Is Multi-Path Transport Suitable for Latency Sensitive Traffic?

Kiran Yedugundla[a,*], Simone Ferlin[b], Thomas Dreibholz[b], Özgü Alay[b], Nicolas Kuhn[c,d], Per Hurtig[a], Anna Brunstrom[a]

[a] *Dept. of Computer Science, Karlstad University, Karlstad, Sweden*
[b] *Simula Research Laboratory, Oslo, Norway*
[c] *IMT Télécom Bretagne, IRISA, Cesson-Sévigné, France*
[d] *Centre National d'Etudes Spatiales (CNES), Toulouse, France*

Abstract

This paper assesses whether multi-path communication can help latency-sensitive applications to satisfy the requirements of their users. We consider Concurrent Multi-path Transfer for SCTP (CMT-SCTP) and Multi-path TCP (MPTCP) and evaluate their proficiency in transporting video, gaming, and web traffic over combinations of WLAN and 3G interfaces. To ensure the validity of our evaluation, several experimental approaches were used including simulation, emulation and live experiments. When paths are symmetric in terms of capacity, delay and loss rate, we find that the experienced latency is significantly reduced, compared to using a single path. Using multiple asymmetric paths does not affect latency – applications do not experience any increase or decrease, but might benefit from other advantages of multi-path communication. In the light of our conclusions, multi-path transport is suitable for latency-sensitive traffic and mature enough to be widely deployed.

Keywords: Internet, latency, multi-path communication, transport protocols, MPTCP, CMT-SCTP

1. Introduction

Live and interactive applications are sensitive to latency, as the user experience is negatively affected when data is delayed. For instance, freezing a live video just 1% of the video duration is sufficient to turn away 5% of the viewers [1]. Similarly, a latency of 60 ms suffices to degrade user experience in Internet gaming [2]. Multiple ways of improving the user experience of latency sensitive applications are active subjects of research. However, as far as we know, a weakly explored area is to determine whether utilizing all available network interfaces at the end host could improve such experience. In recent times, deployed devices such as tablets and smartphones are often equipped with both Wireless LAN (WLAN) and cellular 3G or 4G interfaces.

*Corresponding author
Email addresses: kirayedu@kau.se (Kiran Yedugundla), ferlin@simula.no (Simone Ferlin), dreibh@simula.no (Thomas Dreibholz), ozgu@simula.no (Özgü Alay), nicolas.kuhn@cnes.fr (Nicolas Kuhn), perhurt@kau.se (Per Hurtig), annabrun@kau.se (Anna Brunstrom)

Multi-path transmission has been proposed to guarantee better resilience to link failures and a better use of resources. For instance, consider a connection using two interfaces simultaneously; if one of the interfaces (or underlying links) fails, the transmission can simply continue over the other interface. In a single-interface scenario, the transmission would be stalled and maybe require a connection re-establishment. It has also been shown that simultaneous transmission of data over multiple interfaces can increase the throughput, due to capacity aggregation [3]. Even if multi-path protocols have been shown to be more resilient to link failures and able to aggregate capacity to provide increased throughput, the impact of using multiple paths on latency has not been thoroughly investigated.

This paper fills this gap by assessing whether multi-path approaches are suitable transport protocols for applications transmitting latency-sensitive traffic, e.g., video, gaming and web traffic. Recent efforts within the Internet Engineering Task Force (IETF) include designing Multi-path

TCP (MPTCP) [4] extensions to TCP [5] to enable end-to-end connections to span multiple paths simultaneously. Similarly, Concurrent Multipath Transfer for SCTP (CMT-SCTP) [6, 7, 8] is an extension to the Stream Control Transmission Protocol (SCTP) [9], enabling simultaneous multi-path communication. We therefore evaluate their suitability to carry out latency sensitive traffic.

In our experiments we consider both symmetric multi-path communication (e.g. WLAN-WLAN) as well as asymmetric (e.g. WLAN-3G). For the actual evaluations we use a combination of simulations, emulations and real experiments to ensure a correct assessment.

The remainder of this paper is structured as follows. Section 2 presents an overview of CMT-SCTP and MPTCP, and how these protocols solve the core issues inherent in transport-level multi-path communication. Section 3 describes the applications used in our evaluation and their latency requirements. In Section 4, the experimental setup is detailed. Section 5 presents and explains the results obtained. In addition to that, Section 6 provides an in-depth discussion of the results. Section 7 discusses related work on multi-path transport. Finally, Section 8 concludes the paper and discusses possible future work in this area.

2. Multi-Path Transport

This section introduces CMT-SCTP and MPTCP, the current key multi-path transport protocols. The core issues of multi-path communication, and how these are addressed by CMT-SCTP and MPTCP, are then described.

2.1. CMT-SCTP

SCTP [9, 10] is a transport protocol originally developed by the IETF Signaling Transport (SIGTRAN) Working Group [11], as part of an architecture to provide reliable and timely message delivery for Signaling System No. 7 (SS7) [12] telephony signaling information, on top of the Internet Protocol (IP) [13]. While motivated by the need to carry signaling traffic, SCTP was designed as a general purpose transport protocol on par with TCP [5] and UDP [14]. While SCTP can offer functionality similar to TCP, such as ordered and reliable transmission or congestion controlled transport, its options can be easily set so that SCTP rather features unordered transmission or multi-homing. This flexibility is one main advantage of SCTP as opposed to TCP.

The multi-homing feature of SCTP allows a single association (or connection) between two endpoints to combine multiple source and destination IP addresses. These IP addresses are exchanged and verified during the association setup, and each destination address is considered as a different path towards the corresponding endpoint. Using the Dynamic Address Reconfiguration protocol extension [15], it is also possible to dynamically add or delete IP addresses, and to request a primary-path change, during an active SCTP association.

While SCTP multi-homing [10, 9] targets robustness and uses only one active path at a time, several researchers have suggested the concurrent use of all paths for sending data. Budzisz et al. [16] provides a survey of these approaches. In this paper, we consider the most complete of these proposals, Concurrent Multipath Transfer for SCTP (CMT-SCTP) [6, 7, 8]. CMT-SCTP improves the internal buffer management procedures of SCTP, transmission over multiple paths and reordering with its single sequence-number space. Assuming disjoint paths, CMT-SCTP applies the original SCTP congestion control [9] for each path independently.

2.2. MPTCP

Multi-Path TCP (MPTCP) [17] is a set of extensions to TCP [5, 18] developed by the IETF MPTCP working group [19] to enable simultaneous use of multiple paths between endpoints. The motivation behind MPTCP is more efficient resource usage and improved user experience through improved resilience to network failure and higher throughput.

To use the MPTCP extensions the initiator of a connection appends a "Multipath Capable" (MP_CAPABLE) option in the SYN segment, indicating its support for MPTCP. When the connection is established, it is possible to add one TCP flow, or subflow, per available interface to this connection by using a "MPTCP Join" (MP_JOIN) option in the SYN segment. Once the MPTCP connection has been fully established, both end hosts can send data over any of the available subflows.

While MPTCP transparently divides user data among the subflows, simultaneous transmission may cause connection-level packet reordering at the receiver. To handle such reordering, two levels of

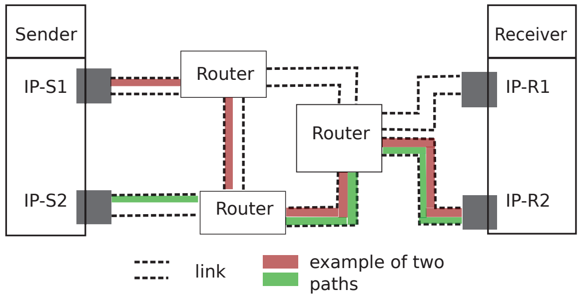


Figure 1: Definition of a path as a sequence of links between a sender and a receiver

sequence numbers are used. Apart from the regular TCP sequence numbers that are used to ensure in-order delivery at subflow level, MPTCP uses a 64-bit data sequence number that spans the entire MPTCP connection and can be used to order data arriving at the receiver. To ensure fairness [20] on bottleneck links shared by subflows of a MPTCP flow and other TCP flows, MPTCP extends the standard TCP congestion control. Running existing TCP congestion control algorithms independently would give MPTCP connections more than their fair share of the capacity if a bottleneck is shared by two or more of its subflows. To solve this MPTCP uses a coupled congestion control [21] that links the increase functions of each subflows' congestion control and dynamically controls the overall aggressiveness of the MPTCP connection. The coupled congestion control also makes resource usage more efficient as it steers traffic away from more congested paths to less congested ones.

2.3. Core Issues

This section presents the core issues that are related to the use of multiple paths and how they are addressed by CMT-SCTP and MPTCP.

2.3.1. Path Management

As shown in Figure 1, a path is a sequence of links between a sender and a receiver [4], over which it is possible to open a subflow. A multi-path protocol must define a path management strategy. The strategy needs to find suitable paths to open subflows over and decide whether one or more subflows should be opened over a specific path. For short or extremely time-sensitive flows, the choice of path for the initial connection establishment might be very important. For example, if (i) two paths (p_1 and p_2) are available, (ii) both paths have the same capacity and (iii) the RTT of p_1, r_1, is significantly higher than the RTT of p_2, r_2, (e.g. $r_1 > 10 \times r_2$), then whether the first subflow will be opened over p_1 or p_2 would seriously impact the latency. The number of subflows to open over a path is a problem that is not very well studied. While the Linux implementation of MPTCP supports this using its ndiffports path manager, as described later in this section, it is often regarded as unnecessary to open more than one subflow per path as they typically would traverse the same links and compete for the same network resources. However, in some specific environments, e.g. datacenters, the network might conduct load balancing between subflows, routing them over disjoint subpaths. In such situations there might be benefits of creating several subflows per path, as shown in [22].

For CMT-SCTP a path is defined by the destination IP address and port number. To manage paths, CMT-SCTP employs a simple strategy where the association is established during a 4-way handshake in which available IP addresses are exchanged and verified. The handshake is conducted over the default interface of the host, and after its completion each destination address is considered as a path and implicitly also as an opened subflow. The interfaces of the hosts are pairwise connected over two different subnets, resulting in two possible paths. In addition to the establishment of subflows during the association setup, there is an extension to CMT-SCTP called Dynamic Address Reconfiguration (DAR) [15] which enables an end-host to dynamically add and remove IP addresses to an existing connection.

Like CMT-SCTP, MPTCP consults the routing table to determine which interface to initiate the connection over. During the establishment phase, realised by a 3-way handshake, IP address information are exchanged between the hosts in a fashion similar to that of CMT-SCTP. However, after connection establishment, MPTCP cannot make full use of the other host's IP address information and start sending data over all paths straight away. Notably, at this point it can only use the path used for the connection establishment. Additional subflows must be opened as separate TCP connections and joined to the MPTCP connection using the "MPTCP Join" option in their SYN segments. Another difference, as compared to CMT-SCTP, is the availability of multiple path managers in MPTCP. For example, the Linux implementation of MPTCP provides four different path managers: default, full-mesh, ndiffports and binder. Using the default path manager, a host does not advertise additional IP addresses but uses the other hosts advertised IP addresses to create new subflows. The full-mesh strategy uses an opposite approach: all available IP addresses are exchanged and used to open a subflow over each and one of all the possible source-destination IP address combinations. The ndiffports manager allows a user to open X subflows over the default interface. Finally, the binder manager, implements Loose Source Routing as defined in [23]. Similar to CMT-SCTP, MPTCP also includes an extension to allow dynamic addition and removal of IP addresses.

2.3.2. Scheduling

If multiple subflows are available, there are different ways to schedule the transmission of data. As an example, a round-robin scheduler may iterate over the available subflows and try to transmit an entire congestion window over each subflow, while another scheduler might only consider the "fastest" available subflow. Scheduling in multi-path communication has therefore a large impact on the performance of data transmission.

One root cause for scheduling problems is the use of paths with asymmetric characteristics. Figure 2 illustrates how asymmetric paths, in terms of RTT ($RTT_2 = 10 \times RTT_1$), can affect data transmission. Figure 2c, shows the so-called head-of-line blocking problem. In this scenario, packets #3 and #4 (residing in the receiver's buffer) cannot be delivered to the application as packets #1 and #2 are still in flight. Further, packets sent over the fast subflow can fill the receiver's buffer while waiting for data transferred over the slow subflow. This issue is known as receiver buffer blocking and is illustrated in Figure 2d.

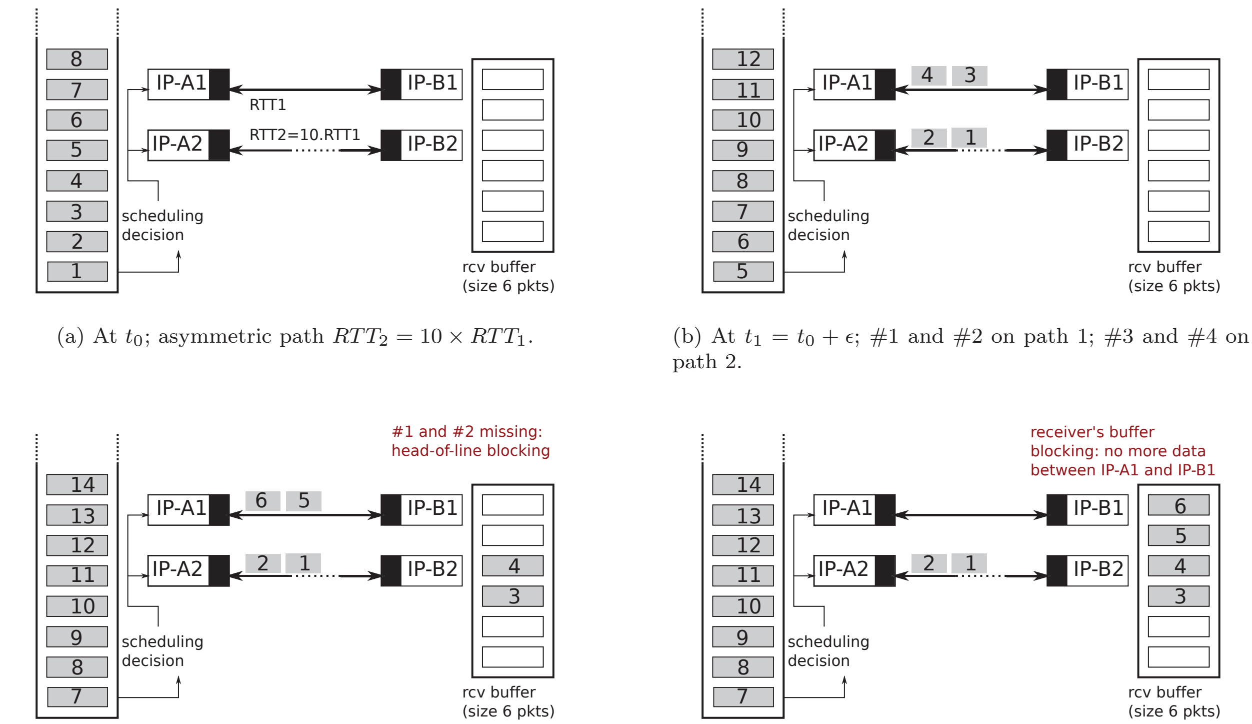
The usual scheduler in CMT-SCTP is a round-robin scheme targeting throughput maximization: for every subflow in sequence, starting from the primary one, it sends as much data over the subflow as the congestion window allows. As mentioned earlier, asymmetric paths can be problematic and this is also true for CMT-SCTP. The problem is due to a combination of the scheduling and occupancy of the shared send and/or receive buffer space, and can cause the aforementioned problems of head-of-line blocking and receiver buffer blocking. Detailed classifications of the blocking issues are provided in [24]. To remedy this problem, other schedulers have been proposed and developed for CMT-SCTP. For example, chunk rescheduling [25, 8] is a mechanism that re-injects the segment causing head-of-line blocking on a different subflow that has space available in its congestion window. Furthermore, Delay-Aware Packet Scheduling (DAPS) [26] is a scheduler that, given the RTT of the different subflows, tries to send packet sequences over them in a manner that guarantees in-order delivery at the receiver.

Similar to CMT-SCTP, several schedulers have been proposed for MPTCP. In the Linux implementation the default scheduler always tries to transmit data over the subflow with the shortest RTT, as long as there is free space in the congestion window. The default scheduling also includes a mechanism called Retransmission and Penalization (RP). This mechanism is similar to CMT-SCTP's chunk rescheduling and re-injects segments causing head-of-line blocking in a different subflow. In addition to the default scheduling mechanism, a weighted round-robin scheme is also available. The schedulers available for Linux have all been evaluated and compared in [27], identifying the shortest-RTT scheduler as the most successful in terms of throughput performance. The different schedulers are detailed in [28].

2.3.3. Congestion Control for Multi-Path Transport

When using multiple paths for transmission, different subflows cannot share a single congestion window, as each subflow is likely to have different characteristics and levels of congestion. There are, however, situations in which the subflows actually do share a bottleneck and thus have the same level of congestion. In such scenarios there must be some kind of collaboration between the congestion controllers of each subflow to ensure that the transport does not achieve more than its fair share of the network resources.

For CMT-SCTP there is no default congestion control mechanism that manages the transmission based on the combined congestion state of the subflows. This is likely due to an initial design assumption that subflows do not share bottlenecks. As discussed above, such an assumption is not always true, making CMT-SCTP potentially unfair to other traffic in the network. This problem has been addressed by several researchers and coupled congestion controllers have been proposed. Examples include e.g., CMT/RPv1 [29] and CMT/RPv2 [30].



(a) At t_0; asymmetric path $RTT_2 = 10 \times RTT_1$.

(b) At $t_1 = t_0 + \epsilon$; #1 and #2 on path 1; #3 and #4 on path 2.

(c) At $t_2 = t_1 + RTT_1$; #3 and #4 are received but can not be read because #1 and #2 are missing ⇒ head-of-line blocking.

(d) At $t_3 = t_1 + 2 \times RTT_1$; #1 and #2 are not received and the receiver's buffer can not receive #7 ⇒ receiver buffer blocking.

Figure 2: Head-of-Line Blocking and Receive Buffer Blocking

The problem of not considering shared bottlenecks was addressed already in the design phase of MPTCP. The reason to why coupled congestion control should be used, the benefits of using it, and what goals it has to achieve are all documented in [21]. To achieve these goals, various coupled congestion control schemes have been proposed for MPTCP. These include the Linked-Increases Algorithm (LIA) [21], the Opportunistic Linked-Increases Algorithm (OLIA) [31] and the BALanced Linked Adaptation (BALIA) [32]. At the time of writing, the default congestion control in the Linux MPTCP implementation is LIA.

2.3.4. Handling Loss and Retransmissions

When data is lost in multi-path transmission the protocol must decide whether to retransmit this data over the same subflow or over a different one.

CMT-SCTP features several schemes for retransmitting data, all detailed in [33]. A CMT-SCTP sender maintains accurate information about the working paths, as new data are transmitted over every available subflow concurrently. Therefore, many distinct strategies can be used. For example, retransmitting lost data over the same subflow, over the subflow with the largest slow-start threshold or using the subflow with the largest congestion window. While there is no default retransmission strategy for CMT-SCTP, we consider retransmissions over the subflow with lowest RTT to give latency-sensitive applications a benefit.

In MPTCP, the loss detection is performed at two levels: subflow level and MPTCP level. While loss typically is detected on subflow level, different strategies can be taken depending on how the loss was detected. If the loss is detected by the fast retransmit algorithm, data is only retransmitted over the same subflow. If, on the other hand, the loss is detected by an expiration of the RTO timer, the data can be retransmitted over both the same subflow and over an additional subflow chosen by the scheduler. The rationale for using different retransmission approaches depending on how the loss was detected is straightforward; fast retransmit only detects loss if feedback from packets sent after the lost packet(s) arrive at the receiver. Therefore, it is safe to assume that no massive congestion event or link breakage has happened, and that a retransmission will arrive safely at the receiver. If no feedback is received, however, the RTO will eventually expire, and it is then safer to retransmit the lost packet(s) over both paths in case the path over which the original transmission occurred is experiencing major congestion or other serious problems.

3. Applications and their Requirements for Multi-Path Transport

Traditionally, Internet has been dominated by web traffic running on top of short-lived TCP connections [34]. For example, Ciullo et al. [35] found that approximately 95% of the client TCP flows and 70% of the server TCP flows were less than 10 segments. Although web traffic still constitutes a large fraction of all traffic, video traffic and gaming traffic are now becoming more common. Recent measurements [36] show e.g. that more than 53% of the downstream traffic in North America is video streaming. Forecasts (cf. [37]) also show that Internet video and gaming will continue to grow with an annual compound growth rate of 29% for video traffic and 22% for gaming.

Although the aforementioned traffic classes differ significantly in many ways, they have a common property – sensitivity to latency. In this paper we will therefore use video, gaming and web traffic to assess whether multi-path protocols are suitable for latency-sensitive applications. The remainder of this section describes the main characteristics of the applications and discusses their requirements.

3.1. Video Streaming

There are two main use-cases of video streaming: Video on Demand (VoD) which is not broadcast live and therefore do not have stringent latency requirements; and direct live video which is broadcast live and have requirements of low latency.

VoD applications has complete knowledge of the content to transfer, and can therefore adapt the sending rate appropriately. The quality of experience of VoD is therefore less vulnerable to one-way delay variations than the quality of experience of direct live video. As the rationale of this work is to assess whether multi-path transport protocols can be used for latency sensitive applications, we will focus on direct live video as it is more sensitive to latency.

The direct live video can be divided into two sub-categories: live broadcast of TV such as BBC iPlayer[1] and private video communications such as Skype[2]. We have focused on the latter category as such applications typically are interactive in their nature and thus more sensitive to latency. Although Skype is proprietary and its communication protocol is closed, and may change over time, we use Skype-like video traffic in our evaluation. We do this for several reasons. First, Skype is a widely used application. Actually, Skype generates almost two percent of the total aggregate traffic in European fixed networks [38]. Second, although Skype mainly tries to use UDP for communication NATs and firewalls often force it to use TCP, making it an interesting use case for our experiments with multi-path and reliable transports. Finally, Skype traffic is well studied and traffic characteristics have been reported by several researchers, making it relatively easy to model. According to [39, 40], it dynamically adapts its sending rate to the network conditions, with a frame rate per second going from 5 frames/s to 30 frames/s and a video bit rate from 30 Kbit/s to 950 Kbit/s.

Requirements: The latency requirements for a good user experience when considering live video communication are: one-way delay should be lower than 150 ms [41] and the difference in delay between packets (jitter) should be lower than 30 ms [41].

3.2. Gaming Traffic

Online gaming is often categorized into three different classes [42] each of them being characterized by specific traffic, as detailed in [43]. The classes are:

- first person avatar, e.g. First Person Shooter games (FPS)
- third person avatar, e.g. Massive Multiplayer Online games (MMO)
- omnipresent, e.g. Real Time Strategy games (RTS)

[1] http://www.bbc.co.uk/iplayer/live/bbcnews/
[2] https://www.skype.com/

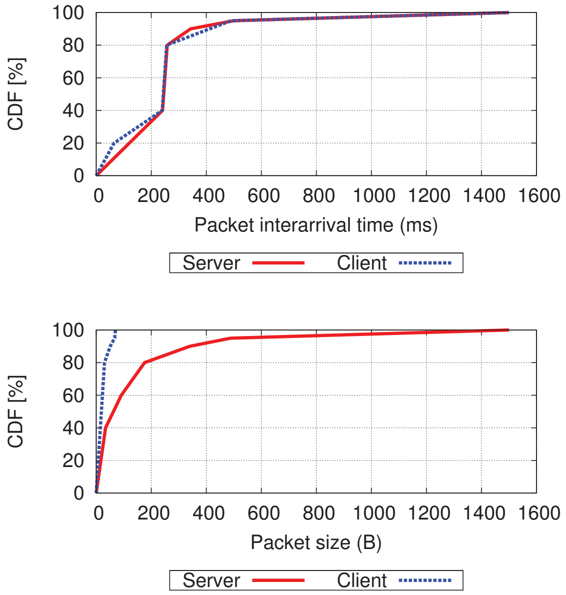


Figure 3: Distribution of Packet Inter-Arrival Time and Packet Size for World of Warcraft [43]

FPS games are tolerant to loss but are very delay sensitive, therefore they often use UDP as transport. MMO games, on the other hand, are less loss tolerant and require less bandwidth compared to FPS games, therefore, a mixture of TCP and UDP is used for transmission. TCP traffic of MMO games is composed of multiple thin TCP flows. Thin flows are characterized by a low transmission rate where the majority of packets are much smaller than the maximum transmission unit (MTU). An example of the traffic characteristics between the server and a client of an MMO game is illustrated in Figure 3. For RTS games, interestingly, latency has a negligible effect on the outcome of the game, indicating that RTS game-play clearly favors strategy over the real-time aspects [44].

Considering the popularity of MMO games [43], and the fact that they use TCP, this paper assesses whether there are any benefits in using multiple paths at the transport layer to carry the traffic generated by an MMO game entitled Age of Conan.

Requirements: The requirements for a good gaming experience highly depend on the class of the game and the particular game itself. However, low latency (lower than 60 ms is indicated in [2]) and a small delay variation [45] are important for a good gaming experience.

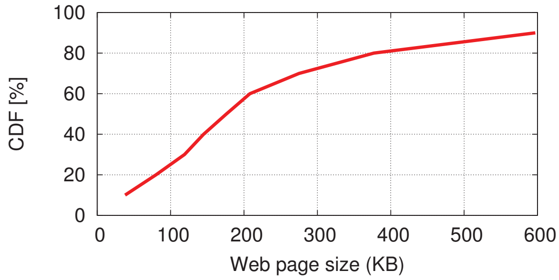


Figure 4: Distribution of Web Page Sizes according to [47]

3.3. Web Traffic

Figure 4 illustrates the distribution of common web site sizes. In our experiments, to be representative of the web, we have selected three web sites of different sizes: small (72 KiB), medium (1024 KiB) and large (3994 KiB). We also use software that emulates the behavior of a real browser, downloading the web sites using 6 concurrent connections over HTTP/1.1. More details on the web traffic can found in Section 4.3.3.

Requirements: The quality of user experience when accessing a web page is highly linked to the download completion time. For example, in [46], the authors report that Google measured that “an additional 500 ms to compute (a web search) [...] resulted in a 25% drop in the number of searches done by users.”. Although the download completion time may not be the most relevant metric for modern browsers, as they often start rendering pages before completion, it is the most suitable metric to use when evaluating transports as it is “browser agnostic” and therefore neutral. For good web browsing experience, the download completion time is required to be as low as possible.

4. Experiment Setup

This section describes the experiment setup used for the performance evaluation of our target applications. The protocol implementations and network models used in the evaluations are also introduced.

4.1. Evaluation Tool Sets

In this study, we focus on the default and latest versions of the protocols. Our choice of evaluation tool sets, presented in this section, has been made on the basis of availability of source code and the fact that we wanted to consider both controlled and real life experiments.

4.1.1. Simulations, CMT-SCTP using OMNeT++

There is no stable implementation of CMT-SCTP for FreeBSD or Linux. Therefore, we could not perform emulations or real experiments using CMT-SCTP. Instead, we performed simulations using OMNeT++ [48] version 5.0b1 with the CMT-SCTP model [8, 49, 50], and the NetPerfMeter application model [51, 8] in the latest version of the INET Framework [52], using the simulation processing tool-chain SimProcTC [53, 54]. For the web traffic simulations in Section 5.3, the HttpTools [55] models provided as part of the INET Framework are used. It was only necessary to add SCTP support. The complete INET Framework sources branch used for this paper is available online[3]. Most changes have already been merged upstream.

Although there is an implementation of CMT-SCTP available for NS-2 [6], it is unmaintained and as of spring 2016 it is fairly out of date. The OMNeT++ implementation, which is used for our evaluations, includes the latest improvements and options for SCTP and is therefore representing the state-of-the-art in SCTP features.

4.1.2. Emulations, MPTCP in a Controlled Environment, using CORE

Because we wanted to evaluate MPTCP in a controlled environment, we ran experiments using the Linux MPTCP implementation and the Common Open Research Emulator (CORE) [56]. CORE enables the use of real protocols and applications together with emulated network links, making the evaluation of MPTCP easy to control and replicate. The Linux kernel implementation is the most complete MPTCP implementation available, so this setup also allowed us to use the most feature complete version of MPTCP. Using the same MPTCP implementation for both the controlled experiments in CORE and the real life testbed experiments also allowed us to more easily compare and validate the results.

4.1.3. Experiments, MPTCP in a Real-Life Environment, using NorNet

Because we may drive biased conclusions if the protocols were to be evaluated only in controlled environments, we also assessed their performance using an environment where the network is used by many other applications than the one we introduce in the network.

In order to realize this, we performed real network experiments on a dedicated testbed, namely NorNet Edge (NNE) [57]. NNE is a multi-homed testbed where each node is connected to multiple UMTS operators via Huawei E392-u12 modems as well as WLAN network. More specifically, in our evaluations, we consider one operational UMTS Mobile Broadband (MBB) network in Oslo, Norway. It is labelled as "3G". The WLAN access point is a public WLAN hotspot, connecting around 100 people during work hours in a large office complex with several interfering WLAN networks. The two WLAN networks used for the WLAN-WLAN scenarios are using the same technology (IEEE 802.11ag) and sharing the same medium, therefore certain level of interference is highly likely depending on the number of users and traffic patterns. We believe, this setup reflects a realistic scenario where the users cannot control these factors. The downside of this testbed experiments is that there might be a statistically insignificant and uncontrolled behavior for few packets. Furthermore, in this paper, we focus on the transport layer, therefore we study how transport layer reacts to such realistic path characteristics. In this real-world environment, MPTCP was tested under different scenarios for different applications as in the emulation setup.

4.2. Configuration of MPTCP and CMT-SCTP

Both MPTCP and CMT-SCTP have open source implementations, making it possible to enable and/or disable specific features. Due to readability we have chosen to only present the most important features, and their settings, in this section. Additionally, Table 1 provides a short summary of this information. For a full description of the protocol configurations and for all the experimental scripts and data, please see [58].

For MPTCP, we used the state-of-the-art Linux MPTCP implementation (v0.89.3)[4]. We use the default options of MPTCP, including e.g. receive buffer optimization and coupled congestion control, with an exception for the Nagle algorithm, which is turned off in all application scenarios. Turning off

[3] https://github.com/dreibh/inet/tree/td-netperfmeter-for-integration.

[4] Linux MPTCP: http://www.multipath-tcp.org.

Nagle is common practice when running applications that require low latency [59].

The simulation uses the CMT-SCTP model for OMNeT++ that is fully described in [8, 49]. As opposed to MPTCP, no default options are given for CMT-SCTP. The latest version of the SCTP simulation model [50, 49] for OMNeT++ is used, implementing SCTP according to RFC 4960 [9] with all state-of-the-art features and extensions. In addition to the settings listed in Table 1, CMT-SCTP was configured with the following features:

- burst mitigation with MaxBurst=4 (default from [9, Section 15]) with "Use It or Lose It" [60] strategy (i.e. behavior like the FreeBSD SCTP implementation [50]);
- buffer splitting [25, 24] to avoid buffer blocking issues [8, Section 7.5].

All data is sent in SCTP/MPTCP messages of up to 1,452 bytes (resp. 1,428 bytes), which corresponds approximatively to full packets (including headers) of 1,500 bytes. In CMT-SCTP, the size of the payload depends on the number of chunks that are gathered in one message, therefore the full packet sizes may vary depending on the application profile.

As explained in Section 2.3.1, (i) with CMT-SCTP, one subflow can be opened on each working path as soon as the 4-way handshaking process has been operated and the primary path (i.e., the first path on which data is transmitted) has to be defined; (ii) with MPTCP, the first subflow that is opened depends on the parameterization of the Linux default interface. In our evaluations, when the paths are homogeneous (i.e., WLAN-WLAN or 3G-3G), the path on which the first subflow is opened is chosen randomly; when the paths are heterogeneous (i.e., WLAN-3G), the WLAN path is used for the first subflow.

4.3. Application Traffic Generation and Metrics

This section presents how we generate video streaming, gaming and web traffics. The rationale for using these applications and the characteristic of the traffic that they generate are detailed in Section 3.

4.3.1. Video Traffic

In this paper, we have not considered Video on Demand traffic since it is hard to accurately model or emulate this traffic in the various cases (emulation, simulation, experimentation) used in this article. Moreover, these applications are not interactive and might be seen as file transfer applications. Therefore, we considered Direct Live Video applications due to their delay sensitive nature. In order to generate Skype-like traffic we considered a constant bit rate application generating 950 Kbit/s with 30 frames/s.

4.3.2. MMO Gaming Traffic

For gaming traffic, we considered a set of trace files from the Massively Multiplayer Online Game Age of Conan, provided by Funcom [61]. These traffic traces extend over a very long time period. As it is extremely difficult and tedious to replay all the traces completely, we selected a set of three traces with a duration of 10 minutes each. The selection of traces was based on the possible full game play being captured in the trace. Full game play constitutes initial loading of game settings, player interaction and infrequent chunk updates depending on the game. All the traces contain a huge chunk of game setup data in the beginning of the connection followed by occasional small bursts of MTU-sized packets and small packets for the rest of the time. The selected traces were replayed using the D-ITG [62] traffic generator. In this process D-ITG is loaded with full trace and it generates packets of exact size and time sequence as seen in the trace file. This is a way of providing trace input to the experiments than generation based on a statistical setting. From here on, the traces are named Trace 1, Trace 2 and Trace 3. They have average packet inter-departure times of 181.4 ms, 74.1 ms, and 167.7 ms respectively. Furthermore, the average packet sizes are 142.7 bytes, 113 bytes, and 101.7 bytes respectively.

4.3.3. Web Traffic

As previously mentioned, we consider three classes (small, medium, large) which are representative for real web sites. The classes and sites are presented in Table 2. For the experiments, we stored the files from the three sites (Wikipedia, Amazon and Huffington Post) on a local server. Each website data contained different umber of objects of different sizes. Of the three, Wikipedia content was the smallest followed by Amazon and Huffington Post. The data stored in the local server was requested and downloaded from a client with 6 concurrent connections.

	Path management Section 2.3.1	Scheduling Section 2.3.2	Congestion control Section 2.3.3	Handling loss Section 2.3.4
MPTCP	full-mesh	(i) LowRTT (ii) Retransmission and penalization	coupled (OLIA)	(i) Fast retransmit on the same subflow (ii) RTO on a subflow chosen by the scheduler
CMT-SCTP	one subflow per working path	(i) packet-based round-robin (ii) chunk rescheduling	uncoupled (NewReno)	retransmission on lowest RTT path

Table 1: Options for MPTCP and CMT-SCTP

Size	Domain name	Number of objects	Size of objects
small	Wikipedia (www.wikipedia.org)	15	72 KiB
medium	Amazon (www.amazon.com)	54	1024 KiB
large	Huffington Post (www.huffingtonpost.com)	138	3994 KiB

Table 2: Web Traffic Generation

4.4. Background Traffic Generation

The congestion level in a network has a significant impact on the behavior of protocols employing congestion control [63]. We therefore conduct experiments both with and without background traffic. Background traffic is generated with NetPerfMeter [64, 51] as a mix of TCP and UDP flows constituting one long TCP flow and 4 UDP on-off flows. The TCP flow has a saturated sender sending as much data as possible with frame size of 1,460 bytes. Each UDP flow generates Pareto on-off traffic with shape 1.5 and scale 0.166667, sending 25 frames per second each of size 5,000 bytes. The aggregate usage of UDP background flows were maintained at 10% of the bottleneck link capacity to be realistic [65]. The UDP flows carry data at an average of 500 Kbit/s each in the WLAN-WLAN scenario and 100 Kbit/s each in the 3G-3G scenario. In each run, the background flows start before the foreground experimental traffic and end after the experimental traffic.

However, in the NorNet experiments even running one experiment with the background traffic, especially for online gaming and video streaming, can eat up all the monthly data quota. Therefore, we decided not to generate background traffic due to the limited data quotas for the 3G subscriptions and in order to provide consistent results, we have run all NorNet experiments without background traffic.

4.5. Network and System Characteristics

4.5.1. Topology

Figure 5 shows the topology used in our evaluations. The same topology was used for the simulations, emulations, and real experiments in NorNet. To understand the basic performance of protocols and highlight their characteristics, a simple topology is more useful than a complex topology. Though the topology can be seen as more general, this type of basic topology is also common practise for evaluating transport protocols, see for instance Common TCP Evaluation Suite [5]. As shown in the figure, there are two paths on the client side and a single path on the server side. In the simulations and emulations the paths that connect the server and client are, of course, modeled. For the experiments to be realistic we used a parameterisation of the paths that is based on measurements that were conducted over NorNet prior to the evaluation.

4.5.2. Path Characteristics

Table 3 shows the capacity, end-to-end delay and packet loss rates of the WLAN and 3G paths that have been measured in the experimental testbed described in Section 4.1. The WLAN links are IEEE 802.11ag. The loss rate is what is experienced on the transport layer (e.g. datagrams), therefore it is the loss ratio after all link layer re-transmission schemes of underlying networks. Path 1 and Path 2 in Figure 5 will be assigned with these characteristics depending on the technology. In Table 3, we

[5] https://tools.ietf.org/html/draft-irtf-iccrg-tcpeval-01

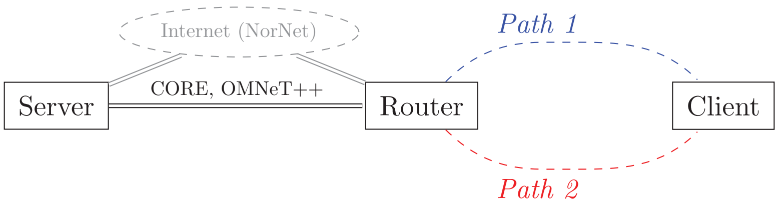


Figure 5: Topology

	WLAN	3G
Capacity [Mbit/s]	20–30	3–5
Propagation Delay [ms]	20–25	65–75
Loss [%]	1–2	0
Homogeneous (WLAN)		
Path 1	x	
Path 2	x	
Homogeneous (3G)		
Path 1		x
Path 2		x
Heterogeneous		
Path 1	x	
Path 2		x

Table 3: Path Characteristics and Scenarios

also detail the three combinations of paths over which the multi-path protocols are evaluated, that are homogeneous WLAN (two WLAN paths), homogeneous 3G (two 3G paths) and heterogeneous (one WLAN and one 3G path).

4.5.3. Buffer Sizes

System characteristics of the source and destination are known to impact the end to end performance of the flows. In order to emulate the realistic network scenarios, we use the system settings close to the standard settings for respective technologies. The TCP buffer sizes (send buffer/receive buffer) are set to be equivalent to the default Android settings, that are configured as follows:

- Homogeneous (3G): 256 KiB/256 KiB.
- Homogeneous (WLAN): 1024 KiB/2048 KiB.
- Heterogeneous (WLAN-3G): 1024 KiB/2048 KiB.

Based on estimations from early measurement in the NorNet testbed, performed during the planning phase of the work, the queue lengths at each interface of the router (see Figure 5) are set to 100 packets for WLAN and 3750 packets for 3G.

Note, that the 3G buffer setting of 256 KiB/256 KiB prevents an overly large bufferbloat [66, 67] in a 3G/3G setup, while the setting of 1024 KiB/2048 KiB will make such a bufferbloat in the WLAN-3G case possible. We will explain this in detail with the results in Section 5 and particularly in Section 5.1.1.

5. Experiment Results

This section presents the experimental evaluation and its results. The protocols are first evaluated through simulations and emulations in controlled environments, to identify the impact of various network parameters. This evaluation is then complemented with measurement results from a real environment.

For each experiment scenario, SCTP is compared to CMT-SCTP, and TCP Cubic is compared to MPTCP. For the homogeneous cases, we consider the average delay using TCP and compare it with that of MPTCP. We assume that we only have information about the technologies used. That is, for a WLAN-WLAN case, the WLAN channels might have different characteristics in terms of loss and delay, but this information is not available to the user. The user will most likely pick one of the WLANs randomly. Therefore, we consider the average WLAN TCP delay performance and compare it with the MPTCP delay performance. However, for the WLAN-3G scenario, the user will most certainly choose WLAN, since it has low delay, high capacity and is probably cheaper to use. Therefore, we compare the MPTCP delay with the TCP delay of WLAN. The evaluation of SCTP and CMT-SCTP is conducted in the same fashion.

5.1. Video Streaming

First, we evaluate video traffic performance for homogeneous and heterogeneous scenarios, considering both competing and non-competing traffic as explained in Section 4.3. We use application layer message delay as the performance metric for the transport protocol latency performance.

5.1.1. CMT-SCTP Simulations

Figure 6 presents the average message delays and the variation in these delays in the form of box plots [68] for video traffic as described in Section 4.1. First, we consider the average SCTP message delay over 128 runs for video traffic without any competing traffic and illustrate the results in Figure 6a. Table 4 presents the percentage of traffic sent over the different paths in CMT-SCTP. Note, that although SCTP uses a certain primary path for payload data transport, there is always a small amount of control traffic (here: mainly heartbeats to check the path status, see [9]) on the other path as well.

In the homogeneous scenarios (WLAN-WLAN and 3G-3G) we observe different behaviors for WLAN-WLAN and 3G-3G cases. In the WLAN-WLAN scenario, with two similar WLAN paths, multi-path transport leads to increased latency, and also increased latency variation, mainly due to the reordering caused by retransmissions on the lossy paths (i.e., 1%–2% packet loss; see Table 1). In the 3G-3G scenario, where both paths are lossless, the performance of SCTP and CMT-SCTP are virtually the same.

For the WLAN-3G scenario, we observe that multi-path transport leads to increased delays and delay variation. The main reason for this is the reordering caused by the heterogeneous path characteristics.

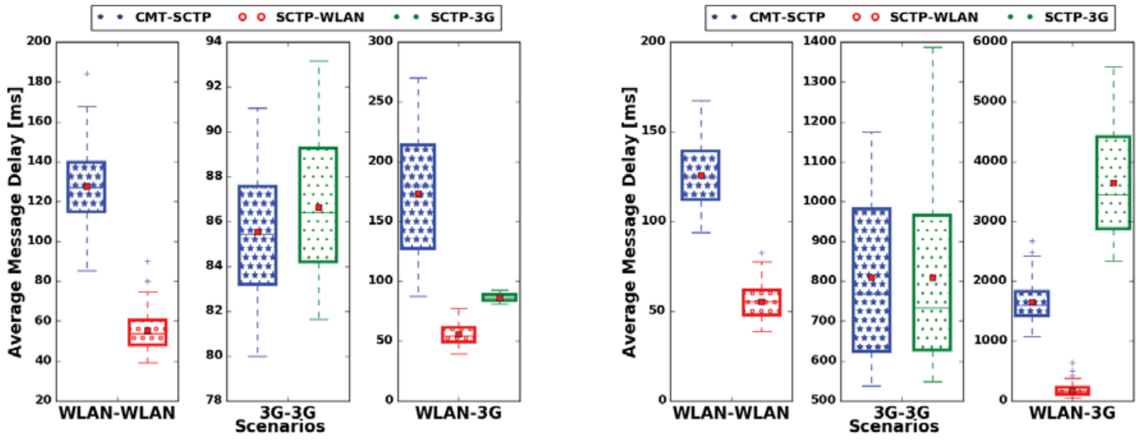
For CMT-SCTP, it is important to note that SCTP has its origins as transport protocol for signaling systems (see Section 2.1), where networks are designed for specific applications. Therefore, the most important implementations – the OMNeT++ simulation model as well as CMT-SCTP implementation in FreeBSD – currently only provide a very simple scheduler; data is scheduled on the paths in a round-robin fashion. The intention of using this scheduler is to improve throughput, without caring about path delays. That is, once data to send is available, and a path's congestion window allows to send it, as much as possible is sent on this path. Then, for further data to be sent, the next path is tried. This mechanism is tightly combined with the SCTP burst mitigation that limits the amount of consecutive packets sent at once over a path. Both SCTP implementations apply burst mitigation by using the "use it or lose it" [60] strategy, with a setting of MaxBurst=4 (default from [9, Section 15]). That is, if a certain number of bytes α is acknowledged by the receiver side the sender would be allowed to send up to α new bytes. The limit of in-flight bytes is given by the congestion window. However, if a non-saturated sender does not fully utilize its allowance given by the congestion window, the congestion window is reduced to the number of in-flight bytes *plus* MaxBurst*MSS. As a result, using MaxBurst=4, only up to 4 packets are sent on a path before the next path is used. The round-robin scheduler together with the burst mitigation results in an increase in the message delay for CMT especially when the paths are heterogeneous or when there exists loss.

Next, the performance of CMT-SCTP was evaluated in the presence of competing background traffic (see Section 4.4) and the corresponding average message delays are illustrated in Figure 6b.

In the WLAN-WLAN scenario, similar to the case without background traffic, multi-path transport leads to increased latency and latency variation due to the reordering caused by retransmissions. We also see that the background traffic has no visible impact in the WLAN/WLAN scenario due to the random packet loss on the path. As the background flows are experiencing loss, the TCP background flow backs off before it causes any noticeable congestion, and as the UDP background traffic is only 10% of the link capacity it also has very limited impact. Hence in the WLAN-WLAN scenario there was no noticeable effect of congestion losses or queuing delay on the foreground flows as compared to the corresponding scenario without background traffic.

For the 3G-3G scenario, also similar to non-competing traffic, we observe no significant performance difference for CMT-SCTP as compared to SCTP. Note that in this scenario, the send and receive buffer sizes are 256 KiB (see Section 4.5.3), and the background traffic leads to an increased delay due to bufferbloat. We observe an average delay of around 800 ms when background traffic exists as compared to 80-90 ms when there is no competing traffic.

The WLAN-3G results differ quite much from the non-background results. Note that for WLAN-3G case, we have different buffer settings: a send buffer of 1024 KiB and a receive buffer of 2048 KiB (see Section 4.5.3). These buffer settings allow the queue on the 3G path to grow, causing a significant bufferbloat. Here, we observe the delay on the 3G



(a) Without Background Traffic

(b) With Competing Background Traffic

Figure 6: Average Message Delay for CBR Video Traffic over CMT-SCTP

Traffic	Background	WLAN-WLAN		3G-3G		WLAN-3G	
		WLAN	WLAN	3G	3G	WLAN	3G
Video on CMT-SCTP	✗	17.5	82.5	6.3	93.7	16.4	83.6
	✓	17.5	82.5	4.5	95.5	79.2	20.8

Table 4: Path 1 Traffic Share (in %) for CBR Video Traffic over CMT-SCTP

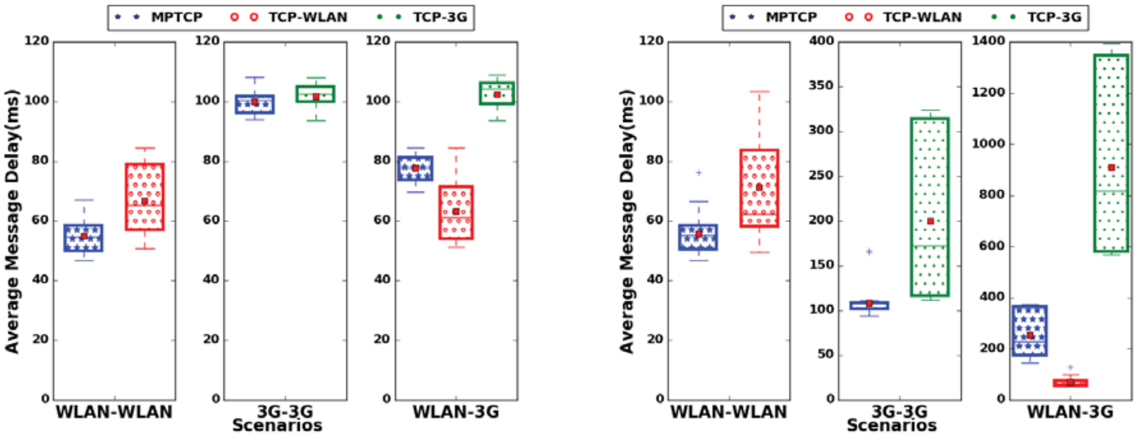
path jumping to values of almost 4 s, making any interactivity virtually impossible. Using CMT-SCTP leads to a significant reduction of the delay – due to the additional usage of the low-latency WLAN path – to values of around 1.8 s. However, this delay is still much higher compared to the delay of SCTP on the WLAN path.

5.1.2. MPTCP Emulation

Figure 7a presents the average message delay for video traffic in all the scenarios considered, i.e., WLAN-WLAN, 3G-3G, WLAN-3G. Each plot represents the data for 30 repetitions when no background traffic was present.

In the WLAN-WLAN scenario, the path delay difference between the two paths is small (i.e., 20ms – 25ms; see Table 1). Thus, the main factor that determines the average message delay is the link losses. Loss on one path causes the scheduler to push data on the other path and eventually exploiting the availability of multiple paths. If both paths have similar delay and loss as per the setup, data is sent on both paths causing traffic to oscillate between paths. Such oscillation of traffic between paths is known as flapping. In some of the repetitions, we observed that flapping and losses caused data to arrive out-of-order at the receiver, resulting in increased delays. However, on average, we observed that MPTCP improves the delay performance and reduces the delay variation as compared to TCP. This is in contrast to the relationship between CMT-SCTP and SCTP discussed above. The difference is due to the fact that MPTCP uses a lowest-RTT scheduler (see Section 4.5.3) that moves the sending of data between the paths in a better way as loss occurs. The limitation imposed by MaxBurst in CMT-SCTP also results in a less suitable distribution of the data as compared to MPTCP where the congestion window stays larger.

In the 3G-3G scenario, there are no losses and all the data is sent over only one path. Hence, there is no performance differences between MPTCP and TCP. Table 5 provides some insights on the share of data over each 3G path. The delay difference between the paths is small, but still enough to make the scheduler use only one of the paths. Certain configurations start with a non-optimal interface as default, and in those cases the scheduler eventually switches to the other path. This is evident in Ta-



(a) Without Background Traffic (b) With Competing Background Traffic

Figure 7: Average Message Delay for CBR Video Traffic over MPTCP in CORE Emulation

ble 5, where 0.16% of the data was sent on the other 3G path.

In the heterogeneous scenario (WLAN-3G), the WLAN path clearly has a lower average delay than the 3G path, even for small amounts of loss on the WLAN link. The behavior of the default scheduler ensures that MPTCP uses the path with lowest RTT. However, the performance of MPTCP was observed to be worse than that of TCP in this scenario. MPTCP uses both paths due to losses in the WLAN, triggering transmission over the 3G path which otherwise would not be used due to the large path delay differences. In the case of video traffic considered for the experiments, the data share shown in Table 5 should be identical to the packet share, due to the fixed size of the packets. The large amount of data transmitted on the 3G link provokes head-of-line blocking and the resulting application-level latency prevents MPTCP from reducing the latency.

To analyse the performance in the case of competing traffic, we considered experiments with background flows as specified in Section 4.4. The results are presented in Figure 7b. Similar to the CMT-SCTP case, the background traffic has negligible impact over the WLAN paths as the loss encountered by the background flows prevents congestion from forming. Hence there is no impact of background traffic on the WLAN-WLAN scenario.

In the 3G-3G scenario, there is an improvement in the performance of MPTCP which was not visible without background traffic (or for CMT-SCTP). MPTCP has a less varying average message delay than TCP in this scenario mainly due to the use of multiple paths and the lowest-RTT scheduling: since the distribution of the traffic considers the delay of each path, it is affected by the current congestion level of each path. The MPTCP scheduler used both paths and the data distribution among the paths is 85/15% as shown in Table 5. The path with the shorter base RTT may not always be the best path as the background traffic builds up queues in the network, leading to a somewhat more even share of the data between the paths as compared to the scenario without background traffic.

In the WLAN-3G scenario, MPTCP increases the delay compared to single path TCP over WLAN as the data was split between asymmetric paths. The underlying issue is the same as when there is no background traffic, but the background traffic causes the delays to get larger. As in the CMT-SCTP scenario, in Figure 7b, the delays for MPTCP and for TCP over 3G is higher than those of the 3G-3G scenario, due to the larger receive buffers in the WLAN-3G scenario.

5.1.3. *MPTCP Real Measurements*

We have run over 30 experiments in the NorNet Edge (NNE) testbed for the video traffic and illustrated the delay measurements in Figure 8. Note that different from WLAN links, in mobile broadband networks, each user has a dedicated channel. Therefore, we assume that the user is only streaming video without running any other bandwidth de-

Traffic	Background	WLAN-WLAN		3G-3G		WLAN-3G	
		WLAN	WLAN	3G	3G	WLAN	3G
Video	✗	54.19	45.81	99.84	0.16	51.96	48.04
	✓	55.67	44.33	85.24	14.76	51.98	48.02

Table 5: Video traffic data share per path using MPTCP

	WLAN-WLAN		3G-3G		WLAN-3G	
	WLAN	WLAN	3G	3G	WLAN	3G
Video	65.05	34.9	93.8	6.19	20.41	79.5

Table 6: Video traffic data share per path using MPTCP with NorNet

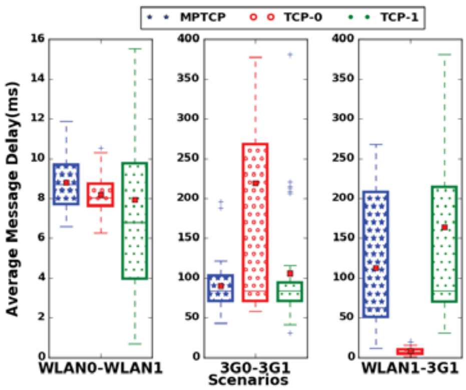


Figure 8: Average Message Delay for CBR Video Traffic over MPTCP in NorNet Experiment

manding applications.

For the homogeneous cases (e.g., WLAN-WLAN and 3G-3G), we observe that the paths can have quite different delay values although we are using the same technology. This results in delay differences between the paths, compared to the emulations, and impact the performance of MPTCP. For example, in the 3G-3G scenario, we observe that delay with MPTCP lies between the delay with TCP of the two 3G paths and MPTCP provide delay values much closer to the 3G path with lower delay with TCP. Similarly, for the WLAN-WLAN scenario, the delay with MPTCP is on average closer to the delay of the WLAN path with the lower TCP delay. We further observe large variations in the delay values among different experiment runs. This is in fact an expected result in real networks where the channel conditions can be very dynamic. These observations can further be verified by looking into the video traffic data share tabulated in Table 6.

For the heterogeneous scenario (WLAN-3G), we observe that the delays achieved by MPTCP is higher than the delay of TCP on the WLAN path, since MPTCP occasionally uses 3G path that has higher delay values as compared to WLAN. Although we have higher variations in the experimental results, this behavior is in general consistent with the emulation results.

5.2. *Gaming Traffic*

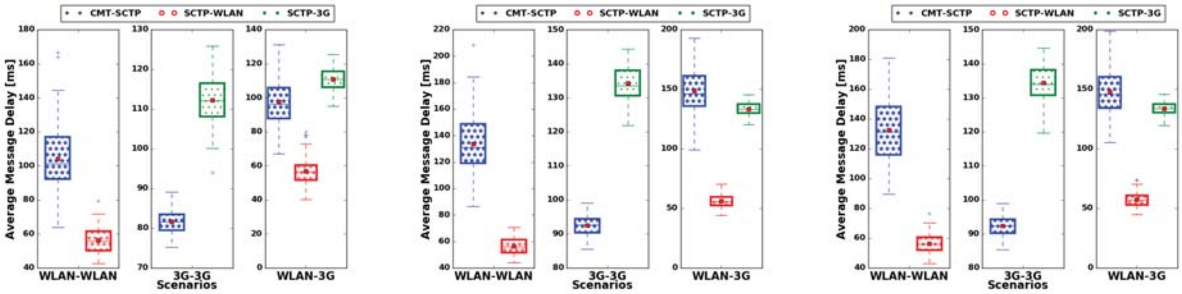
Gaming traffic is the second application traffic type that we consider. We start by presenting the simulation results for CMT-SCTP and continue with emulation and live experimentation results for MPTCP. We use the gaming traffic presented in Section 4.3.2. Same as for video traffic, we use application layer message delay as the performance metric.

5.2.1. *CMT-SCTP*

The average SCTP message delays over 128 runs are presented in Figure 9 (without background traffic) and Figure 10 (with background traffic according to Section 4.4) for the three gaming traces (Trace 1, Trace 2 and Trace 3) described in Section 4.3.2. The traffic share between the paths (i.e. the first path) is provided in Table 7.

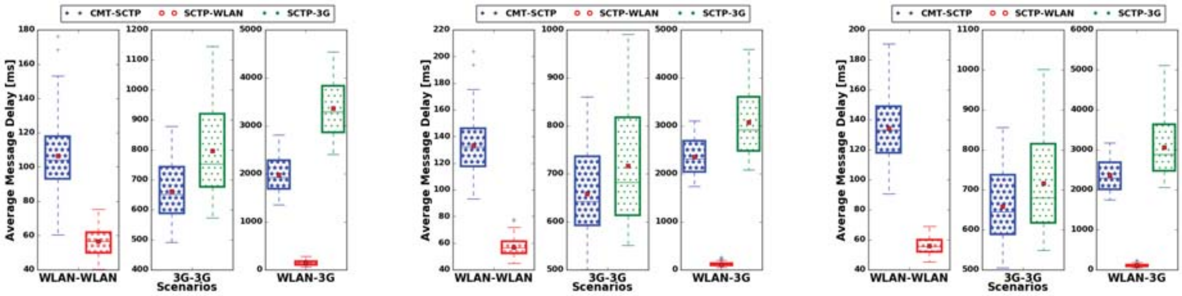
A particular property of the gaming traffic is its mix of traffic patterns due to the different phases of game play (see Section 4.3.2). The CMT-SCTP scheduler provides no gain for the small packets sent during the game. Therefore, when using symmetric WLAN paths in the WLAN-WLAN scenario, CMT-SCTP only leads to additional delay caused by reordering for the occasional small bursts of packets sent.

For the 3G-3G scenario, as explained previously for the video traffic, the small send and receive buffer settings of 256 KiB keeps bufferbloat and packet reordering small. Therefore, the effort for



(a) Trace 1 (b) Trace 2 (c) Trace 3

Figure 9: Average Message Delay for Gaming Traffic over CMT-SCTP (without Background Traffic)



(a) Trace 1 (b) Trace 2 (c) Trace 3

Figure 10: Average Message Delay for Gaming Traffic over CMT-SCTP (with Competing Background Traffic)

Trace	Background	WLAN-WLAN		3G-3G		WLAN-3G	
		WLAN	WLAN	3G	3G	WLAN	3G
Gaming T1	✗	24.1	75.9	30.9	69.1	33.7	66.3
	✓	24.4	75.6	32.3	67.7	57.9	42.1
Gaming T2	✗	36.8	63.2	37.5	62.5	47.4	52.6
	✓	36.9	63.1	23.9	76.1	63.9	36.1
Gaming T3	✗	36.4	63.6	37.3	62.7	48.4	51.6
	✓	36.5	63.5	22.9	77.1	65.0	35.0

Table 7: Path 1 Traffic Share (in %) for Gaming Traffic over CMT-SCTP

reordering messages remains small as well. However, due to the higher network latency, the burst mitigation handling leads to some performance gain with CMT-SCTP; the non-saturated sender does not fully utilize its allowance given by the congestion window. Therefore, the congestion window is reduced by the burst mitigation. This limitation keeps the congestion window small, allowing CMT-SCTP to send messages more quickly on the two independent paths when small bursts of packets need to be sent.

Again, as explained in Section 5.1 there is high bufferbloat on the 3G path in the WLAN-3G case, due to the send/receive buffer sizing of 1024 KiB/2048 KiB. This leads to a significant reordering for CMT-SCTP during the initial large burst of the game. Therefore, no significant performance improvement is achieved when no background traffic is present. However, in the scenario with background traffic, CMT-SCTP leads to some improvement due to the distribution of traffic over two paths. Nevertheless, the latencies caused by the bufferbloat of at least 2 s make any gaming interactivity impossible.

5.2.2. MPTCP Emulation

Figure 11 shows the delay values calculated over 30 runs for each gaming trace and the distribution of the data on each path is shown in Table 8.

In the WLAN-WLAN scenario, it is clear that one path is more used than the other, which was not the case for the video traffic. This is due to the limited amount of data to transmit for the many small packets. Losses on the WLAN have minimal impact on performance as there is less data to send on the other path in the event of loss during large portions of the game. In the 3G-3G scenario, the average delay using MPTCP is similar to that of TCP. We also observe a split of data over the paths, which was not the case for the video traffic.

Though the scenarios WLAN-WLAN and 3G-3G are symmetric in nature based on the characteristics, one of the interfaces is the best in any given configuration. The path delay settings are randomly drawn from a range of values (see Table 1). The maximum possible difference between two path delays are 5ms in WLAN-WLAN and 10ms in 3G-3G, respectively. This difference is sufficient for the MPTCP scheduler to estimate, adapt and change outgoing path for a packet. If the default interface is the best of the two available interfaces, then the flow uses mostly this interface. However, when the default interface is not the best of the two available interfaces, the MPTCP scheduler will send the first few packets on the default interface before settling with the other interface. Due to the initial large burst and the long-tail nature of the gaming traffic, the first few packets that were sent on a possibly sub-optimal interface represents a large chunk of the data share, although most packets are transferred on the other interface. This also results in very similar performance for MPTCP and TCP.

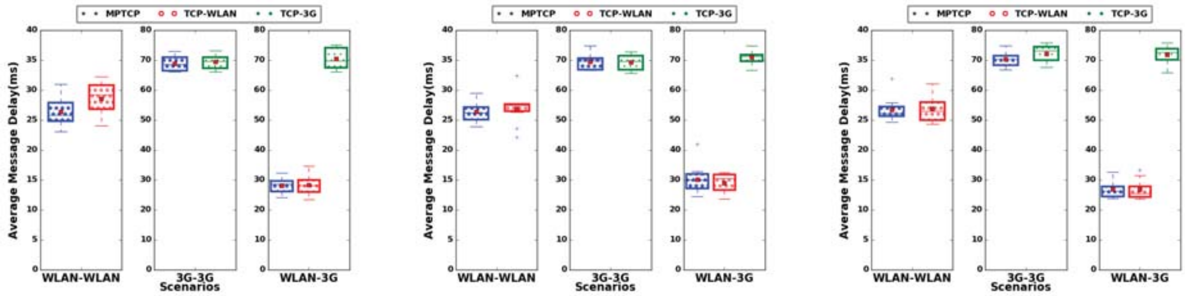
It is also worth noting the difference to CMT-SCTP here, which showed a clear gain in the 3G-3G scenario. The difference for CMT-SCTP is that all paths are available for transmission immediately and it was able to spread the initial burst of data over both paths to reduce the delay. As MPTCP sets up the second subpath in parallel with starting the data transfer, and also uses a larger initial congestion window, there was no gain during the initial burst of data from the game setup.

In the asymmetric WLAN-3G scenario, the average MPTCP delay values are similar to the TCP delay of the WLAN path. This is due to most of the data being sent over the default (better) path, which in this case is the WLAN.

Figure 12 presents the average delay of MPTCP versus TCP for each gaming trace with competing background traffic. The impact of background traffic for gaming is very similar to the video traffic case. In the 3G-3G scenario, there is a lower and less variable average message delay using MPTCP, due to the scheduling. For the WLAN-3G scenario, MPTCP increases the delay compared to single path TCP over WLAN as some of the data is sent over the 3G path. Still, the degradation is smaller and the improvement in relation to TCP over 3G is larger than in the video traffic scenario. This as a smaller fraction of data is sent on the 3G path in this scenario as discussed above. Again, background traffic has no impact in the WLAN-WLAN scenario.

5.2.3. MPTCP Real Measurements

We illustrate gaming traffic delay performance for real-world measurements in Figure 13 and the traffic distribution over the paths is presented in Table 9. For the homogeneous scenarios, we observe that the average TCP delay is very similar to the MPTCP delay. There are small variations among different traces, where in one trace MPTCP's delay is a little bit lower than the average TCP delay and in one trace it is a little bit higher. We observed



(a) Trace 1 (b) Trace 2 (c) Trace 3

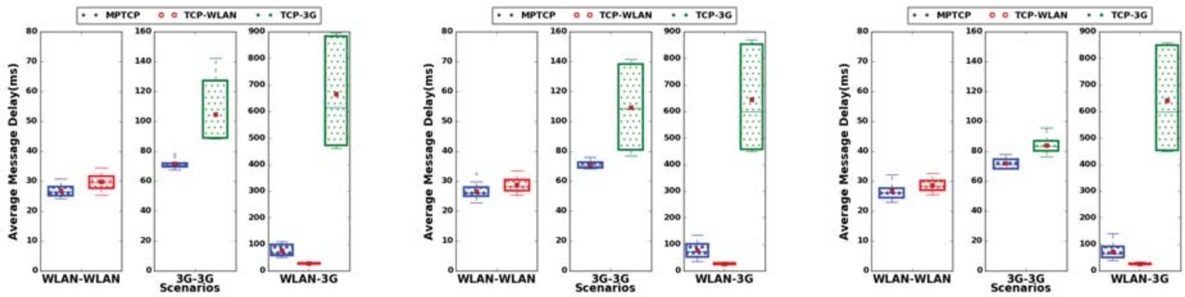
Figure 11: Average Message Delay for Gaming Traffic over MPTCP in CORE Emulation (without Background Traffic)

Traffic	Background	WLAN-WLAN		3G-3G		WLAN-3G	
		WLAN	WLAN	3G	3G	WLAN	3G
Gaming T1	✗	64.88	35.12	80.85	19.15	94.03	5.97
	✓	78.58	21.42	85.93	14.07	94.27	5.73
Gaming T2	✗	76.79	23.21	83.40	16.60	97.98	2.02
	✓	67.65	32.35	93.42	6.58	99.23	0.77
Gaming T3	✗	78.30	21.70	92.01	7.99	99.29	0.71
	✓	69.33	30.67	87.76	12.24	99.31	0.69

Table 8: Gaming traffic data share per path using MPTCP

Traffic	WLAN-WLAN		3G-3G		WLAN-3G	
	WLAN	WLAN	3G	3G	WLAN	3G
Gaming T1	77.67	22.32	0	100.0	40.74	59.25
Gaming T2	59.10	40.89	0	100.0	71.39	28.6
Gaming T3	54.57	45.42	0	100.0	30.31	69.69

Table 9: Gaming traffic data share per path using MPTCP in NorNet



(a) Trace 1 (b) Trace 2 (c) Trace 3

Figure 12: Average Message Delay for Gaming Traffic over MPTCP in CORE Emulation (with Competing Background Traffic)

that the packet shares are slightly more split towards the better path in the emulations compared to the experiments.

For WLAN-3G, we observe that MPTCP delay is slightly higher than the TCP WLAN delay. We observe that the packet share on the 3G path is higher in the experiments compared to the emulations, increasing the relative MPTCP delay slightly compared to the emulation results. Overall, the general trends seen between the protocols are still similar in the experiments and in the emulations.

5.3. Web Traffic

Finally, we evaluate the latency of web traffic for the homogeneous and heterogeneous scenarios, both with and without background traffic. For the web traffic, we chose the web site download time as the metric for transport protocol latency performance.

5.3.1. CMT-SCTP Simulations

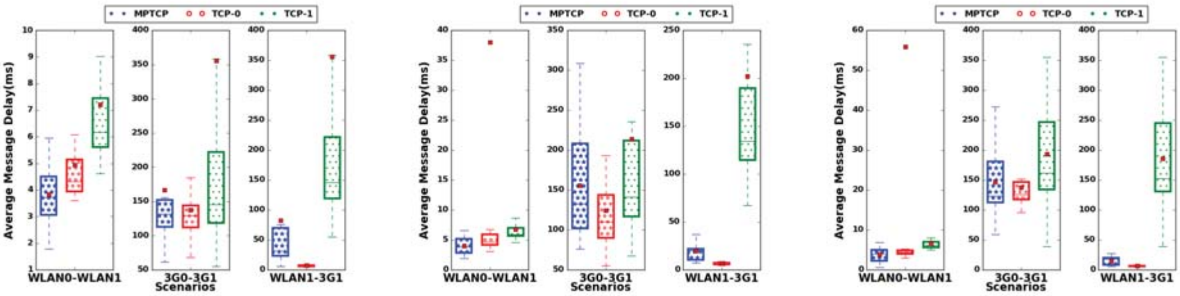
The web site download time results for the three web site scenarios are presented in Figure 14 (without background traffic; 256 runs) and Figure 15 (with background traffic according to Section 4.4; 1024 runs). Table 10 provides the corresponding traffic share for the two paths. Clearly, the benefit of CMT-SCTP usage increases as the web site size grows. The Wikipedia site (see Table 2), having only 72 KiB of payload data, is the smallest of the three sites. Therefore, the benefit of using CMT-SCTP for this web site is only small.

In the two scenarios with 3G path(s), a slight benefit can be seen: the 3G path has a small capacity and also a higher latency. Therefore, combining this 3G path with another 3G path, or even with a WLAN path, results in a faster download of the Wikipedia web site. As expected, for the Amazon (1 MiB) and the Huffington Post (3.9 MiB) web sites, CMT-SCTP reaches a significant download time reduction in most cases. However, for the WLAN-3G scenario with background traffic the path asymmetry is too large and CMT-SCTP performs worse than SCTP over WLAN for all web sites. Here the negative effect from head-of-line blocking dominates the gain from load balancing.

5.3.2. MPTCP Emulation

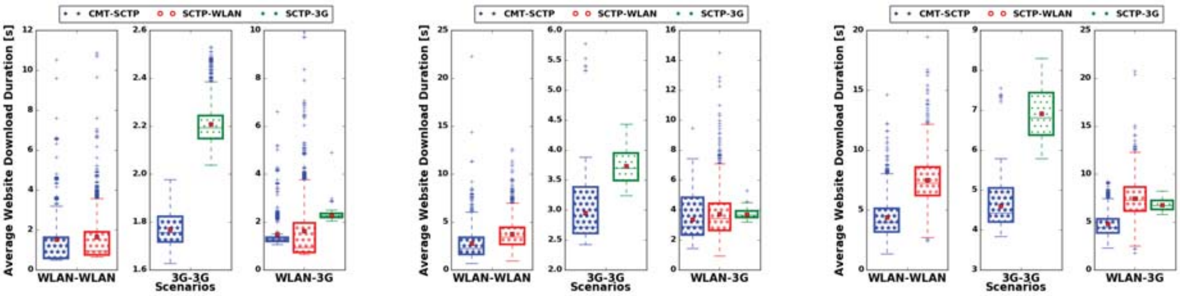
The average web site download times over 30 runs for the three web site scenarios, without background traffic, are presented in Figure 16. The corresponding results with background traffic are presented in Figure 17. Comparing the delay performance of MPTCP to that of TCP, MPTCP only provides limited improvements in download time for Wikipedia, but larger gains for both Amazon and Huffington Post (especially in 3G-3G scenarios). As also seen for CMT-SCTP, the results indicate that the size of the web site is critical to the total download time. With concurrent connections (6 in our setup), small web sites such as Wikipedia, can mostly be transferred within the initial window of TCP, not allowing MPTCP to exploit multiple paths.

With background traffic, the performance trends are similar to that of the non-background case. For the 3G paths, background traffic significantly increases the download time as well as the variation in download time. Background traffic has very little impact over the WLAN paths, but the random loss over the WLAN links still leads to a large variation in download times. The position of the loss in the short web flows can have a huge impact on the total web site download duration.



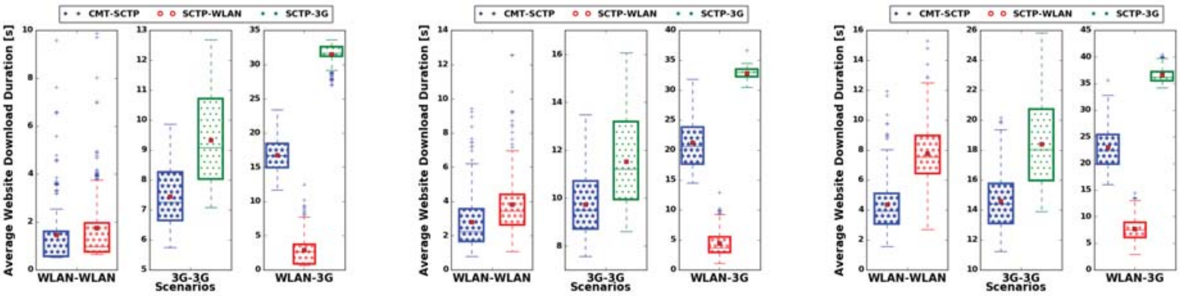
(a) Trace 1 (b) Trace 2 (c) Trace 3

Figure 13: Average Message Delay for Gaming Traffic over MPTCP in NorNet Experiment



(a) Wikipedia (www.wikipedia.org) (b) Amazon (www.amazon.com) (c) Huffpost (www.huffingtonpost.com)

Figure 14: Website Download Times over CMT-SCTP (without Background Traffic)

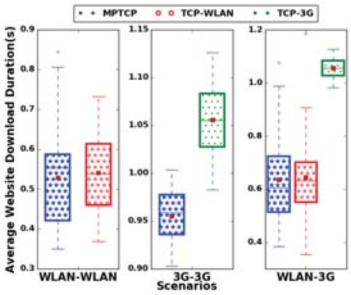


(a) Wikipedia (www.wikipedia.org) (b) Amazon (www.amazon.com) (c) Huffpost (www.huffingtonpost.com)

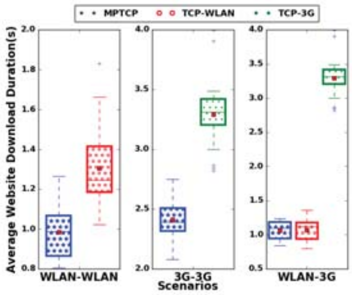
Figure 15: Website Download Times over CMT-SCTP (with Competing Background Traffic)

Website	Background	WLAN-WLAN		3G-3G		WLAN-3G	
		WLAN	WLAN	3G	3G	WLAN	3G
Wikipedia	✗	23.5	76.5	21.7	78.3	24.5	75.5
	✓	23.7	76.3	23.3	76.7	33.0	67.0
Amazon	✗	24.6	75.4	23.6	76.4	31.2	68.8
	✓	24.3	75.7	24.1	75.9	44.2	54.8
Huffington Post	✗	23.9	76.1	23.2	76.8	31.9	68.1
	✓	23.9	76.1	24.6	76.4	50.5	49.5

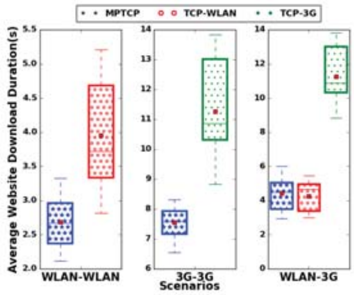
Table 10: Path 1 Traffic Share (in %) for Website Download over CMT-SCTP



(a) Wikipedia (www.wikipedia.org)

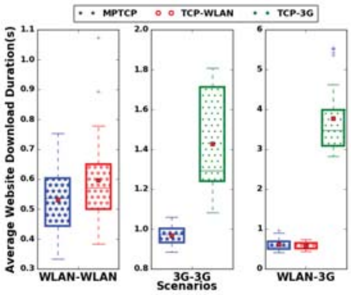


(b) Amazon (www.amazon.com)

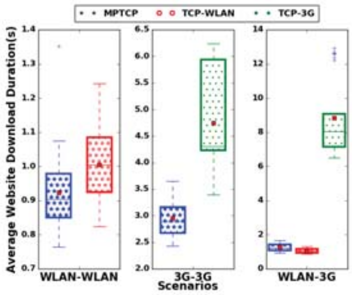


(c) Huffpost (www.huffingtonpost.com)

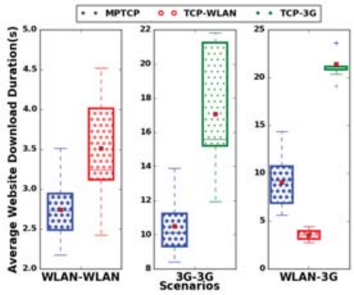
Figure 16: Website Download Times over MPTCP in CORE Emulation (without Background Traffic)



(a) Wikipedia (www.wikipedia.org)

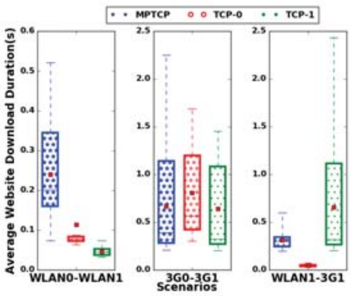


(b) Amazon (www.amazon.com)

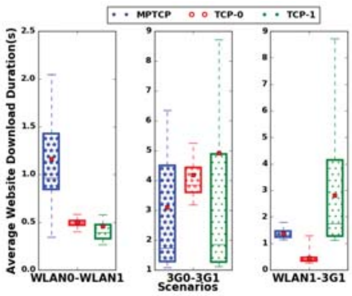


(c) Huffpost (www.huffingtonpost.com)

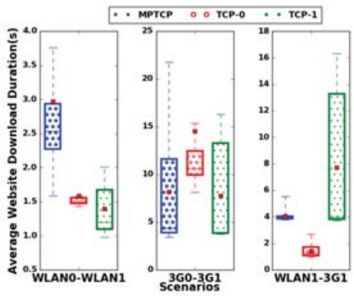
Figure 17: Website Download Times over MPTCP in CORE Emulation (with Competing Background Traffic)



(a) Wikipedia (www.wikipedia.org)



(b) Amazon (www.amazon.com)



(c) Huffpost (www.huffingtonpost.com)

Figure 18: Website Download Times over MPTCP in NorNet Experiment

Table 11 indicates that in symmetric scenarios, data transfer uses both paths. In WLAN-WLAN, random losses causes data to be sent over the second path even in small web sites like Wikipedia. The delay variance on the paths is the primary reason for use of the second path in 3G-3G scenarios. In asymmetric scenarios, most of the data uses the primary faster path WLAN.

5.3.3. MPTCP Real Measurements

In Figure 18, we illustrated the results of download times for web traffic. For WLAN-WLAN, we observe that the MPTCP delay is lower than the average TCP delay for Amazon whereas it is higher than the average TCP delay for the other two web sites: Wikipedia and Huffington Post. Similar observations can be made for the 3G-3G scenario. We observe that when there is enough data to be transmitted, MPTCP can provide benefits. For the WLAN-3G case, similar to other traffic, we observe that for all the web sites, MPTCP delay is a little bit higher than the WLAN delay.

The traffic distribution of data over the different paths is shown in Table 12. We observed that for the heterogeneous cases, almost all traffic is transferred over the WLAN path whereas for the homogeneous cases, the distribution depends on the size of the web site.

6. Discussion of Results

In this paper, we have run extensive measurements to evaluate the capability of multi-path transport protocols to carry latency sensitive application traffic. More specifically, we have analysed the application delays for video traffic, online gaming and web services both with and without competing traffic. Furthermore, we considered end-hosts experiencing multiple homogeneous paths as well as heterogeneous ones. The results are summarized in Figure 19, where the performance of multi-path, as compared to single path, is categorized into four types. We next elaborate on this table and recap our findings from Section 5.

The currently used round-robin scheduler in CMT-SCTP is optimised for throughput, not for low latency. Therefore, when there is a significant delay difference between the two paths, we observe performance degradation with CMT-SCTP compared to SCTP. However, for homogeneous paths, especially when the paths are not very lossy, we observe that CMT-SCTP can significantly reduce latency, especially in the web scenario. For video traffic CMT-SCTP provides similar or higher delay values as compared to SCTP. For example, in WLAN-WLAN scenarios, reordering due to distribution over multiple lossy paths increases the delay both with and without background traffic. On the other hand, in 3G-3G scenarios, we observe similar delay values for CMT-SCTP and SCTP. For the WLAN-3G scenario, packet reordering causes increased delay for CMT-SCTP even when there is no competing traffic. For gaming traffic, while CMT-SCTP leads to a latency increase in case of two similar, low-latency WLAN paths, it becomes beneficial in case of high-latency paths with background traffic; in comparison of using only the higher-delay path, CMT-SCTP is able to take advantage of the lower-delay path to reduce latency. However, as observed for the video traffic, its scheduler is optimised for throughput maximization without taking care of path delay. Therefore, while the latency is lower than using only the high-delay path, it is still much higher than using only the low-delay path in the heterogeneous WLAN-3G case. For web traffic, we observe that using CMT-SCTP improves the web site download speed, especially for homogeneous paths. Since the web traffic is saturated (i.e., send as much data as possible), the round-robin scheduler that is used by the CMT-SCTP implementation performs reasonably well by ensuring that both paths are utilized, although it does not always chose the path with the lowest RTT. The larger the web site, the better the performance improvement achieved by CMT-SCTP. However, for the WLAN-3G case, especially when there is background traffic, CMT-SCTP cannot handle the delay difference between the 3G and WLAN paths, resulting in poor performance for CMT-SCTP compared to SCTP.

For MPTCP, the default scheduler is based on delay and it has been shown to achieve low and stable latency [69]. For different type of traffics, we observe similar or lower delay values for MPTCP compared to TCP, for homogeneous paths. However, the main factor that determines the delay performance of MPTCP is indeed the path heterogeneity, and for heterogeneous paths we observe performance degradation whose degree depends on the traffic and whether there exists background traffic. More specifically, for video traffic, when the links are lossy, e.g. the WLAN-WLAN case, we observe delay gains due to link aggregation. For the 3G-3G

	Background	WLAN-WLAN		3G-3G		WLAN-3G	
		WLAN	WLAN	3G	3G	WLAN	3G
Wikipedia	✗	90.22	9.78	86.98	13.02	99.94	0.06
	✓	86.98	13.02	83.86	16.14	99.94	0.06
Amazon	✗	76.02	23.98	56.73	43.27	99.81	0.19
	✓	70.45	29.55	72.64	27.36	99.9	0.1
Huffington Post	✗	85.3	14.7	68.1	31.9	99.9	0.1
	✓	83.0	17	79.13	20.87	98.86	1.14

Table 11: Web traffic data share per path using MPTCP

	WLAN-WLAN		3G-3G		WLAN-3G	
	WLAN	WLAN	3G	3G	WLAN	3G
Wikipedia	94.3	5.6	100.0	0.0	100.0	0.0
Amazon	26.1	73.8	96.9	3.1	100.0	0.0
Huffington Post	14.5	85.4	100.0	0.0	100.0	0.0

Table 12: Web traffic data share per path using MPTCP in NorNet

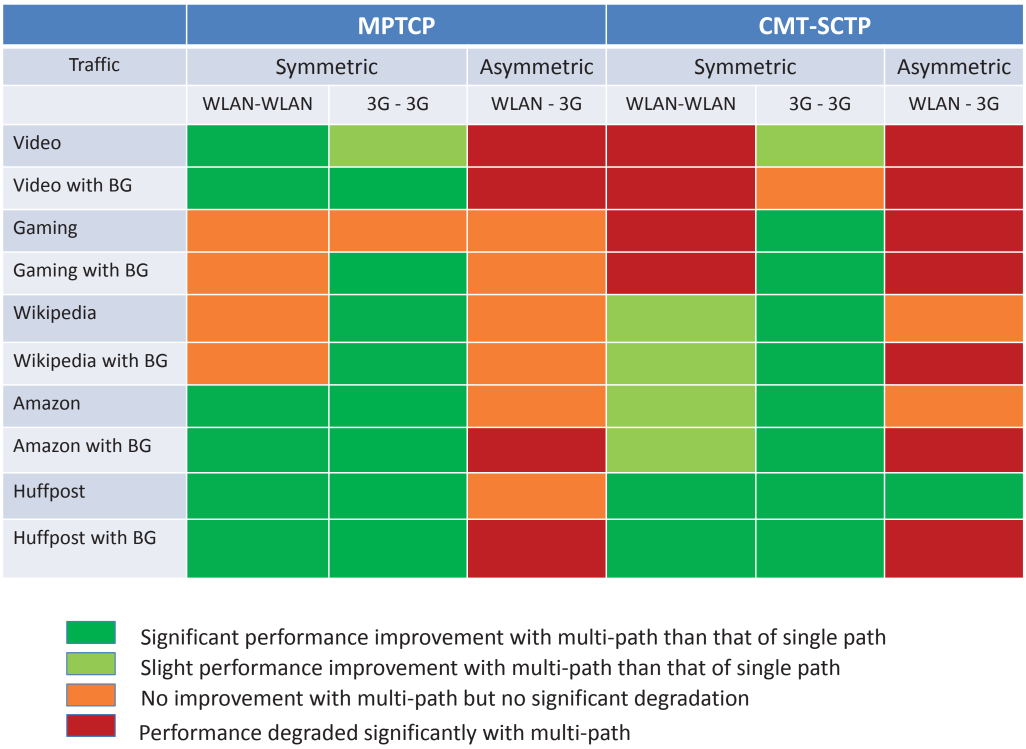


Figure 19: Multi-path versus single path transport protocols depending on the latency sensitive traffic: Summary table

case, where there are no losses, MPTCP selects the best available path resulting in minor gains compared to TCP. In the presence of background traffic, the delay values are in general higher but the benefits of using MPTCP are consistent with that of the non-competing traffic. We further observe that these emulation results are mostly consistent with the NorNet experiments. Here, the main difference is that in real networks, paths have more diverse characteristics, although using the same technology. This results in slight heterogeneity and the delay using MPTCP becomes higher than using TCP over the best path only, but still lower than the average TCP delay. For gaming traffic, we observe very similar delay values to TCP for almost all cases due to the very limited amount of data. The background traffic did not induce enough loss in the foreground flows, therefore the delay values are similar to that of no background traffic. One exception is the 3G-3G scenario where MPTCP keeps sending over one path as long as there is no loss, therefore, providing some delay gains. For the WLAN-3G case, MPTCP uses the WLAN at almost all times, therefore, the MPTCP delay is similar to the TCP delay of WLAN. Similarly, for the results of the real experiments, we observe similar behavior to the emulations, especially for the homogeneous scenarios, with slight variations among different trace files. For the heterogeneous scenario, there is slightly higher delay in real experiments compared to emulations, due to some traffic is being transferred over the slower 3G path. For the web traffic, we observe that MPTCP provides lower delay values, especially for web sites with many objects. The lower delay is a consequence of MPTCP's scheduler which always tries to use the path with the lowest RTT. However, when the paths are very heterogeneous in terms of delay and loss as in the WLAN-3G case, losses in WLAN forces MPTCP to use the suboptimal 3G path; therefore, the MPTCP delay becomes higher than the TCP delay of WLAN. These results hold for emulations with and without background traffic. Similar to the previous applications, the results of the real experiments are mostly consistent with the emulation results. Due to the differences in the paths for the homogeneous cases (e.g. 3G-3G and WLAN-WLAN), MPTCP delay is higher than the best path while still much lower than the average TCP delay.

One conclusion of our study is that multi-path transport protocols can hardly reduce the latency for all the tested applications, when there is some asymmetry between the paths. Moreover, in this case, multi-path transport may increase the latency, mainly because of head-of-line blocking. However, it is worth pointing out that in most symmetric scenarios, multi-path transport protocols enable a significant latency reduction.

7. Related Work

This section discusses work related to ours. While there are numerous articles on MPTCP and its performance in relation to TCP, not much has been written on the relation between CMT-SCTP and SCTP. Instead, most articles on CMT-SCTP propose various optimizations to the protocol itself. There are, of course exceptions; Aydin et al. [70] elaborates the importance of TCP friendliness for single homed SCTP and evaluates the TCP friendliness of CMT-SCTP. Arianpoo et al. [71] propose an adaptive network coding mechanism for CMT-SCTP to desensitize the receiver against packet reordering and consequently eliminate the receiver buffer blocking problem. They claim to have improve the CMT-SCTP performance by 62% over the original implementation in cases of severe path asymmetry.

For MPTCP, a closely related research work is [72], which measures MPTCP performance with the aim to understand the benefit of using two interfaces with MPTCP over using either one of the interface with TCP. This study also focuses on the impact that flow size has on the average latency, and provides insights into the effect of path characteristic diversity on application level performance. Their conclusions are consistent with ours: using multi-path becomes more and more beneficial when the size of the data to transmit increases. We extend their work by considering more application scenarios. In [73], S. Deng et al., studies the performance of MPTCP over wireless technologies using Android application traffic. Their study focuses on energy efficiency and provides new challenges such as dynamic decision making at the mobile applications to select appropriate network technology depending on the flow size and traffic pattern. Handover performance was seen as a potential MPTCP performance impairment especially when the path characteristics are different. Andrei et al. [74] provided a simultaneous association solution using MPTCP for WLAN that avoids fast handover. It also provides possible modifications at the client side implementation, to mitigate the throughput loss in cases where the WLAN characteristics differ due to channel specification. Such approach of reducing the occurrence of handovers is a necessary improvement: it is essential to improve performance where multi-path transport increases the latency, and that has been identified in this paper.

Grinnemo et al. [75] provides a first comprehensive evaluation of MPTCP performance with latency as the quality of experience metric for cloud-based applications. They study three different applications: Netflix, Google Maps and Google Docs, representing high, mid and low intensity cloud-based traffic. The authors conclude that MPTCP provides significant performance gains for high and mid intensity traffic. Furthermore, it is noted that the variation in RTTs among network paths causes higher application latency, and the current Linux standard scheduler is seen as the primary cause of increased latency in such cases.

Raiciu et al. [76] proposed a mobility architecture to allow MPTCP to switch between different technologies and handle mobility at the transport layer instead of at the network layer. The mobility of MPTCP was evaluated with simulations and indoor mobility experiments. The criteria for the evaluation was measured throughput on TCP and MPTCP using WLAN-3G, and power efficiency of both protocols. The study concludes that MPTCP provides performance improvements over TCP when multiple interfaces are used in parallel. Power efficiency of MPTCP depends on the underlying interface power consumption and should be tuned for better performance. Later, the power efficiency of MPTCP drew much attention in [77], which analyses the energy consumption and handover performance of MPTCP in the different operational modes: Full MPTCP Mode, Backup Mode and Single path Mode. This work again provides experimental evaluations using the Linux implementation of MPTCP and commercial access networks providing 3G and broadband access on static nodes. The study concludes that MPTCP handovers might have small impact on application delay and goodput in different operational modes.

With a few exceptions discussed above, most of the prior research on CMT-SCTP or MPTCP measurements focused on the performance of the protocol in terms of throughput, energy consumption, handover performance and RTTs. To the best of our knowledge, this paper is the first to provide a comprehensive analysis of multi-path transport performance with latency as the main metric.

As seen throughout this paper, the performance of multi-path transport is highly dependent on doing efficient scheduling. Paasch et al. [27] provide a detailed study of schedulers and their impact on performance. The authors implement a generic modular framework for evaluating MPTCP schedulers in Linux. Using this framework, different schedulers are then evaluated using various performance metrics and different types of traffic, including bulk and application limited traffic.

8. Conclusions and Future Work

For an increasing number of applications, latency plays an important role as it directly impacts their performance. Still, most work considering multi-path communication is solely focused on resilience and throughput maximization. The work presented in this paper tries to bridge this gap by evaluating whether multi-path communication can help latency-sensitive applications satisfy their users' requirements. Three latency-sensitive applications have been considered: video, gaming and web traffic. Performance have been evaluated using 3G–3G, 3G–WLAN, and WLAN–WLAN paths, in both simulated, emulated and real-life environments considering both CMT-SCTP and MPTCP.

The results indicate that multi-path communication can reduce latency significantly, but only when paths are symmetric in terms of delay and loss rate. The potential gain comes mainly from two factors: the possibility to distribute short bursts of data over multiple interfaces and the ability to select the best of the available paths for data transmission. In asymmetric scenarios where the latency reduction is not as significant (or non-existent), applications may still benefit from other properties of multi-path communication, without increasing latency. This is, however, highly dependent on the scheduling mechanism used. As seen in some of the CMT-SCTP experiments, a scheduler designed mainly for throughput maximization, may lead to increased latency in some scenarios. Considering the importance of scheduling, this is where we direct our attention for future work, and we are currently designing a scheduler targeting latency-sensitive traffic.

Acknowledgments

The authors are funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700). The views expressed are solely those of the authors.

References

References

[1] S. S. Krishnan, R. K. Sitaraman, Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-experimental Designs, in: Proceedings of the 12th ACM Internet Measurement Conference (IMC), ACM, New York/U.S.A., 2012, pp. 211–224.

[2] P. Quax, P. Monsieurs, W. Lamotte, D. D. Vleeschauwer, N. Degrande, Objective and Subjective Evaluation of the Influence of Small Amounts of Delay and Jitter on a Recent First Person Shooter Game, in: Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games (NetGames), ACM, Portland, Oregon/U.S.A., 2004, pp. 152–156, ISBN 1-58113-942-X.

[3] J. Liu, Y. Tan, F. Fu, T. Dreibholz, X. Zhou, Y. Bai, X. Yang, W. Du, Study on MPTCP and CMT-SCTP Congestion Control Mechanism, Computer Engineering 41 (4) (2015) 117–124.

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

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

[6] J. R. Iyengar, P. D. Amer, R. R. Stewart, Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-End Paths, IEEE/ACM Transactions on Networking 14 (5) (2006) 951–964, ISSN 1063-6692.

[7] P. D. Amer, M. Becke, T. Dreibholz, N. Ekiz, J. R. Iyengar, P. Natarajan, R. R. Stewart, M. Tüxen, Load Sharing for the Stream Control Transmission Protocol (SCTP), Internet Draft draft-tuexen-tsvwg-sctp-multipath-11, IETF, Individual Submission (Nov. 2015).

[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). URL https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-29737/Dre2012_final.pdf

[9] R. R. Stewart, Stream Control Transmission Protocol, RFC 4960, IETF, ISSN 2070-1721 (Sep. 2007).

[10] T. Dreibholz, I. Rüngeler, R. Seggelmann, M. Tüxen, E. P. Rathgeb, R. R. Stewart, Stream Control Transmission Protocol: Past, Current, and Future Standardization Activities, IEEE Communications Magazine 49 (4) (2011) 82–88, ISSN 0163-6804.

[11] IETF SIGTRAN WG, IETF Signaling Transport Working Group (2012). URL https://datatracker.ietf.org/wg/sigtran/charter/

[12] ITU-T, Introduction to CCITT Signalling System No. 7, Recommendation Q.700, International Telecommunication Union (May 1994).

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

[14] J. B. Postel, User Datagram Protocol, Standards Track RFC 768, IETF, ISSN 2070-1721 (Aug. 1980).

[15] R. R. Stewart, Q. Xie, M. Tüxen, S. Maruyama, M. Kozuka, Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, Standards Track RFC 5061, IETF, ISSN 2070-1721 (Sep. 2007).

[16] L. Budzisz, J. Garcia, A. Brunström, R. Ferrús, A Taxonomy and Survey of SCTP Research, ACM Computing Surveys (CSUR) 44 (4) (2012) 18:1–18:36.

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

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

[19] IETF MPTCP WG, IETF Multipath TCP Working Group (2012). URL https://datatracker.ietf.org/wg/mptcp/charter/

[20] M. Becke, T. Dreibholz, H. Adhari, 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, 2012, pp. 2666–2672, ISBN 978-1-4577-2052-9.

[21] C. Raiciu, M. Handley, D. Wischik, Coupled Congestion Control for Multipath Transport Protocols, RFC 6356, IETF, ISSN 2070-1721 (Oct. 2011).

[22] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, M. Handley, Improving Datacenter Performance and Robustness with Multipath TCP, in: Proceedings of the ACM SIGCOMM Conference, Toronto, Ontario/Canada, 2011, pp. 266–277, ISBN 978-1-4503-0797-0.

[23] L. Boccassi, M. Fayed, M. 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, FL, USA, 2013, pp. 3–8.

[24] H. Adhari, T. Dreibholz, M. Becke, E. P. Rathgeb, M. Tüxen, Evaluation of Concurrent Multipath Transfer over Dissimilar Paths, in: Proceedings of the 1st International Workshop on Protocols and Applications with Multi-Homing Support (PAMS), Singapore, 2011, pp. 708–714, ISBN 978-0-7695-4338-3.

[25] T. Dreibholz, M. Becke, E. P. Rathgeb, 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., 2010, pp. 1–6, ISBN 978-1-4244-5637-6.

[26] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, R. Boreli, DAPS: Intelligent Delay-Aware Packet Scheduling For Multipath Transport, in: Proceedings of the IEEE International Conference on Communications (ICC), Sydney, New South Wales/Australia, 2014, pp. 1222–1227.

[27] C. Paasch, S. Ferlin, Özgü Alay, O. Bonaventure, Experimental Evaluation of Multipath TCP Schedulers, in: Proceedings of the ACM SIGCOMM Capacity Sharing Workshop (CSWS), Chicago, Illinois/U.S.A., 2014, pp. 27–32, ISBN 978-1-4503-2991-0.

[28] C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchêne, O. Bonaventure, 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., 2012, pp. 1–14.

[29] T. Dreibholz, M. Becke, J. Pulinthanath, E. P. Rathgeb, Applying TCP-Friendly Congestion Control to Concurrent Multipath Transfer, in: Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), Perth, Western Australia/Australia, 2010, pp. 312–319, ISBN 978-0-7695-4018-4.

[30] T. Dreibholz, M. Becke, H. Adhari, 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, 2011, pp. 397–404, ISBN 978-953-184-152-8.

[31] R. Khalili, N. Gast, M. Popović, J.-Y. L. Boudec, MPTCP is not Pareto-Optimal: Performance Issues and a Possible Solution, IEEE/ACM Transactions on Networking 21 (5) (2013) 1651–1665, ISSN 1063-6692.

[32] A. Walid, Q. Peng, J. Hwang, S. H. Low, Balanced Linked Adaptation Congestion Control Algorithm for MPTCP, Internet Draft draft-walid-mptcp-congestion-control-02, IETF, Individual Submission (Jan. 2015).

[33] J. R. Iyengar, P. D. Amer, R. R. Stewart, Retransmission policies for Concurrent Multipath Transfer using SCTP Multihoming, in: Proceedings of the 12th IEEE International Conference on Networks (ICON), Vol. 2, 2004, pp. 713–719.

[34] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, F. Jahanian, M. Karir, Atlas Internet Observatory 2009 Annual Report, in: 47th NANOG, 2009.

[35] D. Ciullo, M. Mellia, M. Meo, Two Schemes to Reduce Latency in Short Lived TCP Flows, IEEE Communications Letters 13 (10) (2009) 806–808.

[36] Sandvine Intelligent Broadband Networks, Global Internet Phenomena Report (Jul. 2013).
URL https://web.archive.org/web/20141216103806/https://www.sandvine.com/downloads/general/global-internet-phenomena/2013/sandvine-global-internet-phenomena-report-1h-2013.pdf

[37] Cisco Visual Networking Index, Forecast and Methodology, 2012–2017 (2013).
URL https://web.archive.org/web/20141216103933/http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[38] Sandvine Intelligent Broadband Networks, Global Internet Phenomena Report (Sep. 2015).
URL http://web.archive.org/web/20160113020433/https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/global-internet-phenomena-report-apac-and-europe.pdf

[39] X. Zhang, Y. Xu, H. Hu, Y. Liu, Z. Guo, Y. Wang, Profiling Skype Video Calls: Rate Control and Video Quality, in: Proceedings of the IEEE INFOCOM, Orlando, Florida/U.S.A., 2012, pp. 621–629.

[40] L. D. Cicco, S. Mascolo, V. Palmisano, Skype Video Congestion Control: An Experimental Investigation, Computer Networks 55 (3) (2011) 558–571.

[41] T. Szigeti, C. Hattingh, End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs Quality of Service Design Overview, Cisco Press, 2004, ISBN 978-1587051760.

[42] M. Claypool, K. Claypool, Latency and Player Actions in Online Games, Communications of the ACM 49 (11) (2006) 40–45.

[43] X. Che, B. Ip, Review: Packet-level Traffic Analysis of Online Games from the Genre Characteristics Perspective, Journal of Network and Computer Applications 35 (1) (2012) 240–252.

[44] N. Sheldon, E. Girard, S. Borg, M. Claypool, E. Agu, The effect of latency on user performance in Warcraft III, in: Proceedings of the 2nd Workshop on Network and System Support for Games (NetGames), Redwood City, California/U.S.A., 2003, pp. 3–14.

[45] K.-T. Chen, P. Huang, G.-S. Wang, C.-Y. Huang, C.-L. Lei, On the Sensitivity of Online Game Playing Time to Network QoS, in: Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM), Barcelona, Catalonia/Spain, 2006, pp. 1–12, ISSN 0743-166X.

[46] O3b Networks and Sofrecom, Why Latency Matters to Mobile Backhaul (Apr. 2013).
URL https://web.archive.org/web/20141216102926/http://www.o3bnetworks.com/media/45606/o3b_latency_mobile%20backhaul_130417.pdf

[47] Google Developers, Web Metrics: Size and Number of Resources (May 2010).
URL https://web.archive.org/web/20140625155405/https://developers.google.com/speed/articles/web-metrics

[48] A. Varga, OMNeT++ Discrete Event Simulation System User Manual – Version 4.6 (Dec. 2014).
URL https://omnetpp.org/doc/omnetpp/Manual.pdf

[49] T. Dreibholz, M. Becke, J. Pulinthanath, E. P. Rathgeb, Implementation and Evaluation of Concurrent Multipath Transfer for SCTP in the INET Framework, in: Proceedings of the 3rd ACM/ICST International Workshop on OMNeT++, Torremolinos, Málaga/Spain, 2010, pp. 15–22, ISBN 978-963-9799-87-5.

[50] I. Rüngeler, SCTP – Evaluating, Improving and Extending the Protocol for Broader Deployment, Ph.D. thesis, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems (Dec. 2009).
URL https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-36393/Diss.pdf

[51] T. Dreibholz, M. Becke, H. Adhari, 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, 2011, pp. 1–6, ISBN 978-953-290-027-9.

[52] A. Varga, INET Framework for OMNeT++ (2015).
URL https://inet.omnetpp.org/

[53] T. Dreibholz, X. Zhou, E. P. Rathgeb, SimProcTC – The Design and Realization of a Powerful Tool-Chain for OMNeT++ Simulations, in: Proceedings of the 2nd ACM/ICST International Workshop on OMNeT++, Rome/Italy, 2009, pp. 1–8, ISBN 978-963-9799-45-5.

[54] T. Dreibholz, E. P. Rathgeb, A Powerful Tool-Chain for Setup, Distributed Processing, Analysis and Debugging of OMNeT++ Simulations, in: Proceedings of the 1st ACM/ICST International Workshop on OMNeT++,

Marseille, Bouches-du-Rhône/France, 2008, pp. 74–81, ISBN 978-963-9799-20-2.

[55] K. V. Jónsson, HttpTools: A Toolkit for Simulation of Web Hosts in OMNeT++, in: Proceedings of the 2nd ACM/ICST International Workshop on OMNeT++, Rome/Italy, 2009, pp. 1–8, ISBN 978-963-9799-45-5.

[56] J. Ahrenholz, Comparison of CORE Network Emulation Platforms, in: Military Communications Conference (MILCOM), San Jose, California/U.S.A., 2010, pp. 166–171.

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

[58] T. Dreibholz, S. Ferlin, O. Alay, N. Kuhn, K. Yedugundla, P. Hurtig, A. Brunstrom, Is Multi-path Transport Suitable for Latency Sensitive Traffic (Sep. 2015).
URL https://git.cs.kau.se/rite/multipath-latency

[59] J. C. Mogul, G. Minshall, Rethinking the tcp nagle algorithm, SIGCOMM Comput. Commun. Rev. 31 (1) (2001) 6–20.

[60] M. Allman, E. Blanton, Notes on Burst Mitigation for Transport Protocols, ACM SIGCOMM Computer Communication Review (CCR) 35 (2) (2005) 53–60, ISSN 0146-4833.

[61] Funcom, Age of Conan (Jul. 2015).
URL https://web.archive.org/web/20150728133142/http://www.ageofconan.com/

[62] A. Botta, A. Dainotti, A. Pescapé, A Tool for the Generation of Realistic Network Workload for Emerging Networking Scenarios, Computer Networks 56 (15) (2012) 3531–3547.

[63] S. Ha, L. Le, I. Rhee, L. Xu, Impact of Background Traffic on Performance of High-speed TCP Variant Protocols, Comput. Netw. 51 (7) (2007) 1748–1762.

[64] T. Dreibholz, NetPerfMeter Homepage (Jul. 2015).
URL http://web.archive.org/web/20150728133903/https://www.uni-due.de/~be0001/netperfmeter/

[65] S. Hassayoun, J. R. Iyengar, D. Ros, Dynamic Window Coupling for Multipath Congestion Control, in: Proceedings of the 19th IEEE International Conference on Network Protocols (ICNP), 2011, pp. 341–352.

[66] V. Cerf, V. Jacobson, N. Weaver, J. Gettys, BufferBloat: What's Wrong with the Internet?, ACM Queue 9 (12) (2011) 10–20, ISSN 1542-7730.

[67] J. Gettys, Bufferbloat – Dark Buffers in the Internet (Jan. 2011).
URL https://www.bufferbloat.net/attachments/9/BufferBloat11.pdf

[68] D. F. Williamson, R. A. Parker, J. S. Kendrick, The box plot: A simple visual method to interpret data, Annals of Internal Medicine 110 (11) (1989) 916–921.

[69] C. Paasch, S. Ferlin, Özgü Alay, O. Bonaventure, Experimental Evaluation of Multipath TCP Schedulers, in: Proceedings of the ACM SIGCOMM Workshop on Capacity Sharing Workshop (CSWS), ACM, New York/U.S.A., 2014, pp. 27–32, ISBN 978-1-4503-2991-0.

[70] I. Aydin, J. Iyengar, P. Conrad, C.-C. Shen, P. Amer, Evaluating tcp-friendliness in light of concurrent multipath transfer, Computer Networks 56 (7) (2012) 1876 – 1892.

[71] N. Arianpoo, I. Aydin, V. C. M. Leung, Network coding: A remedy for receiver buffer blocking in the concurrent multipath transfer of data over multi-hop wireless networks, in: Communications (ICC), 2014 IEEE International Conference on, 2014, pp. 3258–3263.

[72] Y.-C. Chen, Y. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, D. Towsley, A Measurement-Based Study of MultiPath TCP Performance over Wireless Networks, in: Proceedings of the 13th ACM Internet Measurement Conference (IMC), Barcelona/Spain, 2013, pp. 455–468, ISBN 978-1-4503-1953-9.

[73] S. Deng, R. Netravali, A. Sivaraman, H. Balakrishnan, WiFi, LTE, or Both?: Measuring Multi-Homed Wireless Internet Performance, in: Proceedings of the 14th ACM Internet Measurement Conference (IMC), ACM, Vancouver, British Columbia/Canada, 2014, pp. 181–194, ISBN 978-1-4503-3213-2.

[74] A. Croitoru, D. Niculescu, C. Raiciu, Towards Wifi Mobility without Fast Handover, in: Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI), USENIX Association, Oakland, CA, 2015, pp. 219–234.

[75] K. J. Grinnemo, A. Brunstrom, A first study on using mptcp to reduce latency for cloud based mobile applications, in: 2015 IEEE Symposium on Computers and Communication (ISCC), 2015, pp. 64–69.

[76] C. Raiciu, D. Niculescu, M. B. Braun, M. J. Handley, Opportunistic Mobility with Multipath TCP, in: Proceedings of the Sixth International Workshop on MobiArch, MobiArch '11, ACM, Bethesda, Maryland/U.S.A., 2011, pp. 7–12, ISBN 978-1-4503-0740-6.

[77] C. Paasch, G. Detal, F. Duchêne, C. Raiciu, O. Bonaventure, Exploring Mobile/WiFi Handover with Multipath TCP, in: Proceedings of the ACM SIGCOMM Workshop on Cellular Networks (CellNet), 2012, pp. 31–36, ISBN 978-1-4503-1475-6.