On the Fairness of Transport Protocols in a Multi-Path Environment

Martin Becke, Thomas Dreibholz, Hakim Adhari, Erwin Paul Rathgeb
University of Duisburg-Essen, Computer Networking Technology Group
Ellernstraße 29, 45326 Essen, Germany
{martin.becke,dreibh,hakim.adhari,rathgeb}@iem.uni-due.de

***Abstract*—Today, a steadily growing number of devices contains multiple network interfaces. For example, nearly all smartphones are equipped with at least W-LAN as well as 3G/4G interfaces. In consequence, there is a rising demand for so-called multi-path transfer, which utilizes all of these interfaces *simultaneously* in order to maximize the payload throughput of applications. Currently, this so-called multi-path transfer is very actively discussed by the IETF, in form of the Multi-Path TCP (MPTCP) extension for TCP as well as the Concurrent Multi-path Transfer extension for SCTP (CMT-SCTP). Their larger-scale deployment in the Internet is expected for the near future.**

A key issue that prevents the standardization of these approaches is the fairness to concurrent TCP flows. A multi-path transfer should behave "TCP-friendly", i.e. cause no harm to the performance of the very widely deployed TCP-based applications. In this paper, we first extend the notion of "fairness" from single-path transport to multi-path transport. Furthermore, we introduce the relevant congestion control approaches in the IETF context for single-path as well as multi-path transfer. We simulatively analyze these approaches in a couple of interesting network configuration scenarios, in order to show their behavior with special regard to the fairness definitions. Particularly, we also point out items of further discussion which are the result of the current approaches.[1][2][3]

Keywords: Multi-Homing, Multi-Path Transfer, Fairness, Congestion Control, Future Internet

I. Introduction

In the last three decades, TCP has emerged as one of the most widely used network protocols in the Internet – and this is not expected to change anytime soon. With the growth of the Internet, mechanisms to coordinate the transfers and to avoid congestion collapses [1] have become necessary. In this context, so-called *Congestion Control* (CC) mechanisms play a crucial role in stabilizing the whole network [1]. In fact, every network flow should fulfill some requirements which ensure that the available resources are fairly shared among the different users. The de-facto standard for the Internet of today is that each network flow, regardless of its used protocol (e.g. TCP, SCTP [2] or DCCP), should be "TCP-friendly" [3].

The notion of "fairness" is highly crucial in this context. It has therefore already been the topic of research, notably [4]–[6]. All of these approaches have a focus on single-path transfer – which is provided by TCP. However, recent advances in transport protocol development – like Multi-Path TCP (MPTCP) [7] and Concurrent Multi-path Transfer for SCTP (CMT-SCTP) [8] – make use of multi-path transfer. That is, they utilize multiple network paths *simultaneously*, in order to improve the payload throughput performance. The resource pooling approach [9] defines design goals for multi-path congestion control mechanisms to ensure a "reasonably fair" coexistence of multi-path and single-path flows within the same network. However, this approach is just an extension of the single-path fairness approaches and is – from our point of view – inconsistent with multi-path transfer.

In this paper, we contribute to the ongoing fairness discussion for multi-path transfer by clarifying ambiguities of the currently existing approaches. In particular, we discuss the definition of the fairness concept for multi-path transfer and use simulations for validation. Based on a step-by-step analysis in chosen configuration scenarios, we introduce existing fairness definitions known for single-path flows and extend our scenarios to multi-path flows in order to show the new issues related with load sharing. In a second step, we show where the existing Congestion Control (CC) mechanisms (Reno, MPTCP and RPv2) are situated in our discussion. For this purpose, we perform simulations for the mentioned scenarios, in order to compare the results to expected theoretical values.

II. Related Work

To achieve a fair distribution of resources in a network, two classes of approaches have been proposed: centralized and decentralized ones. Centralized approaches utilize a global management instance for making decisions on resource distribution. These decisions may be based on fairness definitions such as *Max-Min Fairness* [10], *Proportional Fairness* [11] or *Weighted Proportional Fairness* [12]. In this case, communication partners and flow characteristics (e.g. bandwidth, delay, etc.) are supposed to be fixed. In reality, however, a communication may be a highly dynamical process, where the actors as well as the network characteristics are continuously changing. Therefore, decentralized approaches – where the decisions about resource allocation are made by the communication partners themselves – have been introduced. With the widespread deployment of TCP, the discussion about fairness – especially *Flow Rate Fairness* [4] which we denote in the followings sections as "flow fairness" – has moved towards a discussion about *TCP-friendliness* [3]. A flow is denoted as TCP-friendly if it behaves like a TCP flow under congestion conditions. The goal is to reach a state where bandwidth is

[1]Parts of this work have been funded by the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG).

[2]The authors would like to thank Michael Tüxen and Robin Seggelmann for their friendly support.

[3]The authors would like to thank the reviewers for their helpful comments.

equally shared among all flows. In the last couple of years, multi-homed devices (e.g. laptops with W-LAN and Ethernet interfaces, or smartphones with W-LAN and 3G/4G interfaces) have become increasingly widespread. However, classic CC mechanisms are not able to achieve a fair resource allocation when dealing with multi-path transfer. Applying TCP-friendly CC on each path independently, two sub-flows using the same shared bottleneck occupy twice the bandwidth of a standard TCP flow [13] – which would not be TCP-friendly any more. A possible solution is *Resource Pooling* (RP) [14], which means that multiple resources (here: paths) should behave like a single, pooled resource. It couples the per-path CC mechanisms in order to shift traffic from more congested to less congested paths. Releasing resources on a congested path decreases the loss rate and improves the stability of the whole network. Therefore, RP introduces a new perspective on fairness, with focus on the complete network instead of a single path only. [9] sets three design goals on RP-based multi-path CCs for a TCP-friendly Internet deployment:

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

III. Basics and Definitions

A. General Terms and Definitions

For further discussion, we formally introduce some basic definitions to have an unequivocal terminology.

Definition III.1. A *Network* $\Gamma = (\Psi, N, L, \rho)$ is defined as:

- Ψ – a finite locator set,
- $N \subseteq \mathfrak{P}(\Psi)$ – a node set,
- $L \subseteq \Psi \times \Psi$ – a link set and
- $\rho : L \rightarrow \mathbb{R}_0^+$ – a bandwidth function.

The "Uniqueness of Locators" condition applies:

$$\forall n_1, n_2 \in N : [n_1 \cap n_2 \neq \emptyset] \Rightarrow [n_1 = n_2].$$

Definition III.2. Let $\Gamma = (\Psi, N, L, \rho)$ be a network. The *Set of Paths* from node n_1 to node n_2 in Γ is defined as $\overline{P}_\Gamma(n_1, n_2) :=$

$$\begin{aligned}\big\{(\lambda_1, \ldots, \lambda_k) \in \Psi^k \ \ \big| \ \ &[1 \leq k \leq |\Psi|] \ \ \wedge \\ [\lambda_1 \in n_1] \ \ \wedge \ \ [\lambda_k \in n_2] \ \ \wedge& \\ \big[\forall i : [1 \leq i \leq k-1] \Rightarrow [(\lambda_i, \lambda_{i+1}) \in L]\big]\big\}&.\end{aligned}$$

An element $P \in \overline{P}_\Gamma(n_1, n_2)$ is denoted as *Path* from node n_1 to node n_2 in Γ. Its bandwidth $\hat{\rho}_P$ is defined as:

$$\hat{\rho}_P := \min_{1 \leq i \leq k-1} \rho\big((\lambda_i, \lambda_{i+1})\big).$$

That is, a path P is a locator sequence from node n_1 to node n_2. Its bandwidth $\hat{\rho}_P$ is the bandwidth of the slowest link (i.e. the "weakest link in the chain").

Definition III.3. Packets belonging to a certain application communication – regardless of the path(s) used – are denoted as *Flow*. A *Sub-Flow* denotes the packets of a flow that use a certain path.

Clearly, the flow of a TCP connection only consists of one sub-flow (i.e. single-path transport). However, for a multi-path transport, e.g. based on CMT-SCTP or MPTCP, one flow may contain multiple sub-flows.

B. Congestion Control Mechanisms

TCP applies a window-based CC using so-called additive-increase/multiplicative-decrease (AIMD) behavior [1] to adapt the throughput of a flow to changing network and congestion conditions as well as to ensure fairness to concurrent flows. This is the common baseline for the Internet of today [1]. However, different variants of AIMD are possible and in use; the basis of all variants to be introduced is the *Congestion Window* c_P. It denotes the upper limit for the number of outstanding bytes on path P. The *Slow-Start Threshold* s_P controls the growth rate of c_P: for $c_P \leq s_P$, the CC is in the *Slow Start* phase and c_P may increase exponentially. Otherwise, i.e. for $c_P > s_P$, the phase is denoted as *Congestion Avoidance* and only allows a linear growth.

1) Single Path Reno (Reno-SP): This is the AIMD approach applied by TCP [15] and SCTP [2]. On α newly acknowledged bytes on path P in a fully-utilised congestion window, c_P is adapted as follows [2]:

$$c_P = c_P + \begin{cases} \min\{\alpha, \mathrm{MSS}_P\} & (c_P \leq s_P) \\ \mathrm{MSS}_P & (c_P > s_P \wedge p_P \geq c_P) \end{cases}.$$

MSS_P denotes the maximum segment size (MSS) on path P. On a retransmission (RTX) on path P, s_P and c_P are adapted as follows:

$$\begin{aligned} s_P &= \max\{c_P - \frac{1}{2} * c_P, 4 * \mathrm{MSS}_P\}, \\ c_P &= \begin{cases} s_P & \text{(Fast RTX)} \\ \mathrm{MSS}_P & \text{(Timer-Based RTX)} \end{cases}. \end{aligned}$$

Fast RTX occur frequently, e.g. when queues are temporarily full; timer-based RTX are a sign of severe congestion – CC therefore goes back into slow start [15].

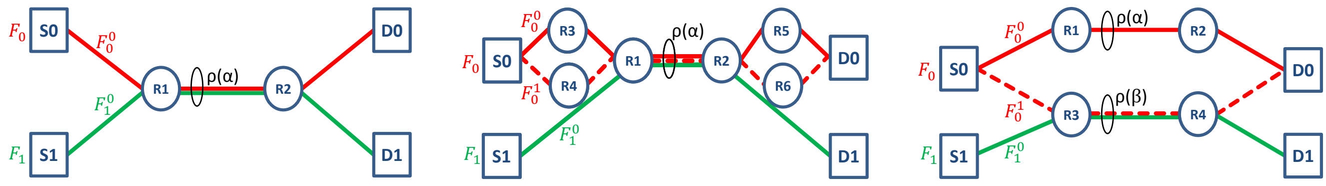
2) Multi-Path Reno (Reno-MP): For multi-path transfer, e.g. CMT-SCTP [8], [16] or MPTCP [7], a simple approach is to apply Reno-SP on each of the paths independently. However, this leads to fairness issues when some of the paths share the same bottleneck [13].

3) Resource Pooling Version 2 (RP-MP-v2): This variant from [13] applies the idea of RP to couple the CCs of the paths. To increase c_P on α acknowledged bytes on path P in a fully-utilized congestion window, the *Increase Factor* $\hat{i}_P$ – representing the bandwidth share of path P – is applied:

$$\hat{i}_P = \frac{\frac{c_P}{\mathrm{RTT}_P}}{\sum_i \frac{c_i}{\mathrm{RTT}_i}}.$$

$$c_P = c_P + \begin{cases} \lceil \hat{i} * \min\{\alpha, \mathrm{MSS}_P\} \rceil & (c_P \leq s_P) \\ \lceil \hat{i} * \mathrm{MSS}_P \rceil & (c_P > s_P \wedge p_P \geq c_P) \end{cases}.$$

The ceiling function here ensures a congestion window growth of at least one byte, in order to retain the AIMD behaviour.



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

Figure 1. The Evaluation Scenarios

For reducing c_P on a packet loss on path P, the *Decrease Factor* $\hat{d}_P$ is applied:

$$\begin{aligned}
\hat{d}_P &= \max\left\{\frac{1}{2}, \frac{1}{2} * \frac{\sum_i \frac{c_i}{\mathrm{RTT}_i}}{\frac{c_P}{\mathrm{RTT}_P}}\right\}, \\
s_P &= \max\left\{c_P - \lceil \hat{d}_P * c_P \rceil, \mathrm{MSS}_P\right\}, \\
c_P &= \begin{cases} s_P & \text{(Fast Retransmission)} \\ \mathrm{MSS}_P & \text{(Timer-Based Retransmission)} \end{cases}.
\end{aligned}$$

$\hat{d}$ represents the factor by which the bandwidth of path P should be reduced in order to halve the total flow bandwidth. That is, c_P may decrease to one MSS_P. If $\hat{d}$ would reduce c_P to a smaller value (prevented by the max function), the path P may not be used for further data transmissions during the time of one retransmission timeout (RTO) [15] on path P.

4) Multi-Path TCP (MPTCP): This CC mechanism is used by [9] to support TCP-fairness for MPTCP [7].

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

Like for RP-MP-v2, the ceiling function here also ensures an increase of at least one byte. $\hat{a}$ denotes the per-flow aggressiveness factor, which is defined as:

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

This formula is based on [9], but has been transferred from a congestion window given in TCP MSS to a congestion window given in bytes. This has been necessary, since the congestion windows of message-oriented protocols like SCTP are counted in bytes. Furthermore, the congestion window decrease behaviour has been slightly modified. In case of a retransmission (i.e. fast or timer-based) on path P, s_P and c_P are reduced as follows:

$$\begin{aligned}
s_P &= \max\left\{c_P - \frac{1}{2} * c_P, \mathrm{MSS}_P\right\}, \\
c_P &= \begin{cases} s_P & \text{(Fast Retransmission)} \\ \mathrm{MSS}_P & \text{(Timer-Based Retransmission)} \end{cases}.
\end{aligned}$$

That is, c_P may decrease to MSS_P instead of $4 * \mathrm{MSS}_P$.

IV. Fairness Perspectives

Different perspectives on "fairness" are possible.

A. Link-Centric Sub-Flow Fairness

We define *Link-Centric Sub-Flow Fairness* as the fairness interpretation based on the number of the sub-flows on a link l. Here, a fair resource allocation is defined as a bandwidth allocation of $\rho(l)/m$ for each of the m sub-flows. This is equal to the flow fairness [4] used in the current Internet, which uses a fixed 1:1 relationship between sub-flows and flows.

B. Link-Centric Flow Fairness

By using multi-path transfer, the ratio of flows to sub-flows changes from 1:1 to 1:x ($x \geq 1$). For n different flows sharing a link l, the bandwidth allocation for each flow is $\rho(l)/n$. The bandwidth share of a flow F is furthermore shared among all sub-flows of flow F using link l. We denote this kind of fairness as *Link-Centric Flow Fairness*.

C. Network-Centric Flow Fairness

The alternative to link-centric fairness is to consider fairness in the whole network instead. We denote this approach as *Network-Centric Flow Fairness*. Let there be n flows, all sharing the same paths P_1, …, P_a. Then, we define *Network-Centric Flow Fairness* as flow bandwidth allocation of

$$\left(\sum_{1 \leq i \leq a} \hat{\rho}(P_i)\right) \Big/ n.$$

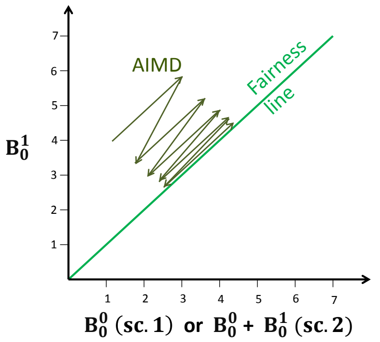
Note, that this fairness definition assumes that all flows can use all paths (i.e. a homogeneous case). A heterogeneous case, where one flow supports only a subnet of these paths and another flow a different subset, makes the definition of a network-centric fairness significantly more complex, as we will show in the following section.

V. Scenarios for Fairness Evaluation

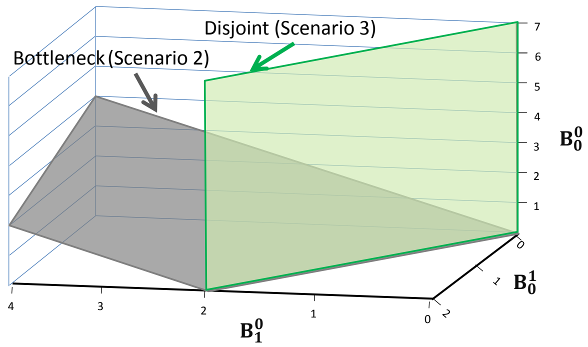
To demonstrate the challenges of fair resource allocation on network design, we have selected three interesting scenarios.

A. Evaluation Scenario 1

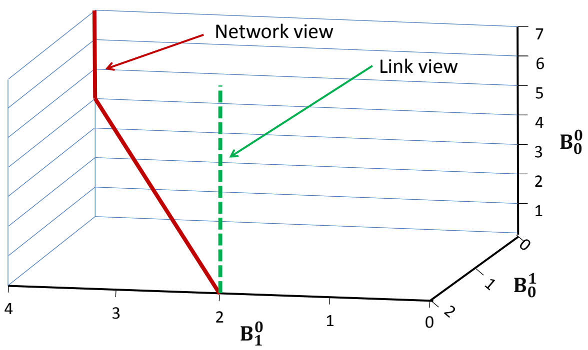
The basic scenario is shown in Subfigure 1(a). Here, four communication partners are transferring data through a shared bottleneck. The complete capacity of this link is denoted as $\rho(\alpha)$. The flow between S0 and D0 is denoted as F_0 and is composed of only one sub-flow F_0^0. The bandwidth occupied by F_0^0 is denoted as B_0^0 and the bandwidth occupied by F_0 is denoted as B_0. In this context, we assume that a protocol



(a) Fairness Line (Scenarios 1 and 2)



(b) Fairness Planes (Scenarios 2 and 3)



(c) Fairness Lines for $\rho(\beta)$=4 (Scenario 3)

Figure 2. Fairness Lines and Fairness Planes

should use a link as efficiently as possible. In this case, a link-centric flow fairness leads to the following network allocation:

$$B_0 = B_0^0 = \frac{\rho(\alpha)}{2} \quad ; \quad B_1 = B_1^0 = \frac{\rho(\alpha)}{2}.$$

Here, the network contains a single link α shared by all flows. The fairness is obvious to determine and can be visualized on the curve shown in Subfigure 2(a). The line shows that for each $B_1^0 = B_0^0$, a fair resource allocation is achieved. The convergence to this fair allocation, denoted as optimal point, is performed by the AIMD algorithm. Here, the AIMD behaviour is an adequate approach to the congestion and fairness challenge for all previously presented fairness perspectives.

B. Evaluation Scenario 2

In the next step, Scenario 1 is extended by making S0 and D0 multi-homed, as shown by Subfigure 1(b). Again, both flows share the same bottleneck and the second RP goal (i.e. “do not harm”, see Section II) has to be fulfilled. Here, a link-centric flow fairness [4] is involved; the multi-path flow should get as much resources as the single-path flow. Therefore, fairness can be described as:

$$B_0 = B_0^0 + B_0^1 = \frac{\rho(\alpha)}{2} \quad ; \quad B_1 = B_1^0 = \frac{\rho(\alpha)}{2}.$$

Subfigure 2(a) can again be used to demonstrate the fairness line. However, and in contradiction to the previous scenario, the fairness in this case is still a link-centric flow fairness but the relationship of flow to sub-flow is different, here. This has influence on the behavior of the AIMD mechanism, which has to pay attention to the flow splitting. B_0 is split between two sub-flows (F_0^0 and F_0^1). Both sub-flows are only allowed to reserve as many resources together as F_1^0 on the shared path. This is illustrated in Subfigure 2(b) by the lower plane which shows the fairness plane for the sub-flows F_0^0, F_0^1 and F_1^0. Details on the CC for such a scenario are discussed in [13].

C. Evaluation Scenario 3

In the Scenarios 1 and 2, the fairness of the bandwidth allocation has been obvious in the single-homed as well as the multi-homed case. The next scenario, which is shown in Subfigure 1(c), is going to demonstrate the ambiguity related with multi-homed configurations. Here, S0 transfers data to D0 over two paths (R1↔R2 and R3↔R4). The sub-flow F_1^0, belonging to the communication between S1 and D1, is sharing the network segment R3↔R4 with sub-flow F_0^1. In this case, especially if resilience [17] is a further goal, the fairness related with the bandwidth distribution can be considered from two points of views: a link-centric flow fairness or a network-centric flow fairness.

In the link-centric flow fairness, the allocation is:

$$B_0^0 = \rho(\alpha) \quad ; \quad B_0^1 = B_1^0 = \frac{\rho(\beta)}{2}.$$

The result of this allocation is shown in Subfigure 2(b) (upper plane). The fairness plane for Scenario 3 demonstrates that the allocation here is solely based on the parameters concerning the shared path (i.e. R3↔R4). Note, that both, Scenarios 2 and 3, have almost the same topology, with the flow F_0 in both cases being composed of two sub-flows F_0^0 and F_0^1 and with a single-path flow F_1 composed of a single sub-flow F_1^0. The only difference between both scenarios is a shared bottleneck for all paths in the first case and disjoint paths for the multi-path flow in the latter one. That is, the same flow setups lead to different fairness planes on varying topologies. For the disjoint paths of Scenario 3, the fairness plane is parallel to the B_0^0 axis, since the allocation does not depend on B_0^0 values. This is in contrast to Scenario 2, where the fairness plane takes all three parameters into consideration. That is, the AIMD mechanism has to adapt to the topology. However, this is not a simple task on a layered protocol architecture, where the Network Layer is transparent to the Transport Layer. Usage of the window-based CC mechanism, whose update frequency is based on round-trip time (RTT) measurements, may lead to unfairness, as will be discussed in the next section.

In contrast, the network-centric flow fairness considers the characteristics of the complete network. Here, F_0^0 and F_0^1 are considered as an entity which must be fair to F_1^0. The resource distribution depends on the difference between the available capacities $\rho(\alpha)$ and $\rho(\beta)$. That is, the multi-homed flow between S0 and D0 has to restrain its resource usage on the shared bottleneck when the upper path (which is used exclusively) becomes faster. This can be formalized as:

$$B_0^0 = \rho(\alpha) \;\; ; \;\; B_0^1 = \max\left\{0, \frac{\rho(\beta) - B_0^0}{2}\right\} \;\; ; \;\; B_1^0 = \rho(\beta) - B_0^1.$$

In order to demonstrate the difference between the link-centric and the network-centric flow fairness, let $\rho(\beta)$=4 Mbit/s. Subfigure 2(c) shows two lines; the dotted line presents the link-centric flow fairness in this case, the solid line presents the network-centric flow fairness. Note, that in comparison to Subfigure 2(b), the link-centric flow fairness is now only a line (instead of a plane), since only the case of $\rho(\beta)$=4 Mbit/s is considered here, in order to simplify the illustration.

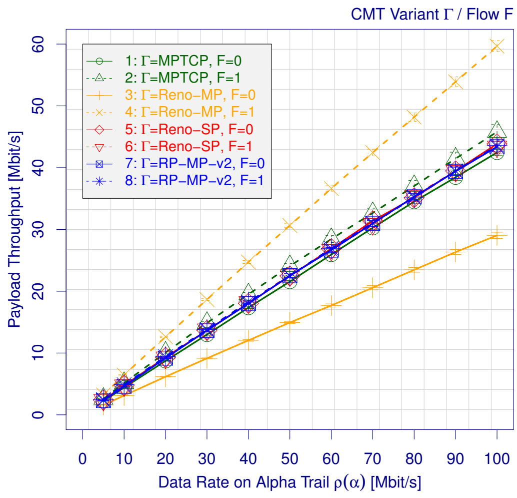
The curve shows that the bandwidth of the upper path influences the allocation on the lower one. Three different cases are of interest:

- $\rho(\alpha) = 0$: The flow between S0 and D0 can then be considered as a single-path flow; the bandwidth allocation is then equal to Scenario 1.
- $0 < \rho(\alpha) < \rho(\beta)$: In this case, the third CC goal ("balance congestion", see Section II) has to be regarded. Here, the multi-path flow should move as much traffic as possible off its most-congested path. With a growing $\rho(\alpha)$, more and more bandwidth is switched from F_0^1 to F_1^0, to fulfill the condition of $F_0^0 + F_0^1 = F_1^0$.
- $\rho(\alpha) \geq \rho(\beta)$: With the limit of $\rho(\alpha)$=4 Mbit/s reached, the multi-path flow between S0 and D0 can be considered as a single-path flow where F_0^1=0 and F_1^0=$\rho(\beta)$.

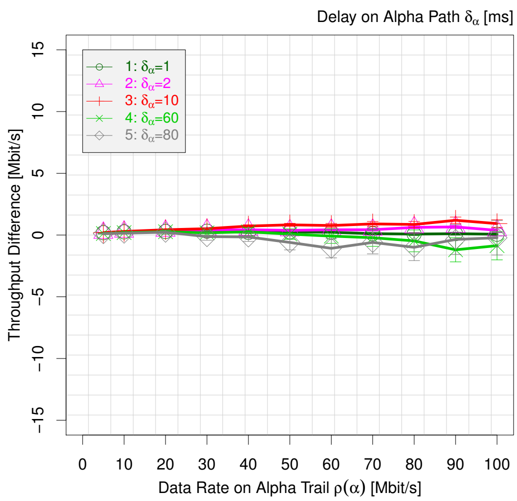
In this context, it should be noticed that the estimation of the available resources of all possible paths and their combination is much easier in this example scenario than in the Internet.

VI. Simulation Setup

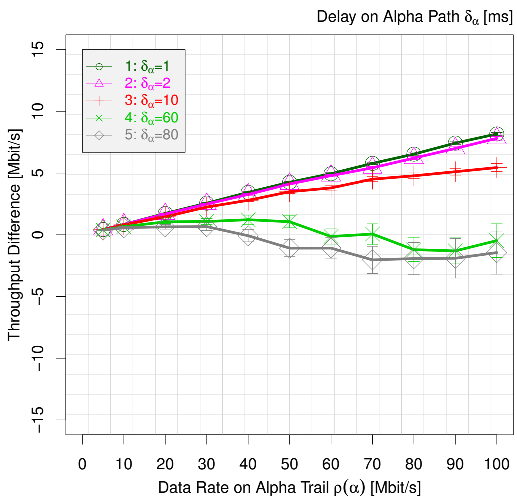
For our fairness evaluation, we have utilized the OMNeT++-based INET framework. The CC mechanisms considered in this paper have been implemented in our CMT-SCTP simulation model [13], [18]. For this purpose, we also had to port the CC mechanism introduced by [9] to our model. Note, that SCTP behaves TCP-friendly [2]. In addition to it, SCTP offers more additional functions which make it possible to avoid other side effects which would influence the simulation. In this case, in order to avoid buffer blocking issues, all messages have used unordered delivery, the send and the receive buffer sizes have been set to 5,000,000 bytes and buffer splitting [19], [20] as well as NR-SACKs [21] have been used. For parameterization and result processing, the SimProcTC [22] tool-chain has been applied. Unless otherwise specified, the following parameters have been configured: The



(a) Flow Throughput



(b) $B_0 - B_1$ for RP-MP-v2



(c) $B_0 - B_1$ for MPTCP

Figure 3. Simulation Results for Scenario 2

sender has been saturated (i.e. it has tried to transmit as much data as possible); the message size has been set to 1,452 bytes at an MTU of 1,500 bytes. In addition to it, RED queues (MinTh=30; MaxTh=90; MaxP=10% – based on [23]) have been configured on the routers. The bandwidth of each independent path has been 50 Mbit/s; the delay has been 10 ms. The simulation runtime has been 300 s, after a transient phase of 20 s. Each run has been repeated at least 25 times in order to ensure a sufficient statistical accuracy. Plots show the average values and their corresponding 95% confidence intervals.

VII. Multi-Path Congestion Control Comparison

In the following analysis, we neglect Scenario 1, since it is trivial and only needed for comparison.

A. Scenario 2

Figure 3(a) presents the results for Scenario 2, for varying $\rho(\alpha)$ from 5 Mbit/s to 100 Mbit/s. Subfigure 3(a) shows the achieved SCTP *payload* throughput for Flow n ($F = n$ denotes the flow between Sn and Dn). That is, a solid line represents the single-path flow, while a dashed line represents the multi-path flow.

Curves 5 and 6 show the results for using Reno-SP CC, i.e. they provide the baseline performance by single-path flows (the flow between S0 and D0 only uses one path here). As expected, each of the flows occupies approximately half of the available bandwidth.

The behaviour of Reno-MP CC (curves 3 and 4) differs significantly: the multi-path flow occupies a fraction of about $\frac{2}{3}$ of the available bandwidth, while the single-path flow only gets $\frac{1}{3}$. This confirms the observation from [13] that using Reno CC on each path independently, while transferring data via a shared bottleneck, leads to an unfair resource allocation from the perspective of link-centric flow fairness.

In contrast, RP-MP-v2 CC (curves 7 and 8) as well as MPTCP CC (curves 1 and 2) are able to achieve a relatively fair bandwidth share in this scenario referring to the link-centric flow fairness. That is, the difference between the bandwidths of the two flows remains small.

For a more detailed analysis, Subfigure 3(b) and Subfigure 3(c) show the difference between the throughput of the multi-path and the single-path flow for RP-MP-v2 CC and MPTCP CC, respectively, for varying settings of the delay on the bottleneck δ_α. Note the different axis scaling in comparison to Subfigure 3(a). Values larger than zero mean that the multi-path flow is too aggressive. For RP-MP-v2, the throughput of the multi-path flow is mostly below the single-path flow curve (i.e. difference$<$0) which means that this CC mechanism behaves fair – fulfilling the “do not harm” goal of Section II – for all examined delay settings. On the other hand, MPTCP CC shows an increasing unfairness (i.e. difference$>$0) for smaller delay settings. At δ_α=1 ms and $\rho(\alpha)$=100 Mbit/s, the throughput of the multi-path flow is by about 7 Mbit/s higher than for the single-path flow.

Obviously, MPTCP CC behaves not really TCP-friendly at small delays δ_α, while this effect is reduced with an increasing delay. The reason for this behaviour is the RTT. With a minimum of $c_P = \mathrm{MSS}_P$ for the congestion window of path P (see Subsubsection III-B4), at least a “stop and wait” transfer is possible on path P – regardless of the congestion. That is, the smaller the RTT, the larger the possible throughput on path P. This effect seems to be constant. In fact, the linear behavior (see curves 1 to 3) makes it possible to estimate the impact of this effect and to cancel it out. Alternatively, a correction factor could be introduced, based on the smoothed RTT, or even other approaches such as RP-MP-v2.

As shown in Subfigure 3(b), the delay variation for RP-MP-v2 has a negligible effect on the aggressiveness of this CC mechanism. When congestion is high, chances are good that a congestion window decrease would lead to a congestion window size below one MSS (which is prevented by the $\max$ function; see Subsubsection III-B3). However, in this case, the path will not be used for any new data during a time span of one RTO – which reduces the congestion. Note, that this mechanism cannot be adapted to MPTCP CC easily: extreme drops of the congestion window are rare here (see also [13, Subsection IV.C] for an example), i.e. the blocking mechanism would not be triggered often enough.

B. Scenario 3

For the analysis of Scenario 3, a fixed bandwidth of $\rho(\alpha)$=20 Mbit/s has been used, while varying $\rho(\beta)$ from 1 Mbit/s to 100 Mbit/s. Figure 4 presents the resulting SCTP payload throughput. Again, the solid line represents the performance of the multi-path flow, while the dashed line shows the results of the single-path one.

As expected, Reno-MP CC (curves 3 and 4) results in using about half of the shared path bandwidth $\rho(\beta)$ for the single-path flow, while the multi-path flow uses the full bandwidth $\rho(\alpha)$ of the other path. Note, that Reno-MP corresponds to the link-centric sub-flow fairness.

However, the most interesting results are for RP-MP-v2 CC (curves 5 and 6) as well as MPTCP CC (curves 1 and 2), since they are approximating the network-centric fairness distribution. At first, it is observable that the multi-path flow gets at least as much bandwidth as a single-path flow would get on the best path available to it (“improve throughput” goal; see Section II). RP-MP-v2 is the first CC mechanism whose performance converges to the fairness line. Also, the curves for the single and multi-path flow cross later on. After that, at $\rho(\beta)\geq$30 Mbit/s, the single-path flow obtains more bandwidth than the multi-path flow. The same effect also occurs for MPTCP CC, at $\rho(\beta)\geq$70 Mbit/s.

While this effect – which is a result of the “balance congestion” goal – seems surprising at first, it is in compliance with the goals defined in Section II. Also, from a network-centric perspective, it is furthermore useful. However, it points out to fairness questions that still have to be discussed. Consider for example a consumer paying for two high-speed Internet access lines. Even if the CC mechanisms considered in this paper are able to achieve their goals, it should be discussed whether it is really “fair” if a consumer with only one Internet access (i.e. the single-path flow) may get even more resources than a consumer who is paying twice (i.e. the multi-path flow).

Furthermore, another general aspect has become visible in the performed simulations: while both, RP-MP-v2 as well as

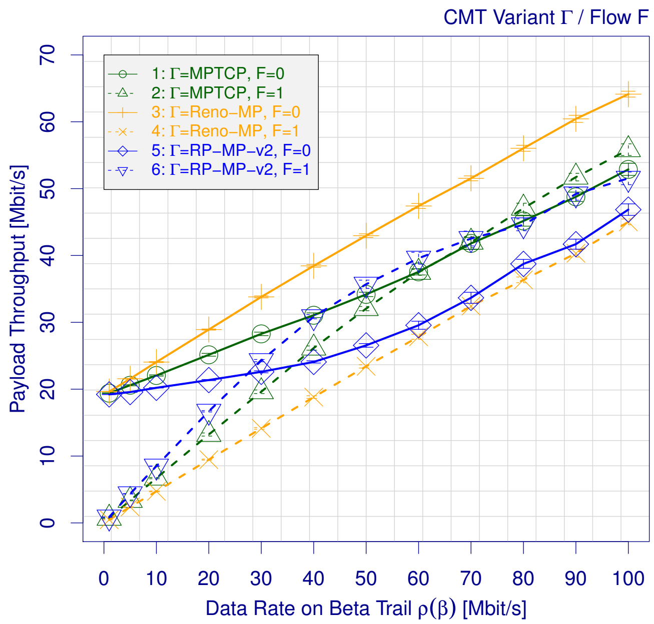


Figure 4. Throughput in Scenario 3

MPTCP CC, are targeting the same three goals, they realize them in different ways – which makes the one more aggressive than the other. A simultaneous deployment of both approaches in the Internet would cause a possible unfairness between multi-path transport protocols. The current state is that we expect a growing deployment of MPTCP CC [9] in the near future. With this deployment, we also expect that the fairness discussion will move on, in the same way as for TCP, to a discussion of "MPTCP-friendliness" instead.

VIII. Conclusion

At the moment, the direction of how to deploy the multi-path transport protocols MPTCP and CMT-SCTP are determined in the IETF. A topic which is currently still under discussion is the fairness of multi-path transport flows in comparison to TCP flows. For the standardization, it is crucial that new multi-path congestion control approaches behave TCP-friendly, in order to avoid harm to the widely deployed applications based on standard, i.e. single-path, TCP flows.

In this paper, we have given an overview of existing fairness definitions and extended the notion of fairness from single-path flows to multi-path flows. Furthermore, we have introduced the congestion control mechanisms which are currently relevant in the IETF context for multi-path transfer standardization. Based on these approaches, we have compared their performance with respect to the fairness definitions. Particularly, we have shown that there are still open issues, and the term of "fairness" for multi-path transfer still needs further research with regard to certain network topologies.

References

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

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

[3] B. Braden, D. D. Clark, J. Crowcroft, B. Davie, S. E. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet," IETF, Informational RFC 2309, Apr. 1998, ISSN 2070-1721.

[4] B. Briscoe, "Flow Rate Fairness: Dismantling a Religion," *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 63–74, Mar. 2007, ISSN 0146-4833.

[5] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," DEC, Tech. Rep. Research Report TR-301, Sept. 1984.

[6] T. Bonald and L. Massoulié, "Impact of Fairness on Internet Performance," in *Proceedings of ACM SIGMETRICS*, Cambridge, Massachusetts/U.S.A, June 2000, pp. 82–91, ISBN 1-58113-334-0.

[7] A. Ford, C. Raiciu, M. Handley, S. Barré, and J. R. Iyengar, "Architectural Guidelines for Multipath TCP Development," IETF, Informational RFC 6182, Mar. 2011, ISSN 2070-1721.

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

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

[10] E. L. Hahne, "Round-Robin Scheduling for Max-Min Fairness in Data Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1024–1039, Sept. 1991.

[11] R. Mazumdar, L. G. Mason, and C. Douligeris, "Fairness in Network Optimal Flow Control: Optimality of Product Forms," *IEEE Transactions on Communications*, vol. 39, pp. 775–982, May 1991, ISSN 0090-6778.

[12] F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, Jan. 1997.

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

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

[15] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," IETF, Standards Track RFC 5681, Sept. 2009, ISSN 2070-1721.

[16] T. Dreibholz, I. Rüngeler, R. Seggelmann, M. Tüxen, E. P. Rathgeb, and R. R. Stewart, "Stream Control Transmission Protocol: Past, Current, and Future Standardization Activities," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 82–88, Apr. 2011, ISSN 0163-6804.

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

[18] T. Dreibholz, H. Adhari, M. Becke, and E. P. Rathgeb, "Simulation and Experimental Evaluation of Multipath Congestion Control Strategies," in *Proceedings of the 2nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Fukuoka/Japan, Mar. 2012.

[19] T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tüxen, "On the Use of Concurrent Multipath Transfer over Asymmetric Paths," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Miami, Florida/U.S.A., Dec. 2010, ISBN 978-1-4244-5637-6.

[20] H. Adhari, T. Dreibholz, M. Becke, E. P. Rathgeb, and 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, Mar. 2011, pp. 708–714, ISBN 978-0-7695-4338-3.

[21] P. Natarajan, N. Ekiz, E. Yilmaz, P. D. Amer, and J. R. Iyengar, "Non-Renegable Selective Acknowledgments (NR-SACKs) for SCTP," in *Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP)*, Orlando, Florida/U.S.A., Oct. 2008, pp. 187–196, ISBN 978-1-4244-2506-8.

[22] T. Dreibholz, X. Zhou, and 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, Mar. 2009, pp. 1–8, ISBN 978-963-9799-45-5.

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