SCISPACE
formerly Typeset

Open access • Proceedings Article • DOI:10.1109/GLOCOM.2013.6831263

Comparison of Multipath TCP and CMT-SCTP based on intercontinental measurements — Source link

Martin Becke, Hakim Adhari, Erwin P. Rathgeb, Fu Fa ...+2 more authors

Institutions: University of Duisburg-Essen, Hainan University

Published on: 01 Dec 2013 - Global Communications Conference

Topics: Multipath TCP, Network congestion, Stream Control Transmission Protocol, The Internet and Testbed

Related papers:

- TCP Extensions for Multipath Operation with Multiple Addresses
- Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths
- Stream Control Transmission Protocol
- How hard can it be? designing and implementing a deployable multipath TCP
- On the fairness of transport protocols in a multi-path environment

Share this paper:

View more about this paper here: https://typeset.io/papers/comparison-of-multipath-tcp-and-cmt-sctp-based-on-1979fqwge4

Comparison of Multipath TCP and CMT-SCTP based on Intercontinental Measurements

Martin Becke, Hakim Adhari, Erwin P. Rathgeb
University of Duisburg-Essen,
Institute for Experimental Mathematics
Ellernstraße 29, 45326 Essen, Germany
{martin.becke, hakim.adhari, rathgeb}@uni-due.de

Fu Fa, Xiong Yang, Xing Zhou
Hainan University,
College of Information Science and Technology
Renmin Avenue 58, 570228 Haikou, China
{fufa, xyang, zhouxing}@hainu.edu.cn

***Abstract*—The market penetration of access devices with multiple network interfaces has increased dramatically over the last few years. As a consequence, there is a strong interest to use all of the available interfaces concurrently to improve data throughput. Corresponding extensions of established Transport protocols are receiving considerable attention within research and standardization.**

Currently two approaches are in the focus of the IETF: The Multipath TCP (MPTCP) extension for TCP and the Concurrent Multipath Transfer extension for SCTP (CMT-SCTP). This paper evaluates and compares implementations of these two loadsharing protocols by using both lab measurements and intercontinental testbed realized via the Internet between Europe and China. The experiments show that some performance critical aspects have not been taken into account in previous studies. Furthermore, they show that the simple scenario with two disjoint paths, which is typically used for evaluation, does not sufficiently cover the real Internet environment. Based on these insights, we highlight that the different path management strategies of the two protocols have a significant impact on their performance in real Internet scenarios.[1]

Keywords: Multipath Transfer, Loadsharing CMT-SCTP, MPTCP, Buffer, Congestion Control, Performance Analysis

I. Introduction and Related Work

Nowadays, the Internet is the predominant global communication infrastructure, increasing day by day in number of users as well as in diversity of services used. Network providers obviously aim at maximizing the utilization of the available network resources while users expect, among other QoS criteria, optimum and stable data throughput. In addition, it is a basic Internet policy to ensure that each user gets his fair share of network resources. As the network layer only provides a very simple delivery service, current transport protocols play a major role in striking this balance. This is exemplified by TCP with its elaborate congestion control which tries to optimize goodput, to limit network congestion and distribute available network capacity in a fair manner among competing connections.

This issue is sufficiently understood for the current Internet scenario where two endpoints are interconnected via a single network path, and the established solutions work reasonably well. However, multi-homing and in particular loadsharing over multiple network paths, create novel challenges which require significant research effort to be fully understood. These challenges have a major impact on the design of multipath protocols and their mechanisms.

[1]Partly funded by the National Natural Science Foundation of China (funding number 61163014)

Although loadsharing can be applied on various OSI layers, this paper - as well as current standardization efforts - focuses on approaches on the Transport layer since only the Transport protocol can easily provide a common service across the borders of provider networks [1]. Over the years, multiple approaches were proposed such as the Reliable Multiplexing Transport Protocol [2], Parallel TCP [3] or mTCP [4], but none of them has been deployed in the Internet. This could be changed by the approaches currently discussed in the context of the IETF. Multipath TCP (MPTCP [5]) and Concurrent Multipath Transfer for SCTP (CMT-SCTP [6]) are two approaches supporting loadsharing for end-to-end transport. Both protocols are in an advanced stage of the standardization in the IETF.

One major issue for multipath loadsharing in Transport protocols is the so called ”shared bottleneck” problem [7] arising if several paths of a multipath connection share a common bottleneck link and compete against classical TCP traffic - which can neither be avoided nor detected reliably. In this case, the multipath protocol would occupy more than the fair share of bandwidth on the bottleneck link if every path of the multipath connection would behave like an individual (TCP-)connection. Therefore, the idea of Resource Pooling [8] has been adopted for both MPTCP and CMT-SCTP [6] to provide TCP-friendliness [9] under all circumstances which is a basic prerequisite for standardization by the IETF and ubiquitous deployment in the Internet.

[10] describes the current MPTCP concept and most of the design decisions. The current effort to standardize CMT-SCTP [11] is based on ideas first published in [6] and recently improved and extended significantly [7], [9], [12]. Although the standardization process is, especially for MPTCP, fairly advanced, the behaviour of these protocols is not yet fully understood. Some simulative performance evaluations have already been published [6], first measurements have also been reported [7], [9], [12]–[15]. All these evaluations mainly focus on congestion control issues and use very simple network topologies while evaluations in real world Internet scenarios are still missing. Our contribution is to evaluate and compare the specific behavior of MPTCP and CMT-SCTP both in a lab environment and in a real world intercontinental Internet scenario. We will first show that some performance relevant aspects of real implementations have not been modeled sufficiently in simulations so far. In a next step we will highlight that the path management concept, which is different for MPTCP and CMT-SCTP, has significant performance implications in real Internet scenarios.

II. Basics

MPTCP and CMT-SCTP are extensions of the well-know protocols TCP [16] and SCTP [17] and are currently under standardization.

Both protocols feature multiple mechanisms like *Path Management* or *Congestion Control* (CC) that interact in a complex and sometimes nontransparent manner. Our evaluations [7], [9], [12] have clearly shown that all of these mechanisms, which have typically just been adopted from the singlepath protocols, have an impact on the performance of the multipath extensions. The common underlying problem to be solved is the lack of information about the status of the communication paths within the Internet – which becomes more crucial if the different paths are highly heterogeneous.

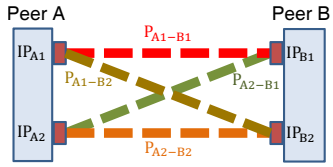
In the following, we will focus our discussion on those aspects and mechanisms which have the most severe performance impact in the intercontinental scenario we have set up.

A. Congestion Control

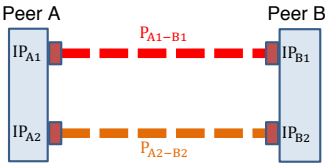
Finding a suitable Congestion Control (CC) mechanism able to handle multiple paths is non-trivial [9]. Simply adopting the mechanisms used for the singlepath protocols in a straight-forward manner does neither guarantee an appropriate throughput [9] nor achieve a fair resource allocation when dealing with multipath transfer [12]. To solve the fairness issue, Resource Pooling (RP) [8] has been adopted for both MPTCP and CMT-SCTP. In the context of RP, multiple resources (in this case paths) are considered to be a single, pooled resource and the CC focuses on the complete network instead of only a single path. As a result, the complete multipath connection (i.e. all paths) is throttled even though congestion occurs only on one path. This avoids the bottleneck problem described earlier and shifts traffic from more congested to less congested paths. Releasing resources on a congested path decreases the loss rate and improves the stability of the whole network. In [18] three design goals are set for RP-based multipath CCs for a TCP-friendly Internet deployment. These rules are:

1) *Improve throughput*: A multipath flow should perform at least as well as a singlepath flow on the best path.
2) *Do not harm*: A multipath flow should not take more capacity on any one of its paths than a singlepath flow using only that path.
3) *Balance congestion*: A multipath flow should move as much traffic as possible off its most congested paths.

The CC proposed for MPTCP was designed with these goals in mind already [18] . The CC of the original CMT-SCTP proposal did not use RP, but we already proposed an algorithm for CMT-SCTP which uses RP and fulfills the requirements (CMT/RPv2, [12]). This algorithm behaves slightly different from the MPTCP CC (see [18]) and, therefore, we also adapted the MPTCP CC to SCTP which will be called " MPTCP-like" in the following. While both mechanisms are still candidates for CMT-SCTP in the IETF discussion, we will only use the MPTCP-like algorithm in this paper to get an unbiased comparison with MPTCP. The MPTCP and MPTCP-like CCs treat each path as a self-contained congestion area and reduce just the *path* congestion window c_P of the path experiencing congestion. In order to avoid an unfair overall bandwidth allocation, the congestion window growth behavior of the CCs is adapted: a per-flow *aggressiveness factor* $\hat{a}$ is used to bring the increase and decrease of c_P into equilibrium.



(a) MPTCP



(b) CMT-SCTP

Figure 1. Paths combinations

The MPTCP CC [18] is based on counting bytes as TCP and MPTCP are byte-oriented protocols. SCTP, however, is a message-oriented protocol and the CC is based on counting bytes which are limited in size by the Maximum Transmission Unit (MTU). The limit for the calculation is defined as Maximum Segment Size (MSS) for TCP and SCTP. So it is, e.g., 1,460 bytes for TCP or 1,452 bytes for SCTP using IPv4 over an Ethernet interface with a typical MTU of 1,500 bytes. For each path P there are independent congestion window c_P , slow-start threshold s_P and partial acknowledgement p_P variables. To adapt the MPTCP CC to SCTP, instead of α acknowledged bytes on path P for a fully utilized congestion window, the MPTCP-like CC adapts c_P as follows:

$$c_P = c_P + \begin{cases} \min\left\{\left\lceil \frac{c_P * \hat{a} * \min\{\alpha, \mathrm{MSS}_P\}}{\sum_i c_i} \right\rceil, \min\{\alpha, \mathrm{MSS}_P\}\right\} \\ (c_P \leq s_P) \\ \min\left\{\left\lceil \frac{c_P * \hat{a} * \mathrm{MSS}_P}{\sum_i c_i} \right\rceil, \mathrm{MSS}_P\right\} \\ (c_P > s_P \wedge p_P \geq c_P) \end{cases}.$$

$\hat{a}$ denotes the per-flow *aggressiveness factor*, defined as:

$$\hat{a} = \left(\sum_i c_i\right) * \frac{\max_i \left\{\frac{c_i/\mathrm{MSS}_i}{(\mathrm{RTT}_i)^2}\right\}}{\left(\sum_i \frac{c_i/\mathrm{MSS}_i}{\mathrm{RTT}_i}\right)^2}.$$

B. Path Management

a) Path Management in MPTCP: A MPTCP connection consists, in principle, of several TCP-like connections (called subflows) using the different network paths available. A MPTCP connection between Peer A (P_A) and Peer B (P_B) (see Figure 1(a)) is initiated by setting up a regular TCP connection between the two endpoints via one of the available paths, e.g., IP_{A1} to IP_{B1}. During the connection setup, the new TCP option *MP_CAPABLE* is used to signal the intention to use multiple paths to the remote peer [5]. Once the initial connection is established, additional sub-connections are added. This is done similar to regular TCP connection establishment by performing a three-way-handshake with the new TCP option *MP_JOIN* present in the segment headers. By default MPTCP uses all available address combinations to set up subflows resulting in a full mesh using all available paths between the endpoints. The option *ADD_ADDR* is used

in the Linux implementation to announce an additional IP address to the remote host. In the case of Figure 1(a), the MPTCP connection is first set up between IP_{A1} and IP_{B1}. Both hosts then include all additional IP addresses in an *ADD_ADDR* option, since they are both multi-homed. After that, an additional subflow is started between IP_{A2} and IP_{B1} by sending a SYN packet including the *MP_JOIN* option. The same is done with two additional sub-connections between IP_{A2} and IP_{B2} as well as IP_{A1} and IP_{B2}. The result of these operations is the use of 4 subflows using direct as well as cross paths: P_{A1-B1}, P_{A1-B2}, P_{A2-B1} and P_{A2-B2}.

b) Path Management in CMT-SCTP: CMT-SCTP is based on SCTP as defined in [17]. Standard SCTP already provides multi-homing capabilities which are directly usable for CMT-SCTP. An SCTP packet is composed of an SCTP header and multiple information elements called *Chunks* which can carry control information (Control Chunks) or user data (DATA Chunk). A connection, denoted as *Association* in SCTP, is initiated by a four-way handshake and is started by sending an *INITIATION* (INIT) chunk. With this first message, the initiating host P_A informs the remote host P_B about all IP addresses available on P_A. Once P_B has received the INIT chunk it answers with an *INITIATION-ACKNOWLEDGMENT* (INIT-ACK) chunk. The INIT-ACK also includes a list of all the IP addresses available on P_B.
When P_A initiates an SCTP connection to P_B, it uses the "primary" IP addresses of both hosts IP_{A1} and IP_{B1} as source and destination address, respectively. This creates a first path between these two addresses, denoted as P_{A1-B1} in Figure 1(b) which is designated as "Primary Path". In standard SCTP this is the only path used for exchange of user data, the others are only used to provide robustness in case of network failures. SCTP, and consequently also CMT-SCTP, uses all additional IP addresses to create additional paths. In contrast to MPTCP, each secondary IP address is only used for a single additional path in an attempt to make the established paths disjoint. In the example, the secondary path P_{A2-B2} is established.

As a result, while the MPTCP creates a full mesh of possible network paths among the available addresses, CMT-SCTP only uses pairs of addresses to set up communication paths. CMT-SCTP only determines the specific source address to specify which path has to be used (source address selection) and leaves it to the IP layer to select the route to the next hop. MPTCP, however, maintains a table in the Transport layer identifying all possible combinations of local and remote addresses and uses this table to predefine the network path to be used.

III. Local Lab Testbed

The first measurements on CMT-SCTP performance [19] have used a testbed within Germany with fully disjoint paths via the German Research Network (DFN)[2] and a provider network via DSL, respectively. These measurements confirmed that CMT-SCTP achieves the performance predicted by simulations in such a scenario. To be able to get reproducible results for our comparison of MPTCP and CMT-SCTP we have set up a local lab testbed allowing to mimic this scenario while still having all relevant network parameters under full control.

[2]http://www.dfn.de

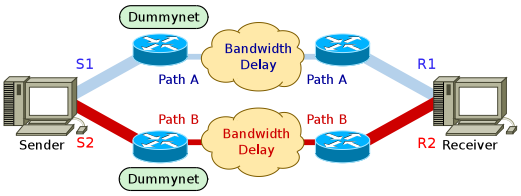


Figure 2. Local testbed in Essen/Germany

A. Local testbed

The topology of the local lab testbed, situated in Essen/Germany, is shown in Figure 2. For the measurements we used state-of-the-art computers, in this case Dell Optiplex 760 with 4 GB RAM and Intel Core™ 2 Duo CPU E8600 @3.33 GHz. Both hosts had a dual boot installation for both Linux (Ubuntu) and FreeBSD. This was necessary because only the Linux operating systems support the newest MPTCP kernel implementation [20] while the CMT-SCTP extension is only available in FreeBSD.

The hosts were interconnected via FreeBSD routers providing the disjoint paths shown in Figure 2 (Path A and Path B). To apply bandwidth limitations and delay variations, DUMMYNET [21] – which is part of the FreeBSD kernel – has been used on the routers. RED queues have been configured on the routers, using the parameters MinTh=30, MaxTh=90 and MaxP=10% (recommended by [22]).

B. Experiment parameters

For MPTCP, the MPTCP CC mechanism as defined in [18] has been used, CMT-SCTP used the MPTCP-like CC introduced in Section II-A to get an unbiased comparison. If not specified otherwise, the following configuration parameters have been used for measurements:

- The sender has been saturated (i.e. it has tried to transmit as much data as possible). All messages have used ordered, reliable delivery as provided by TCP.
- The measurement runtime has been 300 s, preceded by a transient phase of 20 s. Each run has been repeated at least 10 times in order to ensure a reasonable statistical accuracy.
- The result plots show the average values and their corresponding 95% confidence intervals.

C. Simple setup with bandwidth variation

We first considered a scenario with two disjoint paths. DUMMYNET, has been used on Path A to limit the bandwidth to 800 Kbit/s which corresponds to the maximum upload rate available on the DSL link in Essen. The bandwidth on Path B has been varied between 200 Kbit/s and 10 Mbit/s. No delay manipulations have been applied on the paths. Figure 3 shows the application payload throughput reached. Curve #1 and #2 show the throughput reached by CMT-SCTP and MPTCP respectively. The curve #3 is included for comparison purposes and shows the throughput of a non-CMT flow using single path TCP on Path B. The singlepath protocol reached the expected throughput over the full bandwidth range. The CMT flows (curve #1 and #2) reached a throughput which corresponds to the sum of both link bandwidths. This throughput increase confirms the advantage of multipath transfer. For this baseline scenario, both protocols were able to fulfill the multipath CC design goals defined in [18] (see Section II-A).

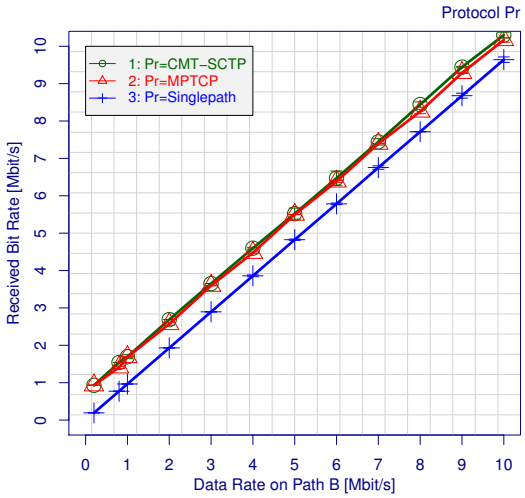


Figure 3. Local testbed – Dissimilar paths without any special delay

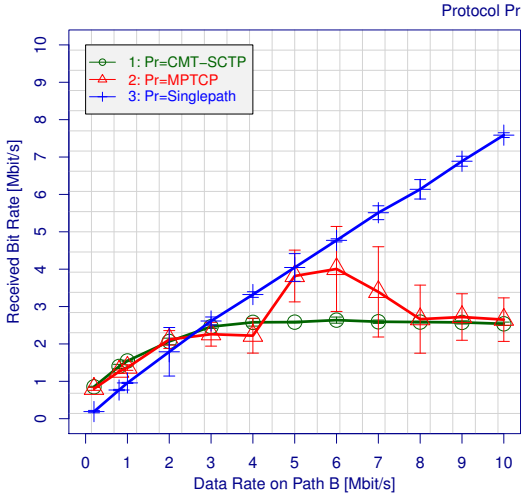


Figure 4. Local testbed – Dissimilar paths with 200 ms delay

D. Simple setup with additional delay

In a second step, we extended our parameter range to a scenario more realistic for wide area connections. We adopted the same bandwidth characteristics as in Section III-C. In addition, we set a delay of 200 ms on both links (value observed in the Internet measurements between Essen/Germany and Haikou/China (see Section IV-C)).

The results of this experiment are shown in Figure 4. For a low bandwidth setting on Path B we see the expected behavior for both multipath extensions. However, this is not the case when the paths become more dissimilar (bandwidth on Path B is higher than 2.5 Mbit/s while bandwidth on path A is 800 Kbit/s). In fact, the achievable throughput starts to saturate quickly and drops below the singlepath throughput.

This can be attributed to the high CPU load as we have observed CPU usage peaks of up to 100%. The reason for this high CPU load is that the protocols have to maintain lists of missing data (sequence number gaps) as both use Selective Acknowledgements (SACK) and lists for re-ordering. These lists have to be searched and updated for every received packet. The RED queues discard packets systematically and the increasing asymmetry between both paths amplifies the problem resulting in a tremendous growth of the lists and the CPU resources required to manage them. This effect is more dramatic for CMT-SCTP as this protocol has to maintain a second list for Non-Revokable SACKs (NR-SACK [23]) in order to avoid buffer blocking effects [7]. However, this is not really relevant as agrumentation for a protocol for the Internet as both protocol implementations clearly fail to perform as expected for highly asymmetrical links. It should be mentioned that this CPU limitation can not be observed in typical event-based simulation experiments. In this case, the high computation requirements cause only a longer simulation time without influencing the results. It is to be mentioned that increasing CPU capacities would not be enough to solve the problem since the costs of maintaining the lists increase in a disproportional way with the dissimilarity of the links. Therefore, an optimized list management or an alternative scheduling mechanism is required.

Based on these insights, we designed and conducted experiments in a global testbed to verify if any other limitations cause significant issues in a real-life Internet scenario.

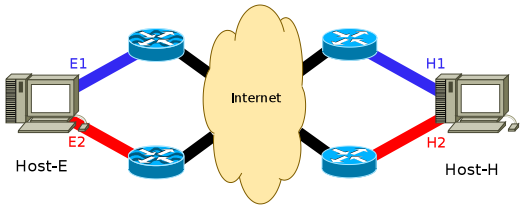


Figure 5. Internet testbed between Essen/Germany and Haikou/China

IV. Internet Testbed and Lessons Learned

A. Internet testbed description

We have set up a distributed testbed between Germany and China to test the behavior of the multipath protocols in a challenging real-life scenario. This setup was considered challenging because high delays and delay variations as well as high hop counts could be expected. Furthermore it was not predictable how asymmetric the paths would be. The topology used (see Figure 5) consists of two multi-homed hosts located in Essen/Germany (Host-E) and in Haikou/China (Host-H), respectively.

Host-E is connected via a high-speed fiber optic connection (E1) to the DFN. The second path (E2) uses an ADSL connection in Essen (Versatel; 800 Kbit/s upstream). Host-H in China is connected via two high-speed fiber optic connections, the first one (H1) connects to the China Education and Research Network (Cernet)[3]. The second one (H2) is connected to the China United Telecommunications Corporation Hainan Province network (Unicom). We denote the path between Essen and Haikou using DFN in Germany and Unicom in China as $P_{DFN-Unicom}$. The other paths are named according to the same pattern as $P_{DFN-Cernet}$, $P_{Versatel-Unicom}$ and $P_{Versatel-Cernet}$. Both hosts have the configuration already described in the last section. The characteristics of the Internet connectivity beyond the first and last hops depends on the current routing state in the Internet and is unknown for the

[3]http://www.edu.cn/english

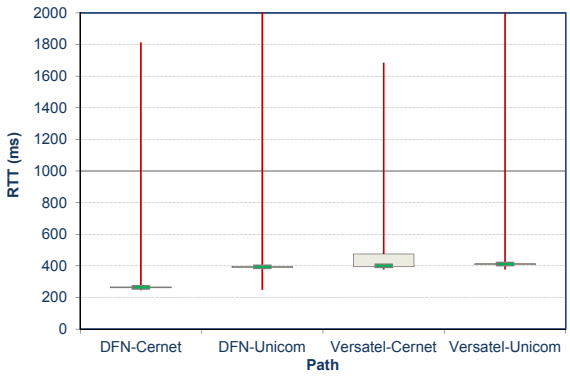


Figure 7. RTT Statistics for each path - Box Plot

transport endpoints. Therefore, we first gathered information about the topology and performed delay and throughput measurements between the 2012-12-07 and 2013-01-01.

B. Path topology through the Internet

To get a deeper understanding of the paths resulting from the Internet topology we used the tool *Traceroute*. We started a trace every three minutes from Essen to Haikou between 2012-12-07 and 2013-01-01 resulting in a total of around 29,000 traces.

The average hop counts encountered on the four paths were 21 ($P_{DFN-Unicom}$), 18 ($P_{DFN-Cernet}$), 20 ($P_{Versatel-Unicom}$) and 23 ($P_{Versatel-Cernet}$), respectively. While the overall paths remained stable, we observed dynamic changes in some path segments regularly.

We used IP localization services[4] to locate the geographical position of the routers and plotted the most representative paths in Figure 6. The illustration shows that depending on the Chinese ISP targeted in the destination address, the traffic was routed from Germany directly to the east or to the west via a transatlantic and a transpacific line. For the path $P_{DFN-Cernet}$ we could confirm this from the relevant literature: The DFN is directly connected to the Gigabit European Academic Network backbone (GÉANT[5]) which is linked to the Chinese research network via the ORIENTplus link[6] directly connecting London and Beijing via Siberia. For the other routes shown we had to rely on the IP localization services.

We can state that several routes exist between the two endpoints and that they are at least partly disjoint. However, compared to the simple scenario used in most of the existing multipath evaluations (see [7], [9], [13]–[15]) we have additional paths which will have a significant impact as we will show. All paths either originating or terminating at the same interface obviously share the first or last hop, respectively. If these are low capacity links (e.g. the DSL link in Essen), they will most likely constitute the shared bottleneck for these paths.

C. Delay/RTT evaluation

Delays were estimated by sending ICMP packets between the four possible address combinations and measuring the Round Trip Times (RTT). A statistical evaluation of the results is shown as a *Box Plot* in Figure 7.

For this, all RTT values observed have been sorted. The vertical red lines denote the complete range of observations with the endpoints indicating minimum and maximum. The green colored dash denotes the median dividing this sorted list in the middle. The black box (best visible for $P_{Versatel-Cernet}$) holds 50% of the observed values with the upper and lower 25% (Upper and Lower Quantile, UQ, LQ) lying outside.

We can see that the $P_{DFN-Cernet}$ has the lowest RTT median of 263 ms, the three other paths are around 400 ms. The small difference between the LQ and UQ for $P_{DFN-Cernet}$, $P_{DFN-Unicom}$ and $P_{Versatel-Unicom}$ indicates that the RTT values are quite stable for these paths. This is different for $P_{Versatel-Cernet}$ where half of the measured values are between LQ=394 ms and UQ=475 ms.

Some delay values are obviously very high, but these values are rarely observed and can be considered as outliers. The highest RTT measured on $P_{DFN-Unicom}$, e.g. was 7282 ms. However, only 3 of 354576 values on this path were over 1000 ms. Therefore, we cropped the maximum value shown at 2000 ms to enlarge the relevant areas.

The conclusion is that 200 ms is a reasonable estimate for the end-to-end delay (half of RTT) and that this delay is fairly stable.

	DFN-Cernet	DFN-Unicom	Versatel-Cernet	Versatel-Unicom
Mean (Kbit/s)	208	255	496	772
Min (Kbit/s)	94	101	102	463
Max (Kbit/s)	295	416	800	800

Table I
THROUGHPUT STATISTICS FOR EACH PATH

D. Bandwidth evaluation

Bandwidth estimations have been performed by using a test application generating a saturated TCP flow between pairs of addresses and measuring the achievable throughput. The results are summarized in Table I.

The first conclusion is that the average achievable throughput is well below 1 Mbit/s for all paths. While the Versatel DSL link could be loaded fully in some cases (800 Kbit/s), the high speed DFN connection performed significantly worse, both with respect to mean and maximum. It was not possible to identify the reasons for this unexpected behavior.

In summary, the tests on the global setup confirmed that the simplistic multipath scenario with two disjoint paths does not reflect the actual connectivity in the Internet. It also showed that even considering the significant delays, both multipath protocols can operate in the region where they should provide the expected benefits. The achievable throughput is well below the 2.5 Mbit/s threshold (see III-C) where the CPUs start to have a significant influence on the measurements.

V. MULTIPATH PROTOCOL EXTENSIONS IN THE INTERNET

After confirming that the Internet setup provides an environment where the multipath protocols can – in principle – operate as expected we started the experiments. However, first measurements have shown a significant difference between MPTCP and CMT-SCTP. As both congestion control and CPU load could be ruled out as reasons, the different path

[4]http://www.maxmind.com, http://www.iplocation.net
[5]http://www.geant.net
[6]http://www.orientplus.eu

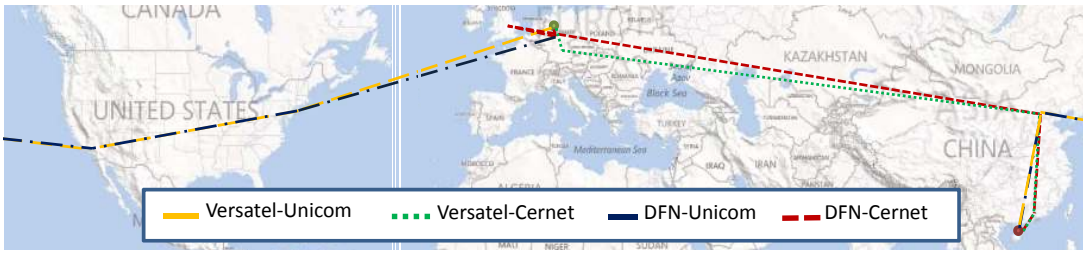


Figure 6. Paths between Essen/Germany and Haikou/China

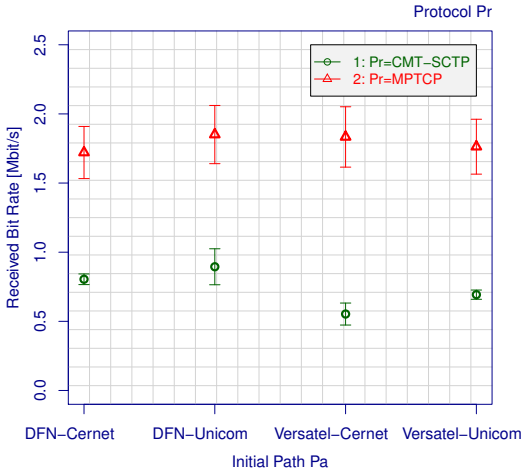


Figure 8. Inter-Continental Scenario

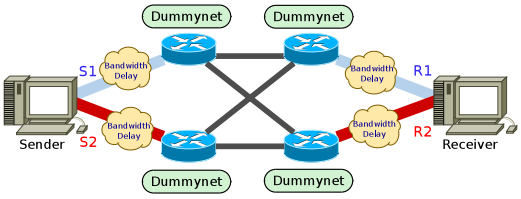


Figure 9. Extended Local testbed in Essen/Germany

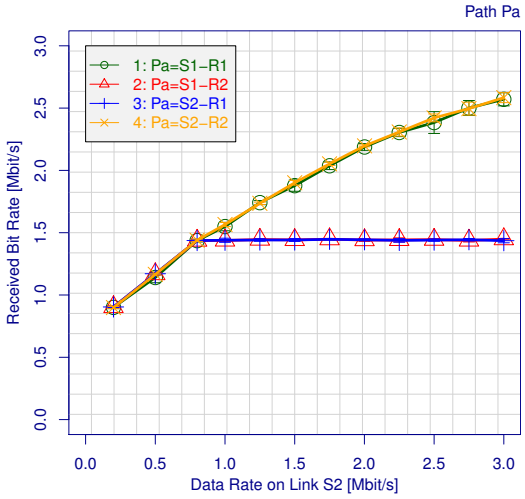


Figure 10. Local testbed - Different Primary Paths for CMT-SCTP

management strategies (see section II-B) could be identified as predominant factor.

While MPTCP builds a mesh using all available address pairs, CMT-SCTP identifies a path by a pair of source and destination addresses, creates only one additional path per additional source address and, even more important, routes to the next hop based on the source address (source address selection). This makes no difference in the simple scenarios with only two disjoint paths or even in fully meshed up scenarios (if all access points provide the same capacity). The choice of the strategy has a significant impact in more complex and asymmetric Internet topologies. Furthermore, for SCTP the selection of the source/destination address pair for the initial handshake determines the first "primary" path – and consequently also the options left for the additional ones. If unfavorable combinations are chosen, this may have a significant impact on the achievable throughput.

To verify and quantify the impact of the path management strategies, we performed sets of measurements where we used each of the four possible address pairs to initiate the MPTCP and CMT-SCTP connections. The measurements were performed using the parameters described in section IV-D and repeated 50 times over a period of several weeks to get representative results independent from the current state of the Internet.

Figure 8 shows that the throughput of the MPTCP connection significantly exceeds the maximum throughput measured for singlepath connections (compare Table I) confirming the benefits of multipath transfer. Furthermore, it is significantly higher in all cases than for CMT-SCTP where the throughput is less than for the best singlepath cases. This demonstrates that the use of the additional paths by MPTCP actually provides significant advantages in this scenario.

It can also be seen that the MPTCP throughput did not significantly depend on the address pair used to set up the connection initially as the variations are basically covered by the confidence intervals. For CMT-SCTP, however, the choice of the initial address pair had a significant influence on the achievable throughput. The singlepath throughput was better if the DSL interface was used in Essen (compare Table I), whereas for CMT-SCTP the performance was better if the connection was initially established via the DFN interface. This effect depends on a multitude of factors, both related to the protocol and the Internet setup, and requires additional research for a proper explanation.

In order to further investigate the path management issue, we extended furthermore our local testbed as shown in Figure 9, to show that this is a general effect in meshed network with asymmetric access points. In contrast to the initial disjoint path scenario, all four possible paths (P_{S1-R1}, P_{S1-R2},

P_{S2-R1} and P_{S2-R2}) can be used. We also configured the bandwidth restrictions on every link separately and according to the situation in the Internet testbed. The links $S1$ and $R1$ have been limited to 800 Kbit/s and link $R2$ has been set to 3 Mbit/s. The bandwidth of $S2$ has been varied between 200 Kbit/s and 3 Mbit/s. A delay of 200 ms has been set on the links $S1$ and $S2$ to match the measured end-to-end delay.

The results of this experiment are shown in Figure 10. As expected, if the initial path is set up via the low speed link at the sender and via the high speed link at the receiver – or vice versa – the throughput does not benefit from the faster link $R2$ and the initial choice of the address pairs is crucial.

As the access configuration of the remote side is typically not known, an intelligent choice would require complex measurements prior to connection setup which is not desirable. An obvious remedy would be to adopt the mesh-type path management of MPTCP. However, the decision for the (CMT-)SCTP strategy was a conscious one. The goal was to keep CMT-SCTP scalable, e.g. for scenarios with more than two addresses, which are common for dual stack IPv4/IPv6 configurations. Another goal was to avoid keeping too much Network layer information in the Transport layer – as it is subject to dynamic changes and consequently requires management effort.

VI. Conclusions and future work

In this paper we presented a comparison of the currently discussed multipath extensions for SCTP and TCP which is based on measurements with real implementations. We demonstrated that even if these approaches perform – in principle – well in the simple scenarios with two disjoint paths which have been used in previous evaluations, there still exist several open issues.

One issue is that in real applications, the maintenance of, e.g., gap lists for the Selective Acknowledgments, can have a significant impact on the performance in realistic wide-area scenarios. This aspect has to be analyzed further and optimized solutions have to be found.

Furthermore, an analysis of the multipath testbed we set up between Germany and China has shown that the simple scenario with two disjoint paths does not reflect the actual situation in the real Internet sufficiently. In fact, the additional paths existing there between endpoints reveal a significant performance impact of the different path management strategies of the two protocol variants. As a first consequence, we suggest that future evaluation scenarios for multipath protocols should definitively include the additional cross-paths introduced here. Our experiments show that in the scenario used, the MPTCP strategy to create a full mesh of paths among the available interfaces performs significantly better than the more restrictive approach of CMT-SCTP. However, it can be expected that the MPTCP strategy may face scalability problems in more complex application scenarios, e.g. with more than two addresses per endpoint. Therefore, a further systematic evaluation of the path management issue is definitely required prior to a comprehensive deployment of these protocols in the Internet.

References

[1] F. Kelly and T. Voice, "Stability of End-to-End Algorithms for Joint Routing and Rate Control," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, pp. 5–12, Apr. 2005, ISSN 0146-4833.

[2] L. Magalhaes and R. Kravets, "Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts," in *Proceedings of the 9th IEEE International Conference on Network Protocols (ICNP)*, Riverside, California/U.S.A., Nov. 2001, pp. 165–171, ISBN 0-7695-1429-4.

[3] H.-Y. Hsieh and R. Sivakumar, "pTCP: An End-to-End Transport Layer Protocol for Striped Connections," in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP)*, Paris/France, Nov. 2002, pp. 24–33, ISBN 0-7695-1856-7.

[4] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths," in *In Proceedings of the USENIX Annual Technical Conference*, Boston, Massachusetts/U.S.A., June 2004, pp. 99–112.

[5] 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.

[6] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-End Paths," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, Oct. 2006, ISSN 1063-6692.

[7] 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.

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

[9] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, "On the Fairness of Transport Protocols in a Multi-Path Environment," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Ottawa, Ontario/Canada, June 2012, pp. 2666–2672.

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

[11] 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, Network Working Group, Internet Draft Version 06, Mar. 2013, draft-tuexen-tsvwg-sctp-multipath-06.txt, work in progress.

[12] 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, June 2011, pp. 397–404, ISBN 978-953-184-152-8.

[13] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving Datacenter Performance and Robustness with Multipath TCP," in *Proceedings of the ACM SIGCOMM*, Toronto/-Canada, Aug. 2011.

[14] C. Pluntke, L. Eggert, and N. Kiukkonen, "Saving Mobile Device Energy with Multipath TCP," in *Proceedings of the 6th ACM International Workshop on MobiArch*, Bethesda, Maryland/U.S.A., June 2011, pp. 1–6, ISBN 978-1-4503-0740-6.

[15] S. Barré, C. Paasch, and O. Bonaventure, "MultiPath TCP: From Theory to Practice," in *Proceedings of the 10th International IFIP Networking Conference*, Valencia/Spain, May 2011, pp. 444–457, ISBN 978-3-642-20756-3.

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

[17] R. R. Stewart, "Stream Control Transmission Protocol," IETF, Standards Track RFC 4960, Sept. 2007, ISSN 2070-1721.

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

[19] 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/Croatia, Sept. 2011, pp. 1–6, ISBN 978-953-290-027-9.

[20] Institute of Information and Electronics Communication Technologies and Applied Mathematics (ICTEAM), "MultiPath TCP – Linux Kernel Implementation," 2013.

[21] M. Carbone and L. Rizzo, "Dummynet Revisited," Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa/Italy, Tech. Rep., May 2009.

[22] S. Floyd, "RED: Discussions of Setting Parameters," Nov. 1997.

[23] 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.