

# Multipath TCP : An annotated bibliography

Olivier Bonaventure

ICTEAM, UCL, Louvain-la-Neuve, Belgium

April 28, 2015

## 1 Introduction

Multipath TCP[31] is a recent TCP extension being developed within the MPTCP workgroup of the Internet Engineering Task Force. Multipath TCP enables a TCP connection to exchange data over different interfaces. This extension has recently received a growing interest from both researchers who publish a growing number of articles on the topic and the vendors since Apple has decided to use Multipath TCP on its smartphones and tablets to support the Siri voice recognition application.

This document assumes that the reader is familiar with Multipath TCP. Tutorials on this TCP extension may be found in [61, 70, 15].

This document gathers an as complete as possible bibliography of the IETF documents and scientific publications related to Multipath TCP. It is maintained at <http://github.com/obonaventure/mptcp-bib>. Comments, suggestions and contributions are more than welcome.

The document is organised as follows. We first discuss in section 2 the publications that propose protocol extensions. Section 3 compares the different congestion control proposals. Section 4 analyses the different use cases for Multipath TCP and summarises the existing work. Section 5 lists that various software tools that can be used by Multipath TCP users and researchers.

## 2 Multipath TCP protocol

The first phase of the work in the IETF MPTCP working group has been focussed on the production of several documents. The architectural guidelines are specified in [30]. This document served as a reference for the work in the IETF working group. The main design decision was that Multipath TCP assumes that the communicating hosts have different addresses and that these addresses are used to identify the flows. The main requirements listed in this document were :

- *Improve Throughput*: Multipath TCP MUST support the concurrent use of multiple paths. To meet the minimum performance incentives for deployment, a Multipath TCP connection over multiple paths SHOULD achieve no worse throughput than a single TCP connection over the best constituent path.
- *Improve Resilience*: Multipath TCP MUST support the use of multiple paths interchangeably for resilience purposes, by permitting segments to be sent and re-sent on any available path.

In addition, [30] lists several compatibility goals that include compatibility with existing applications that use the socket API and compatibility with the network including the middleboxes that it could contain.

The security requirements were discussed in more details in a separate document [4]. This document defines the flooding attacks and the hijacking attacks that were considered for the initial design of Multipath TCP.

The detailed specification for the Multipath TCP protocol extension may be found in [31]. This specification is being updated to include in the revised specification the lessons learned from the utilisation of Multipath TCP in the global Internet [14]. Several of the design choices for the protocol and its implementation in the Linux kernel are discussed in [72]. The revised specification [32] includes several improvements such as a modified `ADD_ADDR` option that includes a HMAC and an option for the RST segments [13].

A companion document discusses the security risks and possible attacks with Multipath TCP and how the identified attacks could be mitigated [5]. The recent modification of the `ADD_ADDR` option is a result of this work.

The standard congestion control scheme chosen by the MPTCP working group for Multipath TCP is described in [69]. This congestion control scheme was first proposed and evaluated in [84].

One of the initial design choices of Multipath TCP was to be as compatible as possible with the existing applications that interact with TCP through the socket interface. Still, [73] describes some basic extensions to the socket API to enable applications to notably disable Multipath TCP.

## 2.1 Protocol extensions

Several extensions to Multipath TCP have been proposed within the IETF MPTCP working group.

TCP Fast Open is one of the last TCP extensions that has been added to TCP. Barre et al. discuss in [7] how to support TFO within Multipath TCP and report their experience in implementing TFO in the Multipath TCP implementation in the Linux kernel.

To cope with some forms of denial of service attacks, various TCP servers rely on syncookies. This approach enables them to respond to `SYN` segments without maintaining any state on the server. State is only created upon reception of a valid third `ACK` segment. With Multipath TCP, it is more difficult to enable servers to operate without maintaining state given the keys exchanged in the `MP_CAPABLE` option. [57] analyses this problem and proposes a new `MP_CAPABLE_EXT` option to cope with this problem.

Single-path TCP uses the timestamp option defined in RFC1323 to estimate round-trip-times and more importantly to protect TCP against problems that could occur when a connection sends data quickly and sequence numbers wrap. For this usage, RFC7323, requires all TCP segments to carry a timestamp option. With Multipath TCP, the second utilisation of the timestamp option is void since Multipath TCP uses 64 bits sequence numbers that protect against problems when the 32 bits sequence numbers wrap. Requiring a timestamp option in all segments is a waste of space in the limited TCP header. [12] proposes a different type of timestamp option for Multipath TCP.

The extensibility of TCP is limited by the difficulty of adding new options in the TCP header that cannot be longer than 64 bytes. [60] proposes a different approach that uses one flag in the `DSS` option to indicate whether such an option maps regular data or control information.

Several researchers have analysed the security of Multipath TCP and proposed different solutions to improve its security. Diez et al. propose in [26] to use hash chains to secure the establishment of subflows. Instead of exchanging keys to secure only the establishment of subflows, [59] assumes that an application layer protocol like SSL, TLS or ssh will negotiate keys to authenticate and encrypt the data. In this case, a better approach is to derive the keys used by Multipath TCP to authenticate the new subflows for the keys exchanged by the application level protocol [59]. [11] goes one step further by proposing MPTLS, a close integration of TLS and Multipath TCP.

In some environments, attacks are not a concern. In these environments, generating random keys for

each Multipath TCP connection can limit performance. [58] proposes a small modification to Multipath TCP that does not use random keys to authenticate the establishment of subflows.

## 2.2 Interactions with other protocols

In several use cases, such as datacenters, the selection of the best paths for the subflows that compose a Multipath TCP connection is an important decision. The current IETF RFCs do not mandate any path selection mechanisms. Proposed techniques include random selection with ECMP [68], modifying the hash function used by ECMP [25] is used, or using other fields like the TTL to influence the path as proposed in [42]. Another approach is to develop a signalling protocol that enables the hosts to query the network for the paths to be used to reach a specific destination. Krupakaran et al. propose in [47] a layer-2 protocol called Traceflow to exchange this kind of information.

## 3 Multipath TCP congestion control

Various researchers have worked on designing congestion control schemes for Multipath TCP. One of the key issues compared to regular congestion control schemes is that due to the utilisation of several not necessarily disjoint paths a Multipath TCP congestion control scheme must ensure that it does not harm regular (single path) TCP traffic.

Wischnik et al. describe in [83] the resource pooling principle that has been the motivation for the design of implementable congestion control schemes for Multipath TCP. Since then, several congestion control schemes have been explored and proposed.

Wischnik et al. describe in [84] the Linked Increase Algorithm that has also been selected by the IETF MPTCP working group [69]. This congestion control scheme is a variant of the standard Reno congestion control algorithm used by single path TCP. This is the default congestion control scheme in the Multipath TCP implementation in the Linux kernel. Another variant of Reno has been proposed [78].

Cao et al. propose in [17] a delay-based congestion control scheme for Multipath TCP. This congestion control scheme relies on delay measurements to estimate congestion and adapt the congestion windows of the different subflows. It has been included in the Linux Multipath TCP implementation.

Khalili et al. describe in [45, 44] several conditions where the LIA congestion control scheme does not operate perfectly and propose the OLIA congestion control scheme [43]. This congestion control scheme is supported by the Linux Multipath TCP implementation.

Peng et al. describe in [66] another improvement to the LIA congestion control scheme called BALIA [79]. This congestion control scheme is also supported by the Linux Multipath TCP implementation.

Delay-based congestion control has been proposed in regular TCP a few decades ago. Cao et al. propose in [17] a delay-based congestion control scheme that is adapted to Multipath TCP. This congestion control has been included in the Multipath TCP implementation in the Linux kernel since version 0.89 [85].

## 4 Multipath TCP use cases

This section describes several of the use cases where Multipath TCP has already been evaluated or used.

## 4.1 Datacenters

Although Multipath TCP was initially designed with smartphones and mobile devices in mind it found a very interesting use case in datacenters. Today's datacenters include multiple paths between any pair of communicating hosts. This redundancy is included for two main reasons. First, it enables load balancing and datacenters often rely on Equal Cost Multipath. Second, having redundancy is important to deal with switches and link failures that are frequent events [88].

Raiciu et al. demonstrate in [68] that Multipath TCP can enable higher utilisation in datacenters. This work considers several datacenter topologies and uses both simulations and real measurements with the Multipath TCP implementation in the Linux kernel running on Amazon EC2 virtual machines to demonstrate the benefits of Multipath TCP. It is interesting to note that for this use case the communicating hosts have a single address in contrast with the requirements listed in [30]. The different paths are created by creating several subflows between the client and the server using different source ports on the client host. The hash function used to perform ECMP on the network nodes forwards the different subflows over different paths and the Multipath TCP congestion control scheme adjusts the repartition of the load among these paths in reaction to congestion events.

Although [68] assumed ECMP, this is not strictly required and variations of this scheme are possible. Detal et al. propose in [25] to replace hash function in load balancing nodes by block ciphers and demonstrates that this technique enables Multipath TCP-enabled hosts to select paths inside the datacenter. Zhou et al. propose in [88] a load balancing scheme that distributes the load according to different weights. Their simulations show that WCMP complements nicely Multipath TCP.

Bredel et al. modify in [16] an Openflow controller and use it with Multipath TCP. Their measurements show performance improvements over disjoint paths.

Tariq and Bassiouni use simulations in [76] to evaluate the benefits of combining Multipath TCP with optical burst switching in datacenters.

## 4.2 Mobile and wireless

The mobile and wireless networks were one of the initial motivations for Multipath TCP. Several researchers have studied Multipath TCP in such environments.

Raiciu et al. [71] discuss how Multipath TCP can be used to support mobile devices and provides different measurement results with a Multipath TCP implementation running on a laptop. They also analyse the energy consumption with Multipath TPC.

Pluntke et al. [67] analyse whether Multipath TCP could reduce energy consumption by using several interfaces simultaneously. They also propose a scheduling algorithm that takes energy into account. Peng et al. go one step further in [65] by proposing algorithm that try to optimise both throughput and energy consumption.

In [49], Lim et al. propose to take energy consumption into account when managing subflows with Multipath TCP. They first develop a model of the energy consumption of TCP and MPTCP on a smartphone based on a collected trace. To reduce energy consumption, that propose a small change to the Multipath TCP implementation that delays the establishment of subflows over the LTE interface and manipulates the congestion window and the `MP_PRIO` flag. When enabling a subflow over an LTE interface, they also reset the round-trip-time measurement to zero to ensure that this subflow will be scheduled. This is not a clean solution to solve this problem.

Paasch et al. propose in [62] several modes for Multipath TCP on multihomed wireless devices and extends the Multipath TCP implementation in the Linux kernel to support these modes of operation. They also analyse the reaction of Multipath TCP to handover events.

Chen et al. report in [19] various measurements conducted between a laptop and a server running Multipath TCP over different wireless networks (cellular and WiFi). They compare the performance of regular TCP, Multipath TCP over two paths and Multipath TCP over four paths with both short connections and long connections. Their general conclusion is that *MPTCP provides a robust data transport and reduces the variability in download latencies.*

Nguyen et al. analyse in [52] the benefits of combining Multipath TCP with WiFi virtualisation. They use virtualisation to enable a laptop with a single WiFi interface to connect to two access points and let Multipath TCP share the load over the two networks. Measurements are performed in lab and home environments and consider bandwidth aggregation and handover.

Boccassi et al. propose Binder in [10]. This system leverages Multipath TCP to aggregate multiple Internet gateways in community networks. The Binder path manager that is included in the Multipath TCP implementation in the Linux kernel is a result of this work.

Zhou et al. explore in [87] the possibility of allowing users to cooperate. The considered scenario is an LTE network and smartphones that serve as relays to enable other users to reach the LTE network.

Nong et al. explore in [55] how Multipath TCP performs in an 802.11s mesh network and propose to use this technique to aggregate multiple wireless gateways.

Williams et al. explore in [81] the performance of Multipath TCP in moving vehicles for interactions between the vehicles and the infrastructure. Their measurements and simulations indicate positive results when paths of similar characteristics. When paths are heterogeneous, there are situations where Multipath TCP does not perform as good as regular TCP, especially for short flows that transfer less than 1 MBytes of data and do not exit slow-start. When using connections with the road infrastructure, their measurements show that it is possible to boost the throughput and perform handover at 40km/h.

### 4.3 Middleboxes

Various types of middleboxes, ranging from simple Network Address Translation or firewalls to Deep Packet Inspection, Normalizers or transparent proxies have been deployed by enterprise and network operators. Honda et al. performed in [41] a detailed study of the interference caused by such middleboxes. This study had a profound impact on the final design of Multipath TCP and the various mechanisms that it includes to detect and mitigate interference from such middleboxes.

Detal et al. propose [tracebox](#) in [23]. This middlebox detection tools leverages ICMP reports to detect middlebox interference. They use it to evaluate where TCP packets using the `MP_CAPABLE` options are blocked.

Hesmans et al. analyse in [39] the various interactions between Multipath TCP and different types of middleboxes. Their measurements with the Multipath TCP implementation in the Linux kernel show that it can correctly cope with a wide range of middleboxes. Since then, other successful interactions with stranger middleboxes have been reported [14].

Nicutar et al. take a different approach in [53] since they propose to leverage Multipath TCP to design new middleboxes that could cooperate with the endhosts.

### 4.4 Multipath TCP Proxies

Multipath TCP is an end-to-end protocol that needs to be deployed on both clients and servers to achieve all its benefits. Measurements with past TCP extensions showed that new TCP extensions are deployed rather slowly and that servers are upgraded after clients. Several researchers have proposed and in some cases implemented different types of proxies to convert Multipath TCP into regular TCP or the opposite.

Hampel and Klein propose in [35] two types of Multipath TCP middleboxes. A proxy can be used to convert a Multipath TCP connection into a regular TCP connection. An anchor is a middlebox that aids mobility. A prototype user space implementation on top of netfilter has been released<sup>1</sup>. Additional details and some performance measurements are provide in [36].

Ayar et al. propose in [2] a Splitter/Combiner architecture that enables TCP connections to use different paths through the network. The proposed architecture is discussed in more details, implemented with netfilter and tested by simulations in [74, 3]

Detal et al. propose in [24] Mimbox (Multipath in the middle(box)), an efficient in-kernel proxy that converts Multipath TCP connections into TCP connections and the opposite. Performance evaluation on 10 Gbps links shows that the proposed solution operates are speeds that are close to line rate and similar to the performance of simple packet forwarding.

Multipath TCP is not the only transport protocol that allows to use multiple paths simultaneously. SCTP-CMT is a variant of the SCTP protocol that is able to use several paths at the same time. Tachibana et al. propose in [75] a proxy that runs on Android smartphones and converts TCP connections into SCTP-CMT connections to reach a server that converts them back to regular TCP to reach the final destination. In practice, [75] uses a userspace implementation of SCTP on the smartphone and tunnels the SCTP packets inside UDP to pass through middleboxes that do not understand SCTP. Each TCP connection is mapped onto one SCTP stream, which minimises the number of SCTP connections that need to be established between the smartphone and the proxy.

Deng et al. discuss in [50] the requirements for on-path and off-path Multipath TCP proxies that could be deployed mobile and fixed in ISP networks.

Wei et al. [80] discuss on-path and off-path Multipath TCP proxies and propose a new flag in the `MP_CAPABLE` option to indicate the utilisation of a proxy.

## 4.5 Simulation studies

To be provided

## 4.6 Measurement studies

The first measurements with the Multipath TCP implementation in the Linux kernel were reported in [9]. Since then, the implementation has significantly changed. Raiciu et al. report in [72] the main design and implementation choices behind Multipath TCP and provide a performance evaluation of several important algorithms including the cost of computing the DSN checksum on 10 Gbps interfaces, the connection establishment latency, the impact of limited buffers. The latter analysis led to the development of the penalisation and opportunistic retransmission algorithms. These two algorithms have been further improved in [64].

Arzani et al. analyse in [1] the impact of scheduling policies and path management on the performance of Multipath TCP.

Yamauchi and Ito evaluate in [86] whether the users observe a benefit when Multipath TCP is used in lossy environments. The measurements with a limited set of users and web applications show some benefit with Multipath TCP in the chosen environment.

Ferlin et al. analyse in [29], the impact of bufferbloat in cellular networks on the performance of Multipath TCP and propose a sender-based solution to mitigate the problem. The proposed solution is tested in

---

<sup>1</sup>See [https://open-innovation.alcatel-lucent.com/docman/?group\\_id=73&selected\\_doc\\_group\\_id=313](https://open-innovation.alcatel-lucent.com/docman/?group_id=73&selected_doc_group_id=313).

the Linux kernel implementation of Multipath TCP. Chen and Towsley have also modelled the impact of the bufferbloat on Multipath TCP in [20].

Most of the articles on Multipath TCP have used test applications like `iperf` or long web download to exchange large amounts of data. Hesmans et al. provide in [40] another viewpoint on the performance of Multipath TCP by analysing a one-week long packet trace collected on <http://www.multipath-tcp.org>.

## 4.7 Testbeds

Nemeth et al. propose in [51] an overlay network on top of the Planetlab testbed to enable researchers to perform experiments with Multipath TCP in the wide area.

The Nornet testbed [33, 48] is an international testbed that gathers multihomed hosts. It is composed of two parts. The Nornet core nodes [33] are servers connected to several ISP networks. They form the backbone of the Nornet testbed. There are Nornet core nodes in several continents. The Nornet edge nodes [48] are embedded devices that are attached to different wireless (WiFi and cellular) networks. The Nornet edge nodes are currently located in Norway. Both types of nodes have been used to perform a wide range of measurements with Multipath TCP [27].

# 5 Multipath TCP Software

This section documents the different Multipath TCP implementations [28] and several interesting applications that have been designed with Multipath TCP in mind.

## 5.1 Multipath TCP implementations

The first implementation of Multipath TCP written by Costin Raiciu inside a userspace port of the Linux TCP stack<sup>2</sup>. This implementation has been used to perform some initial validations of the protocol and the LIA congestion control scheme [84], but it was not useable by real applications.

The first in-kernel implementation of Multipath TCP was written by Sebastien Barre in the Linux kernel with the help of several other researchers. It is described in more details in [6]. Some information about this early implementation may be found in [8] but this document has not been updated. This initial implementation supported the LIA congestion control scheme but did not support TSO/GRO nor the mechanisms that are required to deal with middlebox interference.

Christoph Paasch with the help of Gregory Detal continued the development of the Multipath TCP patch in the Linux kernel. They ported the implementation over the 3.x kernel releases and added support for TSO/GRO and middlebox traversal techniques. This implementation is described in [56]. Several parts of this implementation have been described in more details.

In [63], Paasch et al. propose and implement more flexible packet schedulers for Multipath TCP. In the reference implementation of Multipath TCP in the Linux kernel, the default scheduler selects the subflow having the lowest round-trip-time. In addition to this scheduler, [63] proposes a round-robin scheduler that takes to congestion windows of the available subflows into account.

The next step for the Multipath TCP in the Linux kernel is to be accepted in the mainstream Linux kernel tree. Gucea and Purdila describe in [34] the current status of this implementation and the changes that will be required to be accepted upstream.

---

<sup>2</sup>Available from <http://nrg.cs.ucl.ac.uk/mptcp/implementation.html>.

The Multipath TCP implementation in the Linux kernel is open-source and available from <https://github.com/multipath-tcp/mptcp>.

The most widely deployed implementation of Multipath TCP is developed by Apple and is included in both iOS and MacOS. This implementation is open-source. The latest public version is available from <http://www.opensource.apple.com/source/xnu/xnu-2782.1.97/bsd/netinet/>. Unfortunately, besides [28], there are no public details on the performance of this implementation and as of this writing, there is no public API that enables an application to use Multipath TCP to send data on an Apple device.

Citrix has also developed an implementation of Multipath TCP for the Netscaler load-balancer. The only published details of this implementation are reported in [28].

Nigel Williams and his colleagues develop an implementation of Multipath TCP for FreeBSD 11. This implementation is described in details in [82]. It is available from <http://caia.swin.edu.au/urp/newtcp/mptcp/>

## 5.2 Test tools

Several software tools have been designed to aid Multipath TCP implementors and researchers.

Coene proposed in [22] to validate a Multipath TCP implementation. These tests have not been maintained, but Schils and Creciu have extended `packetdrill` [18] to support Multipath TCP. This extension is available from [https://github.com/aschils/packetdrill\\_mptcp](https://github.com/aschils/packetdrill_mptcp). It can be used to develop unit tests to validate some specific behaviours of a Multipath TCP stack.

`tracebox` is a traceroute-like application that was initially designed by Gregory Deal [23]. `tracebox` includes a small language that enables researchers to write specific tests to expose middleboxes or interact with a Multipath TCP stack. `tracebox` is available from <http://www.tracebox.org>. Another way to inject Multipath TCP packets is to use the `scapy` extension<sup>3</sup> developed by Nicolas Maitre that has been recently extended and is available from <https://github.com/Neohapsis/mptcp-abuse>.

TCP researchers often rely on `tcptrace`<sup>4</sup> to graphically analyse packet traces from TCP connections. Hesmans designed and implemented a similar tool [38] that allows to graphically analyse Multipath TCP connections.

## 5.3 Simulators

Several simulators have been used to evaluate the performance of Multipath TCP and design congestion control schemes or protocol extensions. `htsim`<sup>5</sup> is the original simulator that was used notably for [84, 68].

Several researchers have used `ns-2` to evaluate the performance of Multipath TCP starting from the initial model developed by Yoshifumi Nishida [54]. Two Multipath TCP models have been developed for `ns-3`. The first one was proposed by Chihani and Collange in [21].

Another `ns-3` model was proposed recently [46]. With `ns-3`, it is also possible to use the complete Multipath TCP implementation in the Linux kernel by leveraging the `ns-3-dce` extension [77].

Besides simulators, it is also possible to perform reproducible experiments with Multipath TCP by leveraging the namespaces that are supported by recent Linux kernels [37]. Several researchers have relied on this framework to evaluate modifications to the Multipath TCP implementation [63, 64].

---

<sup>3</sup>See <https://github.com/nimai/mptcp-scapy>.

<sup>4</sup><http://tcptrace.org>

<sup>5</sup>Available from <http://nrg.cs.ucl.ac.uk/mptcp/implementation.html>.

## Acknowledgements

This work was partially supported by the European Commission within the FP7 Trilogy2 project.

## References

- [1] ARZANI, B., GURNEY, A., CHENG, S., GUERIN, R., AND LOO, B. T. [Impact of path characteristics and scheduling policies on MPTCP performance](#). In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on* (May 2014), pp. 743–748.
- [2] AYAR, T., RATHKE, B., BUDZISZ, L., AND WOLISZ, A. [A Transparent Performance Enhancing Proxy Architecture To Enable TCP over Multiple Paths for Single-Homed Hosts](#). Tech. Rep. draft-ayar-transparent-sca-proxy-00, IETF, February 2012.
- [3] AYAR, T., RATHKE, B., BUDZISZ, L., AND WOLISZ, A. [Tcp over multiple paths revisited: Towards transparent proxy solutions](#). In *Communications (ICC), 2012 IEEE International Conference on* (June 2012), pp. 109–114.
- [4] BAGNULO, M. [Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses](#). RFC 6181, March 2011.
- [5] BAGNULO, M., PAASCH, C., GONT, F., BONAVENTURE, O., AND RAICIU, C. [Analysis of MPTCP residual threats and possible fixes](#). Internet-Draft draft-ietf-mptcp-attacks-02, IETF Secretariat, Feb. 2015. IESG Evaluation::Revised I-D Needed.
- [6] BARRE, S. [Implementation and assessment of Modern Host-based Multipath Solutions](#). PhD thesis, UCLouvain, 2011.
- [7] BARRE, S., DETAL, G., AND BONAVENTURE, O. [TFO support for Multipath TCP](#). Internet-Draft draft-barre-mptcp-tfo-00, IETF Secretariat, July 2014.
- [8] BARRE, S., AND PAASCH, C. [MultiPath TCP - Guidelines for implementers](#). Internet-Draft draft-barre-mptcp-impl-00, IETF Secretariat, Mar. 2011.
- [9] BARRE, S., PAASCH, C., AND BONAVENTURE, O. [MultiPath TCP: From theory to practice](#). In *NETWORKING 2011*, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds., vol. 6640 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 444–457.
- [10] BOCCASSI, L., FAYED, M. M., AND MARINA, M. K. [Binder: A system to aggregate multiple internet gateways in community networks](#). In *Proceedings of the 2013 ACM MobiCom Workshop on Lowest Cost Denominator Networking for Universal Access* (New York, NY, USA, 2013), LCDNet '13, ACM, pp. 3–8.
- [11] BONAVENTURE, O. [MPTLS : Making TLS and Multipath TCP stronger together](#). Internet-Draft draft-bonaventure-mptcp-tls-00, IETF Secretariat, Oct. 2014. I-D Exists.
- [12] BONAVENTURE, O. [Multipath TCP timestamp option](#). Internet-Draft draft-bonaventure-mptcp-timestamp-00, IETF Secretariat, Oct. 2014. I-D Exists.

- [13] BONAVENTURE, O., PAASCH, C., AND DETAL, G. [Processing of RST segments by Multipath TCP](#). Internet-Draft draft-bonaventure-mptcp-rst-00, IETF Secretariat, July 2014.
- [14] BONAVENTURE, O., PAASCH, C., AND DETAL, G. [Experience with Multipath TCP](#). Internet-Draft draft-ietf-mptcp-experience-01, IETF Secretariat, Mar. 2015. I-D Exists.
- [15] BONAVENTURE, O., RAICIU, C., AND HANDLEY, M. [An overview of Multipath TCP](#). *USENIX login*; (October 2012).
- [16] BREDEL, M., BOZAKOV, Z., BARCZYK, A., AND NEWMAN, H. [Flow-based load balancing in Multipath Layer-2 networks using OpenFlow and multipath-TCP](#). In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking* (New York, NY, USA, 2014), HotSDN '14, ACM, pp. 213–214.
- [17] CAO, Y., XU, M., AND FU, X. [Delay-based congestion control for Multipath TCP](#). In *Network Protocols (ICNP), 2012 20th IEEE International Conference on* (Oct 2012), pp. 1–10.
- [18] CARDWELL, N., CHENG, Y., BRAKMO, L., MATHIS, M., RAGHAVAN, B., DUKKIPATI, N., CHU, H.-K. J., TERZIS, A., AND HERBERT, T. [packetdrill: Scriptable network stack testing, from sockets to packets](#). In *USENIX Annual Technical Conference* (2013), pp. 213–218.
- [19] CHEN, Y.-C., LIM, Y.-S., GIBBENS, R. J., NAHUM, E. M., KHALILI, R., AND TOWSLEY, D. [A measurement-based study of multipath tcp performance over wireless networks](#). In *Proceedings of the 2013 Conference on Internet Measurement Conference* (New York, NY, USA, 2013), IMC '13, ACM, pp. 455–468.
- [20] CHEN, Y.-C., AND TOWSLEY, D. [On bufferbloat and delay analysis of multipath TCP in wireless networks](#). In *Networking Conference, 2014 IFIP* (June 2014), pp. 1–9.
- [21] CHIHANI, B., AND COLLANGE, D. [A Multipath TCP model for ns-3 simulator](#). Tech. Rep. 1112.1932, arxiv, December 2011.
- [22] COENE, Y. [Conformance tests for Multipath TCP](#). Internet-Draft draft-coene-mptcp-conformance-00, IETF Secretariat, July 2013.
- [23] DETAL, G., HESMANS, B., BONAVENTURE, O., VANAUBEL, Y., AND DONNET, B. [Revealing middlebox interference with tracebox](#). In *Proceedings of the 2013 Conference on Internet Measurement Conference* (New York, NY, USA, 2013), IMC '13, ACM, pp. 1–8.
- [24] DETAL, G., PAASCH, C., AND BONAVENTURE, O. [Multipath in the middle\(box\)](#). In *HotMiddlebox '13* (2013), pp. 1–6.
- [25] DETAL, G., PAASCH, C., VAN DER LINDEN, S., MRINDOL, P., AVOINE, G., AND BONAVENTURE, O. [Revisiting flow-based load balancing: Stateless path selection in data center networks](#). *Computer Networks* 57, 5 (2013), 1204 – 1216.
- [26] DIEZ, J., BAGNULO, M., VALERA, F., AND VIDAL, I. [Security for Multipath TCP: a constructive approach](#). *International Journal of Internet Protocol Technology* 6, 3 (2011), 146–155.
- [27] DREIBHOLZ, T., AND ZHOU, X. [MPTCP Experiences in the NorNet Testbed](#). Internet Draft draft-dreibholz-mptcp-nornet-experience-00, IETF, Mar. 2015.

- [28] EARDLEY, P. [Survey of MPTCP Implementations](#). Internet-Draft draft-eardley-mptcp-implementations-survey-02, IETF Secretariat, July 2013.
- [29] FERLIN-OLIVEIRA, S., DREIBHOLZ, T., AND ALAY, O. [Tackling the challenge of bufferbloat in multi-path transport over heterogeneous wireless networks](#). In *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of* (May 2014), pp. 123–128.
- [30] FORD, A., RAICIU, C., HANDLEY, M., BARRE, S., AND IYENGAR, J. [Architectural Guidelines for Multipath TCP Development](#). RFC 6182, March 2011.
- [31] FORD, A., RAICIU, C., HANDLEY, M., AND BONAVENTURE, O. [TCP Extensions for Multipath Operation with Multiple Addresses](#). RFC 6824, January 2013.
- [32] FORD, A., RAICIU, C., HANDLEY, M., AND BONAVENTURE, O. [TCP Extensions for Multipath Operation with Multiple Addresses](#). Internet-Draft draft-ietf-mptcp-rfc6824bis-03, IETF Secretariat, Oct. 2014. I-D Exists.
- [33] GRAN, E. G., DREIBHOLZ, T., AND KVALBEIN, A. [NorNet Core – A Multi-Homed Research Testbed](#). *Computer Networks, Special Issue on Future Internet Testbeds 61* (Mar. 2014), 75–87. ISSN 1389-1286.
- [34] GUCEA, D.-C., AND PURDILA, O. [Shaping the Linux kernel MPTCP implementation towards upstream acceptance](#). Presented at Netdev01, March 2015.
- [35] HAMPEL, G., AND KLEIN, T. [MPTCP Proxies and Anchors](#). Internet-Draft draft-hampel-mptcp-proxies-anchors-00, IETF Secretariat, Feb. 2012.
- [36] HAMPEL, G., RANA, A., AND KLEIN, T. [Seamless tcp mobility using lightweight mptcp proxy](#). In *Proceedings of the 11th ACM International Symposium on Mobility Management and Wireless Access* (New York, NY, USA, 2013), MobiWac '13, ACM, pp. 139–146.
- [37] HANDIGOL, N., HELLER, B., JEYAKUMAR, V., LANTZ, B., AND MCKEOWN, N. [Reproducible network experiments using container-based emulation](#). In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies* (New York, NY, USA, 2012), CoNEXT '12, ACM, pp. 253–264.
- [38] HESMANS, B., AND BONAVENTURE, O. [Tracing Multipath TCP connections](#). *SIGCOMM Comput. Commun. Rev.* 44, 4 (Aug. 2014), 361–362.
- [39] HESMANS, B., DUCHENE, F., PAASCH, C., DETAL, G., AND BONAVENTURE, O. [Are TCP extensions middlebox-proof?](#) In *Proceedings of the 2013 Workshop on Hot Topics in Middleboxes and Network Function Virtualization* (New York, NY, USA, 2013), HotMiddlebox '13, ACM, pp. 37–42.
- [40] HESMANS, B., TRAN-VIET, H., SADRE, R., AND BONAVENTURE, O. [A first look at real multipath tcp traffic](#). In *Traffic Monitoring and Analysis*, M. Steiner, P. Barlet-Ros, and O. Bonaventure, Eds., vol. 9053 of *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 233–246.
- [41] HONDA, M., NISHIDA, Y., RAICIU, C., GREENHALGH, A., HANDLEY, M., AND TOKUDA, H. [Is it still possible to extend TCP?](#) In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference* (New York, NY, USA, 2011), IMC '11, ACM, pp. 181–194.

- [42] KABBANI, A., VAMANAN, B., HASAN, J., AND DUCHENE, F. [Flowbender: Flow-level adaptive routing for improved latency and throughput in datacenter networks](#). In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies* (New York, NY, USA, 2014), CoNEXT '14, ACM, pp. 149–160.
- [43] KHALILI, R., GAST, N., POPOVIC, M., AND BOUDEC, J.-Y. L. [Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP](#). Internet-Draft draft-khalili-mptcp-congestion-control-05, IETF Secretariat, July 2014.
- [44] KHALILI, R., GAST, N., POPOVIC, M., AND LE BOUDEC, J.-Y. [MPTCP is not pareto-optimal: Performance issues and a possible solution](#). *Networking, IEEE/ACM Transactions on* 21, 5 (Oct 2013), 1651–1665.
- [45] KHALILI, R., GAST, N., POPOVIC, M., UPADHYAY, U., AND LE BOUDEC, J.-Y. [MPTCP is not pareto-optimal: Performance issues and a possible solution](#). In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies* (New York, NY, USA, 2012), CoNEXT '12, ACM, pp. 1–12.
- [46] KHEIRKHAH, M., PARISIS, G., AND WAKEMAN, I. [MultiPath-TCP in network simulator 3](#). Poster presented at NS-3 Workshop, 2014.
- [47] KRUPAKARAN, K., SRIDHARAN, A. P., AND VENKATESAN, S. M. [Optimized Multipath TCP sub-flows using Traceflow](#). Tech. Rep. draft-aravind-mptcp-optimized-subflows-00, IETF, April 2015.
- [48] KVALBEIN, A., BALTRŪNAS, D., EVENSEN, K. R., XIANG, J., ELMOKASHFI, A. M., AND FERLIN, S. [The NorNet Edge Platform for Mobile Broadband Measurements](#). *Computer Networks, Special Issue on Future Internet Testbeds 61* (Mar. 2014), 88–101. ISSN 1389-1286.
- [49] LIM, Y.-S., ET AL. [How green is Multipath TCP for mobile devices?](#) In *AllThingsCellular '14* (2014), ACM.
- [50] LINGLI, D., LIU, D., SUN, T., BOUCADAIR, M., AND CAUCHIE, G. [Use-cases and Requirements for MPTCP Proxy in ISP Networks](#). Internet-Draft draft-deng-mptcp-proxy-01, IETF Secretariat, Oct. 2014. I-D Exists.
- [51] NÉMETH, F., SONKOLY, B., CSIKOR, L., AND GULYÁS, A. [A large-scale multipath playground for experimenters and early adopters](#). In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (New York, NY, USA, 2013), SIGCOMM '13, ACM, pp. 481–482.
- [52] NGUYEN, K., JI, Y., AND YAMADA, S. [A cross-layer approach for improving WiFi performance](#). In *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International* (Aug 2014), pp. 458–463.
- [53] NICUTAR, C., PAASCH, C., BAGNULO, M., AND RAICIU, C. [Evolving the internet with connection acrobatics](#). In *Proceedings of the 2013 Workshop on Hot Topics in Middleboxes and Network Function Virtualization* (New York, NY, USA, 2013), HotMiddlebox '13, ACM, pp. 7–12.
- [54] NISHIDA, Y. [MPTCP implementation on ns-2](#). <http://www.jp.nishida.org/mptcp/>, 2010.
- [55] NONG, T.-H., WONG, R., ALMUHTADI, W., AND MELZER, J. [Aggregating internet access in a mesh-backhauled network through MPTCP proxying](#). In *Computing, Networking and Communications (ICNC), 2014 International Conference on* (Feb 2014), pp. 736–742.

- [56] PAASCH, C. *Improving Multipath TCP*. PhD thesis, UCL, November 2014.
- [57] PAASCH, C., BISWAS, A., AND HAAS, D. *Making multipath tcp robust for stateless webservers*. Tech. Rep. draft-paasch-mptcp-syncookies-00, IETF, April 2015.
- [58] PAASCH, C., AND BONAVENTURE, O. *MultiPath TCP Low Overhead*. Internet-Draft draft-paasch-mptcp-lowoverhead-00, IETF Secretariat, Oct. 2012.
- [59] PAASCH, C., AND BONAVENTURE, O. *Securing the MultiPath TCP handshake with external keys*. Internet-Draft draft-paasch-mptcp-ssl-00, IETF Secretariat, Oct. 2012.
- [60] PAASCH, C., AND BONAVENTURE, O. *A generic control stream for Multipath TCP*. Internet-Draft draft-paasch-mptcp-control-stream-00, IETF Secretariat, Feb. 2014.
- [61] PAASCH, C., AND BONAVENTURE, O. *Multipath TCP*. *Commun. ACM* 57, 4 (Apr. 2014), 51–57.
- [62] PAASCH, C., DETAL, G., DUCHENE, F., RAICIU, C., AND BONAVENTURE, O. *Exploring Mobile/WiFi Handover with Multipath TCP*. In *ACM SIGCOMM CellNet workshop* (2012), pp. 31–36.
- [63] PAASCH, C., FERLIN, S., ALAY, O., AND BONAVENTURE, O. *Experimental evaluation of Multipath TCP schedulers*. In *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop* (New York, NY, USA, 2014), CSWS '14, ACM, pp. 27–32.
- [64] PAASCH, C., KHALILI, R., AND BONAVENTURE, O. *On the benefits of applying experimental design to improve multipath tcp*. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies* (New York, NY, USA, 2013), CoNEXT '13, ACM, pp. 393–398.
- [65] PENG, Q., CHEN, M., WALID, A., AND LOW, S. *Energy efficient Multipath TCP for mobile devices*. In *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2014), MobiHoc '14, ACM, pp. 257–266.
- [66] PENG, Q., WALID, A., HWANG, J., AND LOW, S. *Multipath tcp: Analysis, design, and implementation*. *Networking, IEEE/ACM Transactions on PP*, 99 (2015), 1–1.
- [67] PLUNTKE, C., EGGERT, L., AND KIUKKONEN, N. *Saving mobile device energy with Multipath TCP*. In *Proceedings of the Sixth International Workshop on MobiArch* (New York, NY, USA, 2011), MobiArch '11, ACM, pp. 1–6.
- [68] RAICIU, C., BARRE, S., PLUNTKE, C., GREENHALGH, A., WISCHIK, D., AND HANDLEY, M. *Improving Datacenter Performance and Robustness with Multipath TCP*. In *ACM SIGCOMM 2011* (2011).
- [69] RAICIU, C., HANDLEY, M., AND WISCHIK, D. *Coupled Congestion Control for Multipath Transport Protocols*. RFC 6356, October 2011.
- [70] RAICIU, C., IYENGAR, J., AND BONAVENTURE, O. *Recent advances in reliable transport protocols*. In *Recent Advances in Networking* (August 2013), H. Haddadi and O. Bonaventure, Eds., ACM SIGCOMM, pp. 59–106.
- [71] RAICIU, C., NICULESCU, D., BAGNULO, M., AND HANDLEY, M. J. *Opportunistic mobility with Multipath TCP*. In *Proceedings of the Sixth International Workshop on MobiArch* (New York, NY, USA, 2011), MobiArch '11, ACM, pp. 7–12.

- [72] RAICIU, C., PAASCH, C., BARRE, S., FORD, A., HONDA, M., DUCHENE, F., BONAVENTURE, O., AND HANDLEY, M. [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* (Berkeley, CA, USA, 2012), NSDI'12, USENIX Association, pp. 29–29.
- [73] SCHARF, M., AND FORD, A. [Multipath TCP \(MPTCP\) Application Interface Considerations.](#) RFC 6897, March 2013.
- [74] TACETTIN AYAR, B. R., BUDZISZ, L., AND WOLISZ, A. [A Splitter/Combiner Architecture for TCP over Multiple Paths.](#) Tech. Rep. TKN-12-001, TKN, Tu Berlin, February 2012.
- [75] TACHIBANA, A., YOSHIDA, Y., SHIBUYA, M., AND HASEGAWA, T. [Implementation of a proxy-based cmt-sctp scheme for android smartphones.](#) In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on* (Oct 2014), pp. 660–665.
- [76] TARIQ, S., AND BASSIOUNI, M. [Performance evaluation of MPTCP over optical burst switching in data centers.](#) In *Telecommunications Symposium (ITS), 2014 International* (Aug 2014), pp. 1–5.
- [77] TAZAKI, H., UARBANI, F., MANCINI, E., LACAGE, M., CAMARA, D., TURLETTI, T., AND DABBOUS, W. [Direct code execution: Revisiting library os architecture for reproducible network experiments.](#) In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies* (New York, NY, USA, 2013), CoNEXT '13, ACM, pp. 217–228.
- [78] VO, P., LE, T., LEE, S., HONG, C., KIM, B., AND SONG, H. [mReno: a practical multipath congestion control for communication networks.](#) *Computing* 96, 3 (2014), 189–205.
- [79] WALID, A., PENG, Q., HWANG, J., AND LOW, S. [Balanced Linked Adaptation Congestion Control Algorithm for MPTCP.](#) Internet-Draft draft-walid-mptcp-congestion-control-01, IETF Secretariat, July 2014. I-D Exists.
- [80] WEI, X., XIONG, C., AND LOPEZ, E. [MPTCP proxy mechanisms.](#) Internet-Draft draft-wei-mptcp-proxy-mechanism-01, IETF Secretariat, Mar. 2015.
- [81] WILLIAMS, N., ABEYSEKERA, P., DYER, N., VU, H., AND ARMITAGE, G. [Multipath TCP in Vehicular to Infrastructure Communications.](#) Tech. Rep. 140828A, CAIA, Swinburne University of Technology, August 2014.
- [82] WILLIAMS, N., STEWART, L., AND ARMITAGE, G. [Design Overview of Multipath TCP version 0.4 for FreeBSD-11.](#) Tech. Rep. 140822A, Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, 22 August 2014.
- [83] WISCHIK, D., HANDLEY, M., AND BRAUN, M. B. [The resource pooling principle.](#) *SIGCOMM Comput. Commun. Rev.* 38, 5 (Sept. 2008), 47–52.
- [84] WISCHIK, D., RAICIU, C., GREENHALGH, A., AND HANDLEY, M. [Design, implementation and evaluation of congestion control for Multipath TCP.](#) In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2011), NSDI'11, USENIX Association, pp. 99–112.
- [85] XU, M., CAO, Y., AND DONG, E. [Delay-based Congestion Control for MPTCP.](#) Internet-Draft draft-xu-mptcp-congestion-control-01, IETF Secretariat, Jan. 2015. I-D Exists.

- [86] YAMAUCHI, D., AND ITO, Y. [A study of effect of MPTCP on Web usability](#). In *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on* (Oct 2014), pp. 12–15.
- [87] ZHOU, D., SONG, W., WANG, P., AND ZHUANG, W. [Multipath TCP for user cooperation in LTE networks](#). *Network, IEEE* 29, 1 (Jan 2015), 18–24.
- [88] ZHOU, J., TEWARI, M., ZHU, M., KABBANI, A., POUTIEVSKI, L., SINGH, A., AND VAHDAT, A. [WCMP: Weighted cost multipathing for improved fairness in data centers](#). In *Proceedings of the Ninth European Conference on Computer Systems* (New York, NY, USA, 2014), EuroSys '14, ACM, pp. 5:1–5:14.