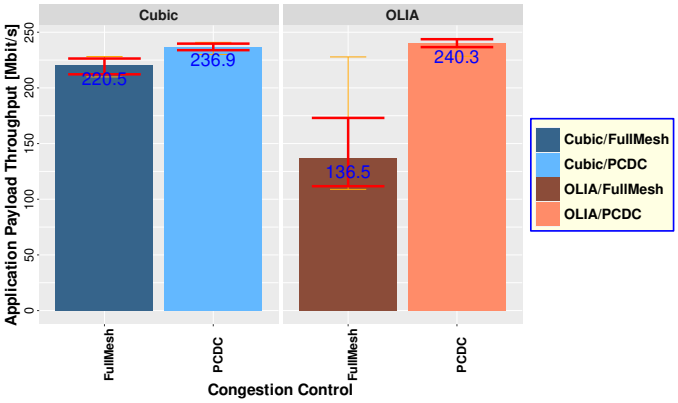


Figure 8. Throughput Comparison for UiB to HiN

for OLIA. Note that, due to just using Uninett-Uninett for OLIA with PCDC (see Table III), there is not much variation any more (thin orange error bars with absolute minimum and maximum, thick red error bars with 10% and 90% quantiles). Clearly, using all "bad" subflows with regular FullMesh path manager has a negative impact (as explained in Subsection VI-A) that also leads to a significant performance variation.

So, while PCDC again improves the performance for large data streams, what about small data streams? In Figure 9, we compare the performance for stream sizes from 128 KiB to 1024 KiB (see also Equation 2). Similar to the results for the first scenario in Subsection VI-A, there is a slight reduction of the average completion time when using PCDC instead of the regular FullMesh path manager. Particularly, PCDC also reduces the variation of the completion times (i.e. the thin orange error bars with absolute minimum and maximum, and the thick red error bars with 10% and 90% quantiles). For OLIA, there is almost no difference. And again, OLIA for small streams achieves a slightly lower completion time than Cubic.

Table III
IMPACT FACTOR Ω FOR UNIVERSITETET I BERGEN (UIB) TO HØGSKOLEN I NARVIK (HIN)

CC	From ISP	To ISP	Samples	Ω_{Mean}	Ω_{Median}	Ω_{Min}	Ω_{Max}	$\Omega_{10\%}$	$\Omega_{90\%}$	RTT [ms]
Cubic	BKK	Broadnet	10	-0.01	-0.02	-0.07	0.06	-0.05	0.03	46.7
Cubic	BKK	PowerTech	10	0.02	-0.00	-0.07	0.17	-0.05	0.09	55.3
Cubic	BKK	Uninett	10	-0.06	-0.04	-0.24	0.14	-0.14	-0.00	61.1
Cubic	Uninett	Broadnet	10	0.04	0.04	-0.01	0.11	-0.01	0.07	40.8
Cubic	Uninett	PowerTech	10	0.00	0.03	-0.15	0.10	-0.06	0.08	50.5
Cubic	Uninett	Uninett	10	0.57	0.58	0.51	0.61	0.53	0.60	24.3
OLIA	BKK	Broadnet	10	-0.02	-0.04	-0.09	0.10	-0.07	0.02	45.4
OLIA	BKK	PowerTech	10	-0.01	-0.00	-0.07	0.03	-0.05	0.03	53.7
OLIA	BKK	Uninett	10	-0.04	-0.03	-0.28	0.14	-0.10	0.03	24.3
OLIA	Uninett	Broadnet	10	-0.02	-0.04	-0.13	0.07	-0.08	0.05	38.6
OLIA	Uninett	PowerTech	10	-0.04	-0.01	-0.17	0.05	-0.15	0.03	41.4
OLIA	Uninett	Uninett	10	0.25	0.15	0.06	0.64	0.10	0.60	22.4

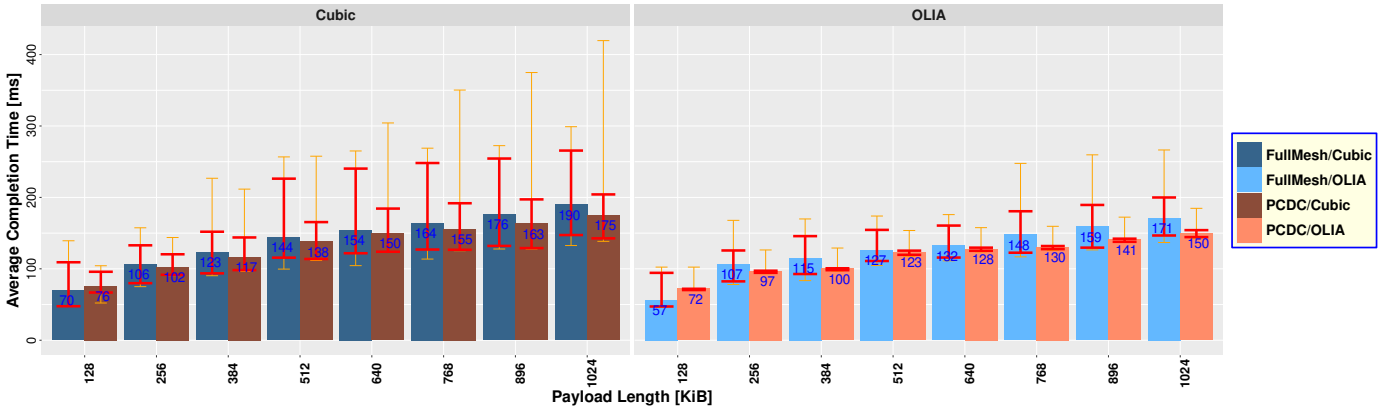


Figure 9. Average Completion Time for UiB to HiN

Table IV
IMPACT FACTOR Ω FOR UNIVERSITETET I BERGEN (UIB) TO HAINAN UNIVERSITY (HU)

CC	From ISP	To ISP	Samples	Ω_{Mean}	Ω_{Median}	Ω_{Min}	Ω_{Max}	$\Omega_{10\%}$	$\Omega_{90\%}$	RTT [ms]
Cubic	BKK	CERNET	10	-0.00	0.08	-0.65	0.35	-0.47	0.33	411.3
Cubic	BKK	CnUnicom	10	-0.05	0.07	-0.73	0.36	-0.48	0.31	346.2
Cubic	Uninett	CERNET	10	0.55	0.68	-0.34	0.97	0.19	0.84	353.0
Cubic	Uninett	CnUnicom	10	-0.18	-0.12	-1.03	0.17	-0.43	0.07	316.4
OLIA	BKK	CERNET	10	0.22	0.24	-0.05	0.41	0.06	0.37	413.1
OLIA	BKK	CnUnicom	10	-0.23	-0.23	-0.55	0.05	-0.52	0.03	345.7
OLIA	Uninett	CERNET	10	0.53	0.55	0.17	0.69	0.42	0.67	325.7
OLIA	Uninett	CnUnicom	10	-0.23	-0.24	-0.74	0.33	-0.46	0.03	315.8

C. UiB to HU

In the last scenario, we examine a corner case: an inter-continental transmission between UiB in Norway and HU in China (see Table I). The geographical distance between the two sites is around 8000 km, while the network communication [20], [22] can take paths from Europe westwards via North America to Asia, as well as from Europe eastwards directly to Asia. HU is connected via the research network ISP CERNET and the consumer ISP China Unicom (CnUnicom). Therefore, the total number of subflows is 4, and the subflows are named as: Uninett-CERNET, Uninett-CnUnicom, BKK-CERNET, BKK-CnUnicom.

Table IV shows the values for the IF Ω obtained from 10 measurement runs, i.e. mean Ω_{Mean}, median Ω_{Median}, absolute minimum and maximum (Ω_{Min}, Ω_{Max}), as well as 10% and 90% quantiles (Ω_{Q10}, Ω_{Q90}). Particularly note the high variation of the values, indicating the strong heterogeneity and volatility of the network path characteristics. Note also the average RTT and its variation from around 315 ms to 413 ms. According to Equation 2, PCDC will use only Uninett-CERNET for Cubic, while it will use Uninett-CERNET and BKK-CERNET for OLIA.

Figure 10 presents the application payload throughput comparison for Cubic and OLIA used with the PCDC and regular

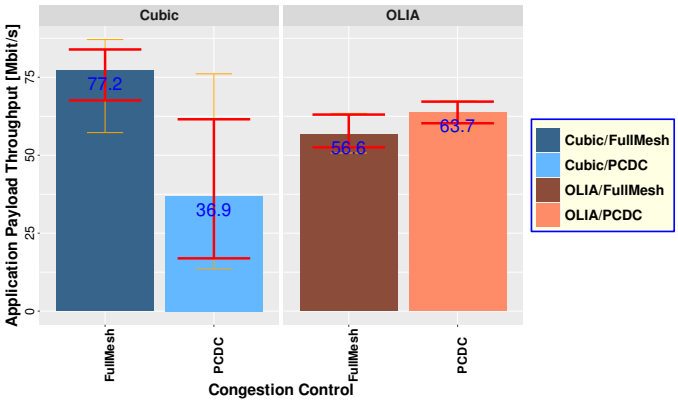


Figure 10. Throughput Comparison for UiB to HU

FullMesh path managers. As expected from the previous scenarios in Subsection VI-A and Subsection VI-B, there is a performance increase by PCDC for OLIA: from 56.6 Mbit/s to 63.7 Mbit/s. However, the performance for Cubic drops from 77.2 Mbit/s to 36.9 Mbit/s. The reason here is the high volatility of the Uninett-CERNET subflow, which is the only subflow used by PCDC for Cubic. We observed that, while Uninett-CERNET *usually* provides the best performance of all paths, there are sometimes performance drops – caused by congestion and likely some bandwidth limitations either at the HU site and/or at the Great Firewall of China. In this scenario, OLIA uses the BKK-CERNET subflow as well. This becomes beneficial in case of variations of the Uninett-CERNET subflow performance. As part of our ongoing work on PCDC, we are investigating possibilities to more quickly adapt PCDC to performance variations of network paths in such corner cases.

Finally, we again examined the performance for small data streams with sizes ranging from 128 KiB to 1024 KiB. The results are presented in Figure 11. Again, we can see that OLIA for small data streams provides a better performance than Cubic, as explained in Subsection VI-A. Note in particular that in this corner case, a major fraction of the total completion time is made up by the overhead caused by the MPTCP connection establishment and increasing the congestion window until reaching the Congestion Avoidance phase. That is, the completion time for 1024 KiB is around 3.4 s for Cubic and around 2.2 s for OLIA, due to the high mean RTTs in this inter-continental setup (around 400 ms, see Table IV). For OLIA, the completion times of PCDC and regular FullMesh path managers are very similar, while they are slightly increased – as explained above – for Cubic.

VII. Conclusions and Outlook

The FullMesh path management algorithm for MPTCP can establish the full mesh of subflows, and provide bandwidth aggregation benefits. All available subflows are used for transmission, and all subflows are transmitted concurrently. However, not all subflow additions can improve network performance, especially in heterogeneous multi-network integration scenarios.

In this paper, we introduced the concept of a subflow impact factor, and analyze the reasons for the poor performance of small data streams over MPTCP. Then, we proposed our Path Characteristic and Data Characteristic (PCDC) algorithm, which uses the impact factor as a subflow characteristic to predict the impact of the subflow on the overall throughput, and adopts different subflow sets for different sizes of traffic. Finally, the results of three real-world scenarios in the NorNet Core testbed show that, compared to FullMesh, PCDC can improve the overall payload throughput, and reduce the completion time of small data streams.

As part of our ongoing and future work, we are currently analyzing the PCDC performance in further detail with additional congestion control algorithms and different buffer size settings. Furthermore, we are working on improving PCDC in corner cases, like the examined inter-continental setup with its very dissimilar paths and highly volatile path characteristics. Particularly, we also intend to extend PCDC with analysis based on machine learning, in order to further optimise the analysis of path performance metrics, the choice of paths, as well as the scheduling of data onto these paths.

References

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

[2] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath TCP: Analysis, Design, and Implementation," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, p. 59609, Feb. 2016, ISSN 1063-6692.

[3] Q. Tan, X. Yang, L. Zhao, X. Zhou, and T. Dreibholz, "A Statistic Procedure to Find Formulae for Buffer Size in MPTCP," in *Proceedings of the 3rd IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing/People's Republic of China, Oct. 2018, pp. 900–907, ISBN 978-1-5386-4509-3.

[4] 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, pp. 9–16, ISBN 978-1-5090-6028-3.

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

[6] R. Barik, M. Welzl, G. Fairhurst, T. Dreibholz, A. M. Elmokashfi, and S. Gjessing, "On the Usability of Transport Protocols other than TCP: A Home Gateway and Internet Path Traversal Study," *Computer Networks*, vol. 173, May 2020.

[7] 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.

[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] 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-20, Jul. 2020.

[10] 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.

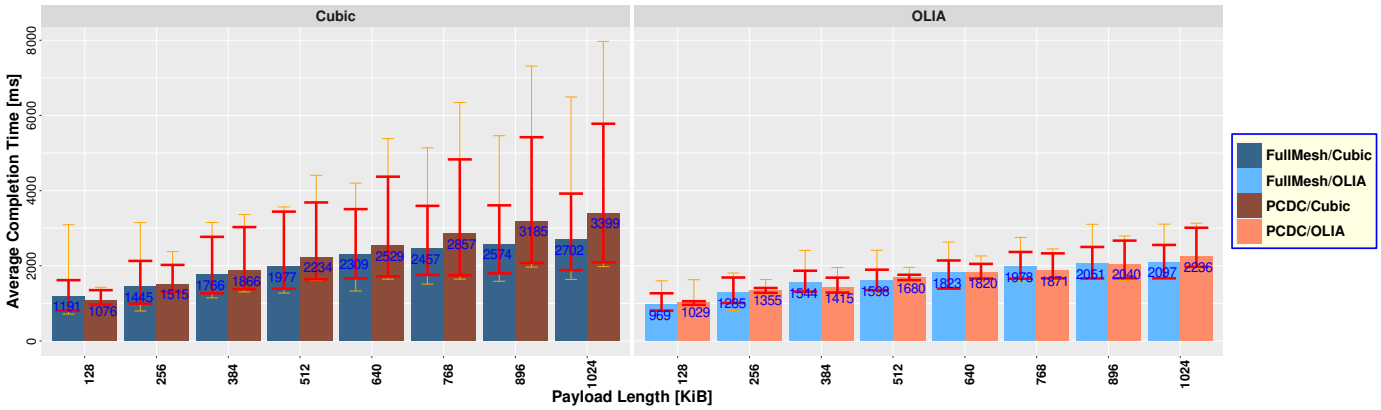


Figure 11. Average Completion Time for UiB to HU

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

[12] 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, pp. 1–8, ISBN 978-1-5090-6028-3.

[13] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A Scalable and Flexible Data Center Network," *Communications of the ACM*, vol. 54, no. 3, p. 9504, Mar. 2011, ISSN 0001-0782.

[14] Z. Zhang, J. Li, and X. Chang, "Longitudinal Study on Evolution of Internet Traffic," *Application Research of Computers*, vol. 32, no. 11, pp. 3215–3221, 2015.

[15] S. Wang, "Traffic Scheduling for Cloud Data Centers," Ph.D. dissertation, Beijing University of Posts and Telecommunications, 2018.

[16] M. Kheirkhah, I. Wakeman, and G. Parisis, "MMPTCP: A Multipath Transport Protocol for Data Centers," in *Proceedings of the the 35th Annual IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2016, pp. 1–9.

[17] 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.

[18] 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.

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

[20] 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.

[21] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," IETF, Standards Track RFC 5681, Sep. 2009.

[22] T. Dreibholz, "HiPerConTracer - A Versatile Tool for IP Connectivity Tracing in Multi-Path Setups," in *Proceedings of the 28th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Hvar, Dalmacija/Croatia, Sep. 2020.

[23] ——, "NorNet at the University of Sydney: From Simulations to Real-World Internet Measurements for Multi-Path Transport Research," Invited Talk at University of Sydney, Sydney, New South Wales/Australia, Jan. 2019.

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

[25] 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.

[26] T. Dreibholz, S. Ferlin, Özgü Alay, A. M. Elmokashfi, I. A. Livadariu, and X. Zhou, "MPTCP Experiences in the NorNet Testbed," IETF, Individual Submission, Internet Draft draft-dreibholz-mptcp-nornet-experience-05, Dec. 2019.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.