Don't Hate the Player, Hate the Game: Safety and Utility in Multi-Agent Congestion Control

Pratiksha Thaker
prthaker@stanford.edu
Stanford University

Matei Zaharia
matei@cs.stanford.edu
Stanford University

Tatsunori Hashimoto
thashim@stanford.edu
Stanford University

Abstract

We posit that unfairness between congestion control algorithms in a network often results from actors optimizing, or attempting to optimize, incompatible utility functions. In order to mitigate this, we propose that algorithm designers should explicitly declare a utility function they hope to optimize, which enables theoretical analysis of the safety of the utility function with respect to other utilities in the network. When we can place utilities in a common game-theoretic framework, we can analytically determine the potential for an application with one of those utilities to be unsafe before it is deployed in a network, rather than determining safety properties ad-hoc from measurements after deployment. We give examples of the types of restrictions and guarantees that can arise from such a model in the context of rate-based congestion control protocols.

CCS Concepts

• **Networks → Network resources allocation**;

Keywords

congestion control, safety

ACM Reference Format:
Pratiksha Thaker, Matei Zaharia, and Tatsunori Hashimoto. 2021. Don't Hate the Player, Hate the Game: Safety and Utility in Multi-Agent Congestion Control. In *The Twentieth ACM Workshop on Hot Topics in Networks (HotNets '21), November 10–12, 2021, Virtual Event, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3484266.3487392

HotNets '21, November 10–12, 2021, Virtual Event, UK

ACM ISBN 978-1-4503-9087-3/21/11…$15.00
https://doi.org/10.1145/3484266.3487392

1 Introduction

In the last decade, congestion control protocol designers have tacitly agreed to move beyond the restrictions of TCP window increase-decrease protocols, designing new algorithms that attempt to optimize for new performance metrics that TCP windowing protocols may perform poorly on. Some proposals make these goals explicit through algorithms that attempt to maximize an explicitly stated utility, such as PCC-Vivace, Remy, and Copa [4, 10, 11, 28], while others, such as BBR [6], describe an algorithm that aims to empirically maximize a local performance objective.

In the midst of increasing diversity in congestion control algorithms, researchers have observed that some algorithms are incompatible, in the sense that an algorithm running an aggressive protocol might empirically prevent another application from achieving even minimally acceptable performance. For example, TCP BBR has been shown to occupy a fixed proportion of the link capacity regardless of the number of competing NewReno flows [26], while NewReno itself has been shown to cause TCP Vegas to degrade to zero throughput [19, 21]. Such outcomes raise questions about whether the goals of these protocols are *fundamentally* incompatible, or whether there exist yet-undiscovered protocols that are both fair and achieve good performance that we can design with sufficient engineering effort. For example, are some latency-sensitive applications doomed to attain low throughput when competing with applications that prioritize throughput, or is it possible for both to coexist in a network?

In the absence of explicit statements about what objectives each protocol aims to optimize, however, the best we can do is reason about the behavior of these algorithms through measurement after deployment [25, 26]. Moreover, when algorithms do not make their own performance objectives explicit, studies of fairness must choose performance metrics (such as throughput or latency) to impose uniformly on the algorithms post-hoc, even if a particular algorithm or application does not prioritize that metric.

In this paper, we argue that we should reason directly about the safety of utilities that algorithms aim to optimize, rather than relying on experiments to validate the mechanistic design of algorithms post-hoc. While the prior literature on game-theoretic analysis of congestion control has largely

focused on whether Nash equilibria exist for a given class of protocols (often window-based increase-decrease protocols) [3, 7, 8, 24], we show in this paper that we can take a step further and determine analytically whether heterogeneous agents will reach *safe* equilibria, for rate-based protocols that aim to optimize local utilities. Fine-grained analysis of equilibria in heterogeneous settings is in general difficult, but we show that it is possible to lower bound equilibrium outcomes in the worst case, which is sufficient to show certain safety conditions. As a concrete example, we show in the context of rate-based protocols that we can define classes of utility functions such that, if all agents have a utility within the class, all agents will have nonzero throughput at the Nash equilibrium – a basic safety condition that is a precursor to more general statements about fairness. We additionally validate our theoretical results in a simulated network. The tools we present are a first step towards characterizing the safety of equilibria for a general class of practical utility functions.

Beyond the concrete guarantees this framework provides for understanding equilibria in networks, it opens the door to a number of possibilities for concretely regulating safety in networks. For example, once we have identified utility functions that are incompatible with each other, how should we translate that to network policy to prevent unsafe outcomes in practice? Can we use in-network mechanisms to isolate applications with incompatible utility functions, allowing them to coexist by changing the structure of the network? Can we enforce truthfulness in reporting utility functions before agents enter the network? Placing protocols in a common analytical framework decouples questions of policy and technical implementation, allowing the community to discuss issues of fairness in terms of utility functions while freeing engineers to design the best possible algorithms to optimize those functions.

In the remainder of the paper, we discuss the relationship of our work to existing work on utility-based congestion control and analyses of fairness and equilibria in congestion control (§2), the network model under which we derive our analytical results (§3), our analysis of equilibria (§4), provide an empirical validation of our results through simulation (§5), and discuss a number of directions for future study (§6).

2 Related Work

Utility-based congestion control. A line of recent work [4, 10, 11, 20, 28] explicitly states an end-user utility function and proposes protocols that attempt to optimize that function. The closest related work to ours is PCC-Vivace [11] and earlier work by Even-Dar et al [12], which use a regret minimization algorithm that can be shown to provably converge to an equilibrium. PCC-Vivace discusses the potential to centrally allocate bandwidth by adjusting a parameter in utility functions as a mechanism to control the bandwidth attained by each agent. Our work, in contrast, focuses on safety for heterogeneous agents who optimize their true utility rather than a centrally prescribed function. Remy [28] performs an offline optimization to attain an algorithm that performs well for a utility on average across a range of network parameters, while we are interested in equilibria in potentially adversarial environments. Copa [4] uses a utility formulation to determine a target equilibrium rate that the algorithm aims to attain when the utilities in a network are globally known.

Safety and fairness in congestion control. Ware et al [25, 26] frame competition between protocols in terms of the *harm* that a new protocol empirically causes an incumbent protocol on a diverse set of possible metrics, rather than focusing on achieving fair allocations. In a similar spirit, our work emphasizes bounding worst-case outcomes as a precursor to making finer-grained statements about fair allocations for different fairness notions.

In the context of loss-based windowing protocols, Zarchy et al [29, 30] prove a number of results about preferences that can coexist in a network. The goals of this work are similar to ours in providing provable guarantees on (or ruling out) certain performance outcomes for protocols, but our analysis abstracts from the mechanistic details of protocols, instead focusing on generalizing to protocols that optimize a utility function within a class.

A large literature has explored "friendliness" to TCP window increase-decrease protocols, which is one notion of safety with respect to legacy algorithms [13, 14, 22, 27]. Brown et al [5] argue that TCP-friendliness restricts innovation in congestion control algorithms in modern networks, supporting our choice in this work to abstract from algorithmic details and focus on the compatibiity of utility functions.

Perhaps the most well-known metric for *fairness* in congestion control is Jain's fairness index [16], which prioritizes equal allocations to all users. The more general metric of alpha-fairness [23] encompasses a number of alternative fairness measures, including proportional fairness, max-min fairness, and minimum potential delay fairness. Extending our techniques beyond basic safety criteria to characterize the quality of equilibria with respect to some of these fairness measures is an important direction for future work.

Game theoretic equilibrium analysis. Game-theoretic analyses have been used previously to analyze and argue about the behavior of many congestion control algorithms, particularly TCP variants [3, 8]. These analyses, however, are limited to "legacy" window-based TCP increase-decrease protocols. As the world of congestion control algorithms expands beyond TCP, these specialized analyses cannot describe the interactions between TCP and newer protocols.

Other classic work that models the congestion control problem as a game [18] aims to design a centralized allocation mechanism to optimize for a global objective. Johari and Tsitsiklis [17] study the price of anarchy in Kelly's model. Another line of work [7, 24] models a general class of protocols that use functions that describe rate increments and decrements to modulate their rate, and show the existence and uniqueness of Nash equilibria in their setting, but do not extend to analyzing the fairness properties of those equilibria.

3 Network Model

In this section, we set up a simple network model with a single bottleneck link, and in Section 4 we will demonstrate how to prove safety properties in this model. While the model is simple, our goal is to give an example of how one can prove such safety properties within a theoretical framework, and safety in a simple model is a prerequisite to safety in a more complex network.

3.1 Preliminaries

We assume multiple agents sharing a single bottleneck link with capacity C packets per time step. At each time step t, each agent i chooses a sending rate $x_i \in \mathbb{R}_+$ packets per time step at which to send packets. The total load on the link is then $B = \sum_i x_i$. Capacity on the link is allocated by a router that allocates bandwidth proportionally to the agents' rates. In particular, the realized bandwidth for each agent is $\min(C, B)\frac{x_i}{B}$ Intuitively, when the total load is below C, agents receive bandwidth equal to their input rate, and when the load is above C, the router allocates the total bandwidth in proportion to the input rates.

If the load is above C, the excess packets $(B - C)$ are stored in a queue, leading to delay. The queue drains at C packets per time step, and therefore the additional load results in $\frac{B-C}{C}$ additional time steps of *queueing delay*. To simplify the model, we assume that the penalty for queueing delay only applies to the time step t at which it was incurred, and the queue resets to zero at the next time step. (Other work that analyzes a similar model, such as PCC-Vivace [11], addresses the issue of cumulative queues by considering only the change in queueing delay at time t; we assume the queue drains to simplify the analysis, and leave a comparison of these two models to future work.)

3.2 Utility Model

We analyze a commonly used utility function [4, 11, 12, 28] that trades off delay and throughput:

$$u_i(x_i, \sum_{j \neq i} x_j) = \min(C, B)\frac{x_i}{B} - \alpha_i x_i g\left(\left[\frac{B-C}{C}\right]_+\right) \quad (1)$$

where $B = \sum_i x_i$.

The agent receives a reward for the allocated bandwidth, and a penalty that is a convex function g of the delay, weighted by α_i. This utility assumes a rate-based protocol that does not take loss into account to simplify the exposition, but loss can be incorporated as a linear term so that the assumptions of our analysis still hold (e.g., as in PCC-Vivace [11]).

The parameter α_i can be thought of as an individual's sensitivity to delay; higher α_i indicates a latency-sensitive application such as a video call, while lower α_i penalizes delay less, as in a bulk transfer application like SCP.

For our later analysis, we additionally require g and g' to be strictly monotone increasing, a condition satisfied by, for example, $g(z) = z^2$.[1]

4 Utility-Based Safety Analysis

In this section, we give an example of a safety condition that we can derive using the utility framework for the network model described in Section 3. We first present the safety guarantee, then instantiate it with a concrete example of real applications in Section 4.1 and sketch the proof in Section 4.2.

In particular, given a fixed delay sensitivity $\alpha_{\min}$ for the *least* delay-sensitive agent (i.e., the most aggressive agent) in the network, we can derive a sensitivity $\alpha_{\max}$ for the *most* delay-sensitive agent such that this agent is guaranteed throughput greater than zero at equilibrium, even when other agents with $\alpha_{\min} \leq \alpha_i \leq \alpha_{\max}$ are also in the network.

While analyzing heterogeneous equilibria can be difficult in general, we show that for this congestion control game, we can derive an upper bound on $\alpha_{\max}$ through a comparison to a "worst-case" *symmetric* (i.e., homogeneous) equilibrium in which all agents have a delay coefficient of $\alpha_{\min}$. We refer to this game as "worst-case" because, as we will show in Lemma 2, the aggregate rates of players at equilibrium in this game is an upper bound on the aggregate at equilibrium in the asymmetric game.

In particular, consider an asymmetric game with n agents with delay coefficients $\alpha_{\min} = \alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n = \alpha_{\max}$ and the corresponding "worst-case" symmetric game with n agents whose delay coefficients are all equal to $\alpha_{\min}$. Let B_{sym} be the *sum* of the rates of the agents in the worst-case game at equilibrium (which will be at least C).

Proposition 1. *Let B_{sym} correspond to the equilibrium total rate in the symmetric game described above. If the delay coefficient $\alpha_{\max}$ of the weakest agent satisfies the condition*

$$\alpha_{\max} < \frac{C}{B_{sym} g\left(\frac{B_{sym}-C}{C}\right)} \quad (2)$$

[1] Prior work [11, 12] implicitly requires g to be close to linear in order to satisfy a technical requirement called "social concavity," which we relax, allowing us to model latency-sensitive applications whose utility decreases rapidly as delays increase.

	Delay	Bandwidth	Fitted α_i
Skype voice call	400 ms	2 Mbps	1.245
Google Meet	100 ms	1 Mbps	10
File transfer	1000 ms	8.8 Mbps	0.879

Table 1: Delay and bandwidth requirements for videoconferencing and file transfer applications.

then the rate x_i at the Nash equilibrium will be strictly positive. Moreover, this result also holds for any number of players smaller than n.

For example, we can instantiate this bound concretely for the function $g(z) = z^2$, which can describe applications whose utility decays rapidly as the latency increases:

Corollary 1. *Under the conditions of Proposition 1 and $g(z) = z^2$,*

$$\alpha_{\max} < \frac{C^3}{B_{sym}(B_{sym} - C)^2}$$

ensures $x_i > 0$ for all i, i.e. all agents send at positive rates at equilibrium.

4.1 Example

To demonstrate how these bounds translate to practice, we give a small example of three applications running in a network: a bulk transfer application and two videoconferencing applications. We use the published connection requirements for these applications to fit values of α_i and then use our theoretical bound to determine whether the applications can coexist safely in a network.

We use minimum connection requirements from Skype [2] and Google Meet (enterprise videoconferencing) [1] to fit delay coefficients α to utility functions. Skype does not provide latency requirements, so we use latency recommendations from the ITU [15] for voice calls. For file transfers, we approximate the minimum bandwidth requirement as the bandwidth at which a 1 GB file would take 15 minutes to download, around 8.8 Mbps, and a maximum latency of 1 second. The delay and bandwidth requirements for these applications are listed in Table 1. We assume a 10 Mbps connection and an MTU size of 1500 bytes to convert the speeds into packets per second.

For each of these applications, we fit a value of α_i by assuming that the minimum requirements correspond to zero utility. For minimum bandwidth b packets/second and delay d seconds, we take $x_i(\frac{b}{x_i} - \alpha_i d^2) \geq x_i(\frac{b}{C} - \alpha_i d^2) \geq 0$, $\alpha_i \geq \frac{b}{Cd^2}$ and solve for a lower bound on the corresponding α_i. Table 1 lists the fitted values of α_i for each application.

For these fitted values of α, we have $\alpha_{\min} = 0.879$ (corresponding to the least sensitive agent), and the corresponding B_{sym} (computed by simulation for 3 agents) is 1169.12. In this case, our theoretical bound gives $\alpha_{\max} = 4.375$. Thus, Google Meet is too delay-sensitive to coexist with Skype and a file transfer in our model, but the latter two applications will each receive nonzero throughput at equilibrium.

4.2 Proof Sketch

We sketch the proof of Proposition 1 by establishing two lemmas. These lemmas are general properties of our congestion control game that may be independently useful as tools for future analysis. The goal is to establish monotonicity properties on the equilibrium rates so that we can show that B_{sym} upper bounds the worst-case aggregate that an agent might have to respond to.

Consider the *best response* function $r_i(\sum_{j\neq i} x_j)$ for an agent i, i.e. the utility-maximizing rate when the aggregate rate of the other players is fixed at $\sum_{j\neq i} x_j$. The first structure we exploit is that our congestion control game turns out to be a game of *strategic substitutes*; that is, when $\sum_{j\neq i} x_j$ increases, $r_i(\sum_{j\neq i} x_j)$ decreases. We show this in the following lemma.

Lemma 1. *Consider the best response function r_i for some agent i in the congestion control game as defined in Section 3.2 and two possible rate aggregates $\sum_{j\neq i} x_j$ and $\sum_{j\neq i} y_j$ for the remaining players. If $\sum_{j\neq i} x_j < \sum_{j\neq i} y_j$ then $r_i(\sum_{j\neq i} x_j) \geq r_i(\sum_{j\neq i} y_j)$.*

Next, we use the aggregative structure of the game to show that the sum of equilibrium rates at any α is upper bounded by the *symmetric* equilibrium corresponding to the least delay-sensitive agent. Intuitively, this symmetric game is the "most competitive" case and upper bounds the queueing delay.

Lemma 2. *Consider a game with strictly monotone increasing g and g', with coefficients α, and let $\alpha_{\min} = \min_i \alpha_i$. Then consider the symmetric game where $\alpha_i = \alpha_{\min}$ for every i. Denote the equilibrium total rate for the asymmetric game by B and the equilibrium total for the symmetric game by B_{sym}. Then $B \leq B_{sym}$.*

The first lemma controls the behavior of any agent in response to changes in the aggregate, while the second lemma bounds the aggregate rate for any symmetric game. Combining the two allows us to lower bound the rate of any agent i at equilibrium for a game with coefficients α.

Consider the least delay sensitive agent, $\alpha_{\min} = \min_i \alpha_i$. Let B_{sym} be the sum of rates at equilibrium for a symmetric game of n agents where $\alpha_i = \alpha_{\min}$ for every i.

Then by Lemma 2, we know that $B(\alpha) < B_{sym}$. Moreover, we have

$$\sum_{j\neq i} x_j \leq x_i + \sum_{j\neq i} x_j = B(\alpha) < B_{sym}.$$

Hence, we also know that $\sum_{j\neq i} x_j < B_{sym}$ at α, and thus by Lemma 1 we have $r_i(\sum_{j\neq i} x_j) > r_i(B_{sym})$. Thus, the best response rate is lower bounded by the best response to B_{sym},

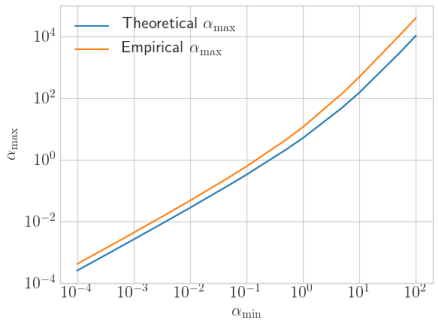

Figure 1: Comparison of theoretical upper bound on safe $\alpha_{\max}$ values with empirical values found by search for $n = 3$ agents.

and the safety condition in Proposition 1 follows by solving for values of α for which this lower bound is 0.

5 Experimental Validation

In this section, we validate our theoretical bound in simulation, and show that the bound gives an accurate and useful guarantee on outcomes in practice.

While our bound provides guarantees that a particular α_{max} results in a nonzero rate at equilibirum, this guarantee can be pessimistic and even larger α may result in nonzero rates in practice. If the bound is too loose, we may be disallowing more delay-sensitive agents that could otherwise be competitive in the network. We explore the tightness of the bound for quadratic $g(z) = z^2$.

We run a network simulator that implements the network model described in Section 3. The simulated agents take gradient descent steps at every time step.[2] We run the simulation until the partial derivatives of all agents' utilities are less than $\varepsilon = 0.001$. The gradient descent step size for all agents is 0.001. The simulator bandwidth is $C = 100$ packets per time step, and agents' initial rates are random integer values in $[0, C]$.

This experiment considers $n = 3$ agents in which one agent is weak (with delay sensitivity α_{max}) and two are strong (α_{min}). We sweep α_{min} values between 0.001 and 100 and find the corresponding total rate at the symmetric equilibrium, $B_{\max}$. We then use our theoretical condition to solve for a safe $\alpha_{\min}$ value. However, since this value is conservative, we continue to brute-force search for values of $\alpha > \alpha_{\max}$ such that the weak agent continues to receive nonzero throughput when the agents' rates have converged.

In Figure 1 we plot our bound against the value of $\alpha_{\max}$ found by brute-force search. The bound is tighter for smaller values of $\alpha_{\min}$ (when the strong agents are stronger) and looser at larger values (when all agents are more sensitive). In all cases, the bound is no more than a factor of 4 away from the empirical value. This suggests that while our bound can be somewhat pessimistic when all agents are highly delay-sensitive, it provides useful guarantees particularly for realistic settings such as the one we explored in Section 4.1.

[2]While we do not have a proof of convergence to Nash equilibrium for our game, we confirmed by brute-force search over the space of rates that when gradient descent converges, as it does empirically in our experiments, no unilateral deviation improves any player's utility.

6 Discussion and Open Questions

In this paper, we presented one example of an analytical bound that can be used to ensure that heterogeneous protocols can coexist safely in a network together, under certain assumptions on the utility functions. Such guarantees can enable network designers to reason about protocol safety at a level of abstraction even if the underlying, mechanistic details of the implementations change over time. In this section, we discuss a number of conceptual questions about network design and regulation that are raised by our model, as well as several directions to extend the model to encompass a larger set of realistic network scenarios.

6.1 Regulating Unsafe Utilities

Our framework provides a way to determine classes of utility functions such that no algorithm with a utility in the safe class will receive nonzero throughput. This raises a natural question: if an agent's true utility falls outside of the class, what decisions should network designers, ISPs, or the agents themselves take next? If a utility is too aggressive, an application with that utility may need to make compromises in order to be safe in the network; if it is too timid, an application may not be able to use the network at all without structural changes to the network. We discuss some possibilities below.

Approximating safe utilities. If an agent's true utility results in aggressive behavior, an ISP may require them to compromise in order to allow more delay-sensitive applications to run safely in the network by optimizing the nearest utility that does fall in the safe class. In this case, it is important to understand how much this approximation degrades outcomes for the aggressive agent: is it worthwhile to continue using the network at all, or does it render applications unusable? Negotiating such compromises may be critical to making the network usable for a large class of applications.

In-network regulation. Is it possible to run applications simultaneously in a network if our analysis determines that they are not compatible? One option in such a scenario is to use in-network mechanisms to isolate safe classes of utilities, allocating a subset of bandwidth to each class. One such mechanism is proposed by Brown et al in [5], in the context of bandwidth allocation for individual flows; this mechanism could be extended to proportional allocation for classes of flows rather than individual flows.

Identifying unsafe behavior. Even if agents declare *a priori* that their true utility falls within the safe class, a malicious agent may lie about their utility and act more aggressively at runtime. Can a central authority compute the expected equilibrium and monitor network traffic to ensure that agents actually adhere to their declared utility? We conjecture that the answer in our model is yes, as there exist well-defined algorithms for (asymptotically) optimal play in the utility model we have analyzed [12], so the goal of a network monitor would be to ensure that all agents approximately adhere to the rates predicted by optimal play. Nevertheless, there are clearly numerous technical hurdles to overcome in implementing such a monitoring system in practice.

6.2 Extending the Model

While the utility class we analyze is flexible enough to model a varied set of application preferences, in this section we discuss several technical extensions to our model that would make it more useful in practice.

Concave utilities. Our current analysis only supports a restricted set of utility functions that are concave (which we use to show monotonicity properties on equilibrium rates)[3]. This model makes sense for latency-sensitive applications whose utility rapidly degrades as the latency increases, and can also model bulk transfers that are insensitive to the amount of queueing delay incurred relative to the throughput obtained. However, it would be important to understand whether the analysis can be extended to non-concave utilities.

Modeling timid agents. In our model, if aggressive agents are using the link, highly delay-sensitive agents may not be able to obtain any throughput if they optimize their true utility. A more accurate utility model might take the agent's own throughput into account in the delay penalty – for example, a penalty that decreases as the throughput decreases to encourage competitive behavior. Designing and analyzing the equilibrium behavior of concave utility functions with this property remains a challenge for future work.

Complex network topologies. Our model studies a single bottleneck link, which can identify unsafe behavior in one simplified scenario (a prerequisite to being safe in more complex settings). However, our game can be extended to more complex settings, in which flows may traverse different routes with different bottlenecks. An outstanding question is whether we can use similar techniques to analyze the safety of the equilibria reached in these more complex topologies (or study even more basic properties, such as whether these equilibria exist at all and whether they are unique).

[3]In particular, we can extend our analysis to a more general class of utilities that satisfy a "strong concavity" assumption [9].

Stackelberg equilibria. Our analysis focuses on Nash equilibria, in which all agents play simultaneously and arrive at an equilibrium from which no agent can unilaterally improve. Real-world networking settings might be better modeled by Stackelberg games, in which a leading agent chooses its rate first independently of the play of the remaining players. (For example, BBR has been shown to occupy a fixed 40% share of the link when competing with TCP CUBIC, regardless of the number of competing flows [26]). If the play of the leading agent is known in advance, it may be possible to determine the class of safe utilities using our current analysis. On the other hand, if the play of the leading agent is unknown, it could mean that many more utilities are "unsafe" in the sense that those agents will obtain zero throughput or poor utility. Consider an incumbent agent that occupes the entire bottleneck bandwidth C – any newly arriving agents are bound to incur high delay.

7 Conclusion

In this paper, we described a framework through which to understand safety for modern congestion control protocols that aim to optimize a diverse array of utility functions. We advocate for an approach that directly reasons about the safety of the utilities themselves when agents play optimally with respect to those utilities, rather than designing protocols and understanding their behavior after deployment. In our work, we demonstrated an example of this type of analysis in the context of rate-based protocols, showing that we can describe classes of utility functions such that all agents receive nonzero throughput at the Nash equilibrium. This preliminary work opens a number of directions for future work in understanding how such safety concepts can be translated into networking practice.

Acknowledgments

Thanks to Rad Niazadeh and Bruce Spang for early conversations related to this work. We also thank Scott Shenker for suggesting the extension to Stackelberg games. Daniel Kang, Deepti Raghavan, and Sahaana Suri provided feedback that improved the writing. This research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Ant Financial, Facebook, Google, and VMware—as well as Toyota Research Institute, Cisco, SAP, and the NSF under CAREER grant CNS-1651570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] 2020. Google Meet hardware requirements. https://support.google.com/meethardware/answer/4541234?hl=en#zippy=%2Cminimum-bandwidth-required. (2020).

[2] 2020. How much bandwidth does Skype need? https://support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need. (2020).

[3] Aditya Akella, Srinivasan Seshan, Richard Karp, Scott Shenker, and Christos Papadimitriou. 2002. Selfish behavior and stability of the Internet: A game-theoretic analysis of TCP. *ACM SIGCOMM Computer Communication Review* 32, 4 (2002), 117–130.

[4] Venkat Arun and Hari Balakrishnan. 2018. Copa: Practical Delay-Based Congestion Control for the Internet. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. USENIX Association.

[5] Lloyd Brown, Ganesh Ananthanarayanan, Ethan Katz-Bassett, Arvind Krishnamurthy, Sylvia Ratnasamy, Michael Schapira, and Scott Shenker. 2020. On the Future of Congestion Control for the Public Internet. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*. 30–37.

[6] Neal Cardwell, Yuchung Cheng, C Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2017. BBR: congestion-based congestion control. *Commun. ACM* 60, 2 (2017), 58–66.

[7] Mung Chiang, SH Low, D Wei, and Ao Tang. 2006. Heterogeneous congestion control: Efficiency, fairness and design. In *Proceedings of the 2006 IEEE International Conference on Network Protocols*. IEEE, 127–136.

[8] Dah-Ming Chiu and Raj Jain. 1989. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN systems* 17, 1 (1989), 1–14.

[9] Luis C Corchón. 1994. Comparative statics for aggregative games: the strong concavity case. *Mathematical Social Sciences* 28, 3 (1994), 151–165.

[10] Mo Dong, Qingxi Li, Doron Zarchy, Philip Brighten Godfrey, and Michael Schapira. 2015. PCC: Re-architecting Congestion Control for Consistent High Performance.. In *NSDI*, Vol. 1. 2.

[11] Mo Dong, Tong Meng, Doron Zarchy, Engin Arslan, Yossi Gilad, Brighten Godfrey, and Michael Schapira. 2018. PCC Vivace: Online-learning congestion control. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. 343–356.

[12] Eyal Even-Dar, Yishay Mansour, and Uri Nadav. 2009. On the convergence of regret minimization dynamics in concave games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 523–532.

[13] Sally Floyd and Kevin Fall. 1999. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on networking* 7, 4 (1999), 458–472.

[14] S. Ha, I. Rhee, and L. Xu. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review* 42, 5 (2008), 64–74.

[15] ITU. 2003. *ITU-T Recommendation G.114: One-way transmission time*. Technical Report.

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

[17] Ramesh Johari and John N Tsitsiklis. 2004. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research* 29, 3 (2004), 407–435.

[18] F. P. Kelly, A. Maulloo, and D. Tan. 1998. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* (1998), 237–252.

[19] Kenji Kurata, Go Hasegawa, and Masayuki Murata. 2000. Fairness Comparisons Between TCP Reno and TCP Vegas for Future Deployment of TCP Vegas. https://web.archive.org/web/20160103040648/http://www.isoc.org/inet2000/cdproceedings/2d/2d_2.htm. In *Internet Society Conference*.

[20] Tong Meng, Neta Rozen Schiff, P Brighten Godfrey, and Michael Schapira. 2020. PCC Proteus: Scavenger Transport And Beyond. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 615–631.

[21] Jeonghoon Mo, Richard J La, Venkat Anantharam, and Jean Walrand. 1999. Analysis and comparison of TCP Reno and Vegas. In *IEEE INFOCOM*, Vol. 3. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1556–1563.

[22] Jitendra Padhye, Jim Kurose, Don Towsley, and Rajeev Koodli. 1999. A model based TCP-friendly rate control protocol. In *Proceedings of NOSSDAVâĂŹ99*. Citeseer.

[23] Rayadurgam Srikant. 2004. *The mathematics of Internet congestion control*. Springer Science & Business Media.

[24] Ao Tang, Jiantao Wang, Steven H Low, and Mung Chiang. 2007. Equilibrium of heterogeneous congestion control: Existence and uniqueness. *IEEE/ACM Transactions on Networking* 15, 4 (2007), 824–837.

[25] 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*. 17–24.

[26] 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.

[27] Jörg Widmer, Robert Denda, and Martin Mauve. 2001. A survey on TCP-friendly congestion control. *IEEE network* 15, 3 (2001), 28–37.

[28] Keith Winstein and Hari Balakrishnan. 2013. TCP ex machina: computer-generated congestion control. In *ACM SIGCOMM Computer Communication Review*, Vol. 43. ACM, 123–134.

[29] Doron Zarchy, Radhika Mittal, Michael Schapira, and Scott Shenker. 2017. An axiomatic approach to congestion control. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*. 115–121.

[30] Doron Zarchy, Radhika Mittal, Michael Schapira, and Scott Shenker. 2019. Axiomatizing congestion control. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 2 (2019), 1–33.