Is It Really Necessary to Go Beyond A Fairness Metric for Next-Generation Congestion Control?

Safiqul Islam
University of South-Eastern Norway
Norway
safiqul.islam@usn.no

Kristian Hiorth
University of Oslo
Norway
kristahi@ifi.uio.no

Carsten Griwodz
University of Oslo
Norway
griff@ifi.uio.no

Michael Welzl
University of Oslo
Norway
michawe@ifi.uio.no

ABSTRACT

A recent article suggests that the potential for deployment of congestion control mechanisms in the future Internet should be evaluated using a new concept called "harm" instead of measuring "fairness". While there are good arguments in favor of this new approach, its practical benefits have not yet been experimentally evaluated, and calculating harm requires producing more experimental data. We apply the harm concept to data produced in real-life experiments with competing pairs of various TCP variants: Cubic vs. Reno, BBR vs. Cubic, and Reno vs. Vegas. These experiments cover various levels of "aggression" as well as different feedback types that the controls are based upon. We present a new linear representation of relative harm between scenarios, which can help us to assess the differences in harm between a variety of situations. Among other results, we can see that BBR is on average 1.6 times more harmful to Cubic in high-BDP situations (when Cubic is most aggressive) than Cubic is to Reno.

CCS CONCEPTS

• **Networks** → **Transport protocols**; **Network experimentation**.

KEYWORDS

Fairness metric, BBR, congestion control, TCP, Cubic, Harm

ACM Reference Format:
Safiqul Islam, Kristian Hiorth, Carsten Griwodz, and Michael Welzl. 2022. Is It Really Necessary to Go Beyond A Fairness Metric for Next-Generation Congestion Control?. In *Applied Networking Research Workshop (ANRW '22), July 25–29, 2022, PHILADELPHIA, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3547115.3547192

1 INTRODUCTION

The evaluation of congestion control mechanisms traditionally encompasses an examination of fairness. Commonly, the throughput of competing transfers ("flows") is measured in experiments or obtained via simulations, and from this data vector, a singular value is calculated with Jains Fairness Index (JFI) [10], using the formula:

$$JFI = \frac{(\sum_{i=1}^{N} x_i(t))^2}{N \sum_{i=1}^{N} x_i(t)^2} \tag{1}$$

Here, N is the total number of flows, and $x_i(t)$ is the throughput of the i^{th} connection — the output ranges from $1/N$ to 1 where the value 1 indicates that all flows get the same allocation.

The concept of fairness has been used to judge how well multiple instances of the same congestion control mechanism interoperate, under homogeneous or heterogeneous conditions, and it has also been used to evaluate whether a new mechanism might be fit for deployment in the future Internet. The latter test is usually done by evaluating fairness when the new mechanism competes with the prevalent Internet congestion control mechanism — traditionally NewReno [1], but more recently Cubic [17].

The authors of [14] question that such a fairness test indeed provides a good basis for reasoning about the deployment of a congestion control algorithm. They argue that it

ANRW '22, July 25–29, 2022, PHILADELPHIA, PA, USA

ACM ISBN 978-1-4503-9444-4/22/07…$15.00
https://doi.org/10.1145/3547115.3547192

would be better to use the concept of how *harmful* a new entrant CC algorithm is to incumbent CC algorithms. Specifically, they identify (and suggest a fix to) three key problems of the current fairness-based approach:

(1) For deployment, it is not necessary to strive for exactly equal capacity-sharing between flows implementing the common and a new congestion control,
(2) throughput alone is not sufficient as an input metric,
(3) a fairness metric such as JFI cannot show the difference between a negative and a positive bias (i.e., whether a new congestion control mechanism takes a *larger* or a *smaller* share of the available capacity).

While the arguments underlying the introduction of harm are solid, so far, its practical merit has not been demonstrated. This is necessary because using harm comes at a cost: its calculation requires producing more experimental data than the calculation of a fairness metric like JFI.

The *contribution of this paper* is to provide the first evaluation of using a fairness metric vs. using harm with representative congestion control mechanisms that are known to exhibit various degrees of "aggression": Reno [1], Cubic [17], and BBR [5][1] as well as the delay-based Vegas [3], which is known to be less aggressive than Reno.

To evaluate the potential multi-metric benefit of harm, we focus not only on throughput but also consider latency in our evaluation.

In the next section, we discuss related work. In Section 3, we present our evaluation setup and explain how we calculate harm as well as fairness in a way that lets us arrive at comparable metrics. We present results in Section 4. Section 5 concludes.

2 RELATED WORK

In the late 1990s, the term "TCP-friendly" was coined to describe a form of fairness among congestion controlled flows that can be used as a criterion for Internet compatibility [7]. A TCP-friendly flow is a flow that, on average, does not exceed the rate of any conformant TCP flow in the same circumstances. For a while, "TCP-friendliness" was used as the criterion to determine whether a congestion control mechanism is fit for Internet deployment, and accordingly, several TCP-friendly mechanisms were developed [16]. Later, this requirement has been implicitly relaxed by the widespread use of Cubic [17]: Cubic is more aggressive than standard TCP when the Bandwidth×Delay Product (BDP) is large, yet falls back to a TCP-friendly behavior in small-BDP conditions.

The term "flow-rate fairness" has been introduced in [4] to stress that requiring equal rates between individual flows is a very specific — and, as Briscoe [4] argues, not very useful — type of fairness. Rather, fairness should be defined in relation to a cost, per economic entity and not per flow (or else a single user can be unfair by simply opening multiple TCP connections). However, close to 15 years after the publication of [4], it is still common to evaluate mechanisms on the basis of flow-rate fairness today.

Ware et al. [14] undertake a first step towards a realistic change in the judgement of whether a mechanism is fit for Internet deployment. They recognize that developers of modern congestion control algorithms focus on various performance metrics: not only throughput, but also delay, loss, and flow completion time, for example. Hence, trying to define fairness by requiring equal rates (i.e., "flow-rate fairness") is not always meaningful, and harm can be expressed in terms of these other performance metrics as well.

In this paper, we set out to test if a harm-based approach is a better practical alternative than evaluating "flow rate fairness". To the best of our knowledge, only Athapathu et al. [2] practically applied the harm metric, to demonstrate the goodput harm done to bulk transfers (using Reno, Cubic, and BBRv1) by a Youtube stream. We do not carry out an in-depth evaluation focused on the deployability of BBR, since this has already been done in several prior works [6, 8, 9, 13, 15] — instead, we provide a first evaluation of the harm-based approach using four well-known congestion control algorithms (Reno, Cubic, BBRv2, Vegas). In the following, we refer to BBRv2 as "BBR" (this is simply the latest version in the Linux kernel, and we used kernel version 5.10.0 in our experiments).

3 MEASUREMENT SETUP

The overarching objective of our experiments is to compare the harm-based approach with fairness, using four well-known congestion control mechanisms: Reno, Cubic, BBR and Vegas. To this end, we ran experiments in our physical TEACUP testbed [18], which consists of a classical dumbbell topology where sources and destinations are in different subnets. In our experiments, we varied the link capacity to 10, 25, 50, 75, and 100 Mbps, and RTT to 10, 20, 50, 100 ms, respectively. The queue size was set to half a BDP and a full BDP for each bandwidth and delay case. To determine a reasonable test run length, we examined the case in which it took the longest for all congestion control algorithms to experience at least 5 congestion events (saw-teeth in case of Reno). This case was the largest-BDP, maximum queue one (i.e., 100 Mbps, 100 ms, 1 BDP queue length) where Reno was competing with Reno. This gave us a run length of 5 minutes,

[1] The provided reference introduces the principles behind BBR, but the mechanism itself has undergone many later changes, implicating a name update to "BBRv2". These changes are documented in a series of presentations which can be found at https://groups.google.com/g/bbr-dev

which we applied for all tests. We also cut the first 3 seconds of all tests to remove the transient effect of Slow Start and focus only on the long-term behavior of the Congestion Avoidance phase.

3.1 How to calculate "harm"?

In [14], the calculation of harm is described as follows:

> *Let x = demand (solo performance); let y = performance after introduction of a competitor connection. For metrics where "more is better" (like throughput and QoE) harm is $(x - y)/x$. For metrics where "less is better" (like latency or flow completion time) harm is $(y - x)/x$.*

This is a meaningful definition of "harm", as it truly reflects harm done to a flow, but it is not useful for comparison against a fairness metric. It is mathematically different than Jain's Fairness Index, but it is not our intention to compare a $(y - x)/x$ fairness calculation against JFI.

Harm, to be useful in this context, needs to be put into relation to something. Accordingly, for example in [2], it is stated: "YouTube is the subject whose harm we are evaluating, the bulk data transfer using iperf3 is the victim, and a standard DASH video service acts as the baseline (which quantifies acceptable harm)". Perhaps more useful for a comparison, [14] contains the following deployment criterion:

> *We suggest that, if the harm done by a new CCA alpha to a widely-deployed CCA beta is comparable or less than the harm done when beta competes against beta, we should consider it acceptable to deploy.*

Here, "CCA" is an abbreviation of "Congestion Control Algorithm".

3.2 Representation suitable for comparisons

Taking this as a basis, we carry out two tests for all our scenarios. In the first, a flow α implementing a "new" congestion control mechanism (e.g. Cubic) competes against a flow β implementing the "baseline" congestion control (Reno in most of our tests). In the second, the two baseline flows β_1 and β_2 compete with each other. This allows us to implement the suggestion in [14] by comparing β with one of the flows from the second test, e.g. β_1. We express our independence from a specific metric such as bandwidth in the following discussion by referring to a mapping from a flow to a specific measurement m : flow $\rightarrow$ metric value.

To understand the fairness of a specific scenario, we must understand how close $\frac{m(\beta)}{m(\alpha)}$ is to 1, which would be perfectly fair. Harm, however, must compare how strongly the fairness of the bandwidth of a baseline flow in the first test diverges from the bandwidth of the baseline flow in the second test. This is measured as $\frac{m(\beta)}{m(\beta_1)}$. Note that we keep the same reference value as the numerator in both cases.

It is easy to talk about the resulting value, but as a ratio term, it is very inconvenient for comparisons. One inconvenience is that it makes no sense to compute the average of invidual samples, taking the harmonic mean is required instead and variance or standard deviation can only be computed asymptotically. Another inconvenience is that the resulting values are not symmetrical: if $m(\alpha)$ is greater than $m(\beta)$, the ratio $\frac{m(\beta)}{m(\alpha)}$ is in the range of 0..1, whereas the opposite yields unbounded values above 1.

We therefore return to a much simpler illustration:

$$metric(x, y) = \begin{cases} 1 - \frac{y}{x} & \text{if } \frac{x}{y} < 1 \\ 0 & \text{if } \frac{x}{y} = 1 \\ \frac{x}{y} - 1 & \text{otherwise} \end{cases} \quad (2)$$

Here, x and y are input measurements; with $x = m(\beta)$ and $y = m(\alpha)$, Equation 2 yields a fairness metric, which we call m_f. With $x = m(\beta)$ and $y = m(\beta_1)$, Equation 2 yields a harm metric, which we call m_h. Negative values correspond to α causing much harm to β, i.e. $m(\beta)$ is much smaller than $m(\beta_1)$ and $1 - \frac{y}{x}$ becomes negative; zero means no harm. Inversely, positive values indicate that β harms α when they compete. Figure 1 shows a high-level overview of our metric (m) calculation from two separate experiments.

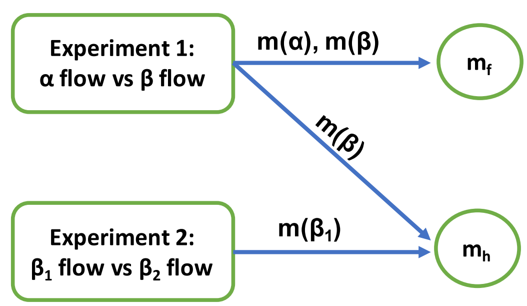


Figure 1: Fairness (m_f) and harm (m_h) calculation from two experiments.

We initially considered to use an expression based on $log_2(\frac{x}{y})$. This value does already allow us to express the relation between two ratio terms, and it does have the benefit that the absolute log_2 values of the ratios $x : y$ and $y : x$ are the same with inverted signs. It is furthermore beneficial that this representation has a continuous derivative and doesn't require cases. However, the logarithmic value means that the standard deviation in illustrations would be asymmetric and our intuitive understanding of absolute distance from 0 allows us only to order different results, not to understand how intense a given harm is. The ability to illustrate the differences in harm and fairness visually, and the possibility to compare them on a linear scale and to compute the variance over the entire range of samples has led us to proceed with Equation 2.

4 RESULTS AND DISCUSSION

In this section, we show the efficacy of the harm metric using four well-known congestion control mechanisms. The rationale behind choosing these congestion control mechanisms is the level of aggression and the congestion control signal they use. Table 1 outlines the properties of the four mechanisms that we use in our experiments.

Name	Aggression	Loss-based	Delay-based
Reno	+	x	o
Cubic	++	x	o
BBR	+++	x♣	x
Vegas	-	o†	x

♣ While BBRv1 ignored explicit loss notifications, BBR has always indirectly used loss (in addition to delay), as it measures the attained throughput reflected by ACKs.

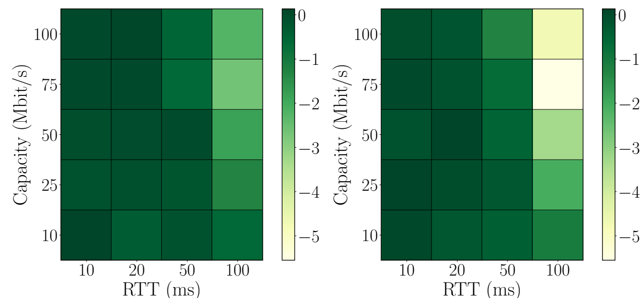
† Vegas does not ignore loss, but it is *primarily* a delay-based mechanism.

Table 1: Overview of congestion control algorithms.

We investigate the harm metric for Cubic vs Reno and Cubic vs BBR flows. We also investigate how the harm-metric behaves when we run tests with a loss-based flow (Reno) against a delay-based flow (Vegas). Finally, we share our experiences on whether the deployability of new congestion control mechanisms could better be judged using a harm-based approach.

Despite being more aggressive than the previously prevalent Reno, Cubic has become widely deployed without causing an Internet "meltdown". Today, it is in fact the dominant congestion control mechanism on the Internet [11]. In principle, if a new, even more aggressive congestion control mechanism such as BBR does no more harm to Cubic than the harm done by Cubic to Reno, rolling out this mechanism on the Internet should also not cause a meltdown. We therefore set out to look at these two scenarios in terms of fairness and harm.

In our first Cubic vs. Reno tests, we found that there was no noticeable difference at all between fairness and harm in a certain parameter range, and fairness was almost perfect. This is because Cubic behaves similar to Reno (in its "TCP-friendly region") when the BDP is small [17]. For a single flow, these operating conditions could be calculated as a function of the BDP, but when two flows compete, the switch between a Reno-like and a more aggressive behavior is less clear. One or both of the flows can end up operating in the "TCP-friendly region" more or less frequently.



(a) Queue sized to half BDP **(b) Queue sized to full BDP**

Figure 2: Fairness of one Cubic connection competing with one Reno connection, for varied RTT and capacity values.

As a first step, we therefore ran experiments with Cubic vs. Reno to identify and eliminate scenarios where Cubic falls back to linear TCP-like growth. Figure 2 gives an overview of fairness over the whole parameter space of our tests. Based on these results, we decide that the fairness comparison between Cubic and Reno is "interesting" at RTT=50 ms, capacity≥75Mbit/s as well as all tests where RTT=100 ms. In the following, we refer to this as a high-BDP scenario, and we restrict our considerations about relative harm and fairness to these settings for all of the tests involving Cubic.

4.1 Harm and fairness distribution

In this section, we examine the distribution of both harm m_h and fairness m_f between several different α and β congestion control algorithm pairs. We consider harm with regards to throughput, which we refer to as $m_h(tput)$, as well as harm with respect to RTT, termed $m_h(RTT)$.

4.1.1 Intuitive interpretation. Figure 3 shows the distribution of $m_h(tput)$ for several α and β congestion control algorithm pairs. Negative values correspond to α causing much harm to β, and positive values indicate that β harms α when they compete (see section 3.2). As expected from the previously known behaviour of these CC algorithms, we see that Cubic causes moderate harm to Reno, BBR causes considerable harm to Cubic, whilst Vegas is actually harmed by Reno. This highlights the directional bias benefit of harm over fairness (problem 3 mentioned in the introduction).

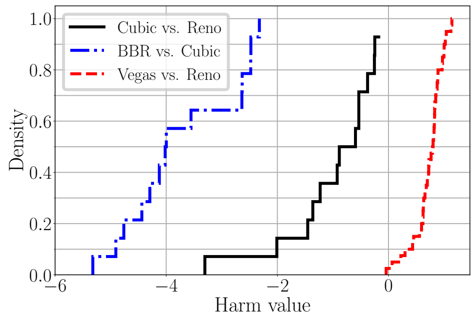


Figure 3: Cumulative distribution function of $m_h(tput)$ values for α=Cubic vs. β=Reno, α=BBR vs. β=Cubic and α=Vegas vs. β=Reno.

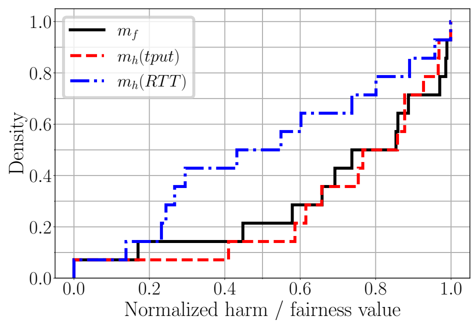


Figure 4: Cumulative distribution function of normalized m_f and m_h values measured for varied α=Cubic and β=Reno pairs, across the high BDP parameter space. Larger values are better for the β flow.

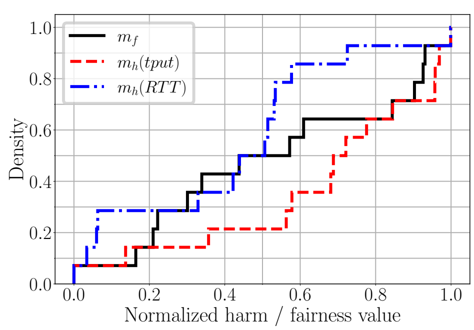


Figure 5: Cumulative distribution function of normalized m_f and m_h values measured for varied α=BBR and β=Cubic pairs, across the high BDP parameter space. Larger values are better for the β flow.

4.1.2 Relative harm and fairness distribution in different cases. To understand the merit of harm, we display the cumulative distribution function of normalized m_f and m_h values in Figures 4 to 6. By normalizing, we are able to compare

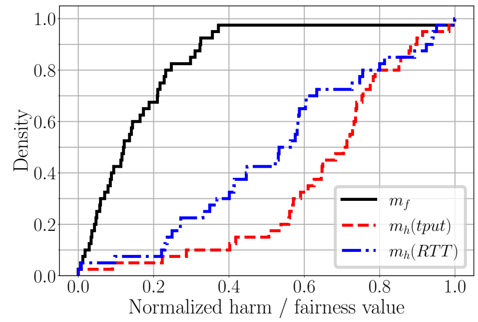


Figure 6: Cumulative distribution function of normalized m_f and m_h values measured for a Vegas α competing with a Reno β, across the entire parameter space. Larger values are better for the β flow.

the behavioral trend/traits of the m_f and m_h without taking into account how far they are "shifted" along the x index. Since all the values per scenario stem from the same test, we can apply normalization across all the data for each scenario (i.e., per diagram). That is, we individually call the dataset of each of the metrics m_f, $m_h(tput)$ and $m_h(RTT)$ a vector $\vec{x}$, and we calculate its normalization $n(\vec{x})$ as:

$$n(\vec{x}) = \frac{\vec{x} - min(\vec{x})}{max(\vec{x}) - min(\vec{x})} \tag{3}$$

As a result, the value 0 represents the most aggressive behavior of α vs. β while the least aggressive (most fair, minimal harm) behavior is represented by the value 1.

Figure 4 shows a case where the normalized distribution of m_f closely follows that of $m_h(tput)$. In this case it is not obvious that there is much value to looking at $m_h(tput)$ rather than the simpler m_f metric. However, the $m_h(RTT)$ distribution does give us an insight: the impact of Cubic on Reno is worse in terms of the RTT than in terms of throughput.

In Figure 5 we observe that in the BBR vs. Cubic case, the difference between m_f and $m_h(tput)$ is much more pronounced. Remember that our harm metric reflects the difference between the harm that congestion control algorithm α does to congestion control algorithm β on the one hand, and the harm that one instance of the congestion control algorithm β (dubbed $β_1$) does to another instance of β (dubbed $β_2$) on the other hand. Fairness can be interpreted as quantifying the deviation from perfect capacity sharing. If, say, the fairness between multiple instances of algorithm β is perfect, we therefore do not expect to see a difference between a normalized version of our fairness and harm metrics. The fairness between two Reno flows is better than the fairness between two Cubic flows (Cubic is a more complex, modal algorithm), and thus the gap between m_f and $m_h(tput)$ is wider in Figure 5.

In Figure 6 we show the harm and fairness distributions of the delay-based TCP Vegas algorithm competing with Reno, sampled across the entire measured parameter space. From a quick glance, the m_f line being so far to the left (where small values mean a more aggressive behavior) might seem confusing, as Vegas was α in this scenario, yet it is known to be *less* aggressive than Reno. Indeed, Reno's more aggressive behavior is clearly visible in Figure 3. Since all values in Figure 6 are normalized, the m_f line being so far to the left only means that Vegas behaves very differently in terms of fairness than in terms of harm. This is caused by the Vegas flows (β_1 vs. β_2) being quite unfair to each other. Simple delay-based mechanisms such as Vegas are prone to various unfairness problems, e.g. "latecomer advantage", where one flow mistakes another's constantly produced delay as the "base delay" in an uncongested network [12].

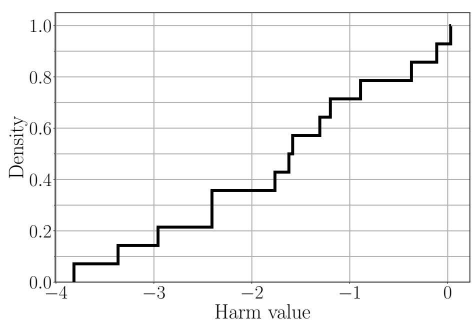


Figure 7: Throughput harm comparison between (α: Cubic, β: Reno) and (α: Cubic, γ: BBR) cases in the high BDP scenario.

4.1.3 Case study of absolute fairness and harm. In Figure 7 we compare two scenarios that are relevant with regards to the deployment of new CC algorithms. The first scenario is that of the historical introduction of Cubic at a time when Reno was the dominant algorithm, the second is the contemporary case of possibly having BBR supersede Cubic. This is different from Figure 1: we now calculate m_h by using Cubic as α and Reno as β in experiment 1, and in experiment 2, we use α (Cubic) and γ (BBR) instead of β_1 and β_2. Figure 8 illustrates how we calculate harm for these experiments.

Here, in contrast to Figures 4 and 5, we show raw (m_h) values.

Figure 7 shows that BBR captures more resources from Cubic than Cubic captures from Reno in nearly all cases. More specifically, it shows that BBR captures at least 1.6 times more resources for 50% of the cases and at least 2 times more in 38% of the cases.

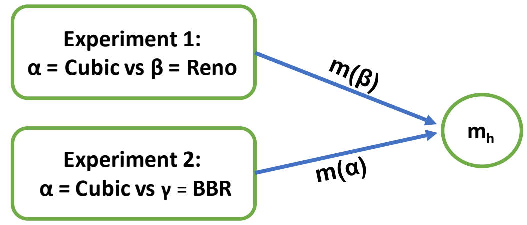


Figure 8: Harm (m_h) calculation from two experiments (α (Cubic) vs β (Reno) and α (Cubic) and γ (BBR)).

Clearly, this is a far more aggressive behaviour that could pose worse problems in a network environment shared between BBR and Cubic than we experienced when Reno and Cubic coexisted.

5 CONCLUSION

In this paper, we have applied the harm concept to data produced in real-life experiments with competing pairs of various TCP variants: Cubic vs. Reno, BBR vs. Cubic, and Reno vs. Vegas. Our experiments have covered various levels of "aggression" as well as different feedback types that the controls are based upon. A new linear representation of relative harm between scenarios is presented to better assess the differences in harm between a variety of situations. Results show that BBR is on average 1.6 times more harmful to Cubic in high-BDP situations than Cubic is to Reno.

Based on our experiments, we have found that the harm based approach is more useful to judge whether a next-generation congestion control mechanism is safely deployable in the future Internet. The harm metric supports a wide range of quality metrics, making it more attractive to use for an evaluation of next-generation flow management than a fairness metric. We intend to investigate the efficacy of harm with some other quality metrics such as loss in our future work.

REFERENCES

[1] M. Allman, V. Paxson, and E. Blanton. 2009. TCP Congestion Control. RFC 5681 (Draft Standard). https://doi.org/10.17487/RFC5681

[2] Rukshani Athapathu, Ranysha Ware, Aditya Abraham Philip, Srinivasan Seshan, and Justine Sherry. 2020. Prudent: A Watchdog Service for Measuring Congestion Control Harm on the Internet. In *ACM SIGCOMM 2020 Networking Networking Women Professional Development Workshop (N2Women'20)*. https://www.justinesherry.com/papers/athapathu-n2women20.pdf

[3] L.S. Brakmo and L.L. Peterson. 1995. TCP Vegas: end to end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications* 13, 8 (1995), 1465–1480. https://doi.org/10.1109/49.464716

[4] Bob Briscoe. 2007. Flow rate fairness: dismantling a religion. *ACM SIGCOMM Computer Communication Review* 37, 2 (2007), 63–74.

[5] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2016. BBR: Congestion-Based Congestion Control. *ACM Queue* 14, September-October (2016), 20 – 53. http://queue.acm.org/detail.cfm?id=3022184

[6] Ferenc Fejes, Gergő Gombos, Sándor Laki, and Szilveszter Nádas. 2019. Who Will Save the Internet from the Congestion Control Revolution?. In *Proceedings of the 2019 Workshop on Buffer Sizing* (Palo Alto, CA, USA) *(BS '19)*. Association for Computing Machinery, New York, NY, USA, Article 7, 6 pages. https://doi.org/10.1145/3375235.3375242

[7] S. Floyd and K. Fall. 1999. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking* 7, 4 (1999), 458–472. https://doi.org/10.1109/90.793002

[8] Mario Hock, Roland Bless, and Martina Zitterbart. 2017. Experimental evaluation of BBR congestion control. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 1–10.

[9] Per Hurtig, Habtegebreil Haile, Karl-Johan Grinnemo, Anna Brunstrom, Eneko Atxutegi, Fidel Liberal, and Åke Arvidsson. 2018. Impact of TCP BBR on CUBIC Traffic: A Mixed Workload Evaluation. In *2018 30th International Teletraffic Congress (ITC 30)*, Vol. 01. 218–226. https://doi.org/10.1109/ITC30.2018.00040

[10] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. 1984. *A quantitative measure of fairness and discrimination*. Technical Report. Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA.

[11] Ayush Mishra, Xiangpeng Sun, Atishya Jain, Sameer Pande, Raj Joshi, and Ben Leong. 2019. The Great Internet TCP Congestion Control Census. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 3, Article 45 (Dec. 2019), 24 pages. https://doi.org/10.1145/3366693

[12] David Ros and Michael Welzl. 2013. Less-than-Best-Effort Service: A Survey of End-to-End Approaches. *IEEE Communications Surveys Tutorials* 15, 2 (2013), 898–908. https://doi.org/10.1109/SURV.2012.060912.00176

[13] Dominik Scholz, Benedikt Jaeger, Lukas Schwaighofer, Daniel Raumer, Fabien Geyer, and Georg Carle. 2018. Towards a Deeper Understanding of TCP BBR Congestion Control. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. 1–9. https://doi.org/10.23919/IFIPNetworking.2018.8696830

[14] Ranysha Ware, Matthew K. Mukerjee, Srinivasan Seshan, and Justine Sherry. 2019. Beyond Jain's Fairness Index: Setting the Bar For The Deployment of Congestion Control Algorithms. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks (HotNets '19)*. Association for Computing Machinery, 17–24. https://doi.org/10.1145/3365609.3365855

[15] Ranysha Ware, Matthew K Mukerjee, Srinivasan Seshan, and Justine Sherry. 2019. Modeling BBR's interactions with loss-based congestion control. In *Proceedings of the internet measurement conference*. 137–143.

[16] J. Widmer, R. Denda, and M. Mauve. 2001. A survey on TCP-friendly congestion control. *IEEE Network* 15, 3 (2001), 28–37. https://doi.org/10.1109/65.923938

[17] Lisong Xu, Sangtae Ha, Injong Rhee, Vidhi Goel, and Lars Eggert. 2021. *CUBIC for Fast and Long-Distance Networks*. Internet-Draft draft-ietf-tcpm-rfc8312bis-04. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-tcpm-rfc8312bis-04 Work in Progress.

[18] Sebastian Zander and Grenville Armitage. 2015. Teacup v1. 0-a system for automated TCP testbed experiments. *Centre for Advanced Internet Architectures, Swinburne University of Technology, Tech. Rep. A* 140314 (2015), 2014.