Learning and utility in multi-agent congestion control

Pratiksha Thaker **Matei Zaharia** **Tatsunori Hashimoto**
Department of Computer Science
Stanford University
{prthaker,matei,thashim}@cs.stanford.edu

Abstract

We study the setting of congestion control in computer networks, in which agents compete for a share of bandwidth on a network link. Prior work studying equilibrium behavior in this setting focuses on *socially concave* utilities, a condition that ensures that distributed regret minimization will converge to a Nash equilibrium. In this work, we show that while social concavity is a convenient condition, many realistic utilities for networking applications do not fall into the class of socially concave utilities. We make progress on studying the behavior of more realistic utility functions that are not socially concave, and show through simulations that they can exhibit favorable convergence behavior, which suggests that directly optimizing for a more realistic class of utilities may be tractable in practice.

1 Introduction

In order to transfer data on a network, computers make use of a shared resource: a network link. These computers must make distributed decisions about how much data to send at a time to avoid congestion on the link. In the computer networking literature, this problem is referred to as *congestion control* and has received a great deal of attention over the last few decades, including classic algorithms such as TCP NewReno [16] and TCP Vegas [6] as well as recent work founded in learning and optimization [24, 10, 11, 18].

It is natural to view the problem of congestion control as a resource-allocation game where each user obtains utility according to quality of service measures such as throughput and latency. While most proposed congestion-control algorithms do not explicitly optimize for the utility obtained by each user, the designers of each algorithm implicitly make such arguments by arguing that the algorithm improves the network for end users by improving an aspect of quality of service such as throughput. For example, researchers at Google recently proposed BBR [7], an algorithm that attempts to actively "probe" for available bandwidth in order to maximize link utilization.

A major drawback of most of these congestion control algorithms is that they rely primarily on empirical evaluations and only provide heuristic justifications for improving overall quality of service. As a result, it is challenging to provide provable guarantees about the utilities of individual users when protocols reach a steady-state equilibrium, or about whether they reach an equilibrium at all. While the networking community has made progress towards analyzing the theoretical behavior of classic algorithms such as TCP NewReno [16], this theory does not exist for newer algorithms such as Google's BBR.

These concerns highlight the need for an analytical framework to understand the equilibrium behavior of congestion-control algorithms. The framework of regret minimization in online learning provides one lens through which to understand these dynamics. In particular, we can understand the behavior of an algorithm that plays (in a particular sense) "optimally" with respect to its utility, and then ask questions about the behavior of multiple agents with symmetric or asymmetric utilities.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

In this paper, we build on recent work [11, 13] that places congestion-control algorithms in the framework of game theory and online learning. These papers show that assuming a particular functional form for utility, regret-minimizing senders will attain a Nash equilibrium with low latency.

Inspired by this work, we use a game-theoretic framework to address broader questions about the equilibrium behavior of congestion-control algorithms. Prior work constrains utility functions to be *socially concave*, a convenient condition that implies convergence to a Nash equilibrium under distributed regret minimization dynamics. Unfortunately, we show in our work that the social concavity condition often does not reflect real-world networking utilities, particularly for interactive applications that are sensitive to delay: an important class of applications that includes voice calls, videoconferencing, and online gaming. We then demonstrate that some utilities that do not satisfy social concavity (and are therefore difficult to understand analytically) nevertheless can have analytically tractable equilibria structure and show promising convergence behavior in simulation. This leads us to conjecture that a broader and more realistic class of utilities is tractable in practice.

The rest of the paper is organized as follows. In Section 2 we review the congestion-control game, the class of utility functions we are concerned with, and define social concavity. In Section 3 we examine the real-world degradation in utility as a function of network delay, and in Section 4 we show that socially concave utility functions cannot adequately model these real-world utilities. We also analyze the equilibria of more general utilities and obtain simple conditions for unique symmetric equilibria when utilities are symmetric. Finally, in Section 5 we study the behavior of more realistic utility functions in simulation and show that agents running local gradient descent with these utilities show promising convergence behavior.

Societal impact and ethics. We examine questions about the equilibrium behavior of congestion-control algorithms, which can help us understand whether new algorithms with differing objectives will behave well or unfairly with respect to their competitors in a network. This is especially relevant when large entities have a dominant role in deciding which protocols run in the Internet, and may or may not have goals that align well with other users of this shared resource. This work is a step toward understanding whether we expect a new algorithm to be "fair" *before* it is deployed in the real Internet.

2 Model and Definitions

We use a network model similar to the one presented in Dong et al. [11] and Even-Dar et al. [13].

Network model. We assume multiple agents sharing a single bottleneck link with capacity C packets per time step. Time is divided into discrete chunks. At each time step t, each agent i chooses a sending rate $x_i^t \in \mathbb{R}_+$ packets per time step at which to send packets. The total load on the link is then $\sum_i x_i^t$. 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 $\min(C, \sum_j x_j^t)\frac{x_i^t}{\sum_i x_i^t}$ 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, $\sum_j x_j - C$ packets are queued in one time step. The queue drains at C packets per time step, and therefore the additional load results in $\frac{\sum_j x_j - C}{C}$ 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 0 at the next time step.

We follow the model of [11] in which agents experience a penalty for packet losses according to $1 - \frac{C}{\sum_i x_i}$ (that is, every agent incurs a penalty that is a fraction of the total excess rate).

Utility model. We assume agents' utilities take the following form:

$$u_i(x) = \min(C, \sum_j x_j)\frac{x_i}{\sum_j x_j} - \alpha_i x_i g\left(\left[\frac{\sum_j x_j - C}{C}\right]_+\right) - \beta_i x_i \left(\max\left(0, 1 - \frac{C}{\sum_j x_j}\right)\right) \tag{1}$$

In words, the agent receives a reward for the allocated bandwidth, a penalty weighted by α_i for the incurred queueing delay, and a penalty weighted by β_i for any packets lost. This is similar to the utility function used by [11], with two exceptions. First, the agent is rewarded for the *allocated* bandwidth rather than the sending rate. Second – importantly – the penalty on delay g is a linear function in [11], while we consider nonlinear functions in this work.

Social concavity. Even-Dar et al. [13] introduce the notion of *social concavity*. Intuitively, in the same way that convex zero-sum games are analytically tractable via the convex-concave structure of the players' utilities (e.g. [23]), social concavity requires utility functions that are concave in one player and convex in the rest (while the game may be general-sum). Restricting the game to be socially concave is conceptually appealing as it ensures that distributed regret minimization will reach a Nash equilibrium [13]. For a game to be socially concave, the agents' utilities must satisfy two conditions:

1. There exists a convex combination of the players' utilities that is concave in x_i, and
2. Each utility is convex in the rates of the remaining players, x_{-i}.

The utility function of [11] (as well as the modified utility in Equation 1), where the delay penalty g is *linear*, is socially concave. However, as we describe in the next section, a linear delay penalty is not a good fit for real-world user experience and we will show that more realistic and delay-sensitive utilities violate social concavity in Section 4.

3 Utility functions in networking

The utility in Equation 1 can capture a wide range of user preferences ranging from bulk file transfer to voice calls by varying delay sensitivity (α, g) and loss sensitivity β. In this work, we will more carefully investigate delay-sensitive applications such as voice calls, videoconferencing, and online gaming. In each of these applications, the delay-sensitivity is nonlinear since users experience rapid degradation in usability above above small but acceptable levels of delay. For example, Figure 1 in [2] (replicated in Figure 1) suggests that for voice calls, a delay below 200 ms does not affect utility, but utility decays rapidly above 200 ms. (Note that Figure 1 represents the positive reward from a voice call at a given delay while g is a *penalty* for additional delay.) Similarly, Figure 1 in [8] suggests that utility drops sharply above a delay of about 150 ms in online gaming.

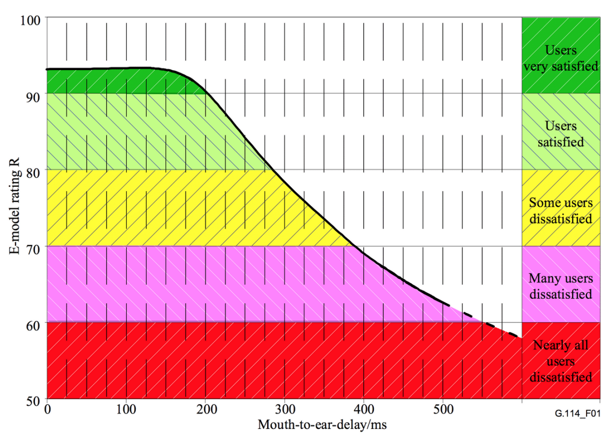


Figure 1: Reproduction of Figure 1 in [2]. The y-axis plots the E-model rating, a quality of service metric published by the International Telecommunications Union [1].

In terms of Equation 1, these studies suggest two constraints on our utility. First, the delay penalty g should be monotonically increasing and sharply convex around a round-trip delay of 150-200 ms (a one-way delay of about 100 ms).[1] Second, g may eventually become concave at higher delays where

[1]Note that this includes both delays from queueing as well as the base propagation delay to transport a packet to and from its destination. The one-way propagation delay between the United States east and west coasts is about 40 ms.

the system is uniformly unusable. As we show in the next section, utilities of this form do not fit the constraints of social concavity.

4 Limitations of social concavity

In this section, we find conditions on g under which the utility in Equation 1 is socially concave.

In order to fulfill the conditions of social concavity, the utility must be concave in x_i and convex in x_{-i}, the action of all other players. Below, we show that fulfilling these conditions requires g to be "close to" linear in delay. This is intuitively unrealistic: a video-conferencing application should strongly prefer smaller delays, with the penalty growing as delay increases.

Proposition 1. *For a game with utilities of the form in Equation 1 to be socially concave, the following condition on the delay penalty g must hold for all* $\{x_1 \dots x_n\} \geq 0$,

$$-\frac{2C}{x_i} g'\left(\frac{\sum_j x_j - C}{C}\right) + \frac{2C^3(1+\beta_i)}{\alpha_i x_i} \frac{x_i - \sum_j x_j}{(\sum_j x_j)^3} \leq g''\left(\frac{\sum_j x_j - C}{C}\right) \leq \frac{(2C^3)(1+\beta_i)}{\alpha_i} \frac{1}{(\sum_j x_j)^3}.$$

Corollary 1. *For a game with utilities of the form in Equation 1 to be socially concave, g must be asymptotically linear in delay.*

This follows immediately from observing that if $\sum_j x_j \to \infty$, both bounds go to 0. Hence in the limit, we have $0 \leq g'' \leq 0$, so g must be (asymptotically) linear in order to satisfy social concavity.

We note that g can be convex for small values of $\sum_i x_i$ without violating social concavity. In particular, delays are nonzero when $\sum_j x_j \geq C$. In this regime, the bound simplifies to

$$g''\left(\frac{\sum_j x_j - C}{C}\right) \leq \frac{2(1+\beta_i)}{\alpha_i}.$$

[11] uses values of $\alpha_i = 900$ and $\beta_i = 11.35$ for their experimental evaluation, for which this threshold evaluates to 0.027. That is, even for small delays, the social concavity condition requires near-linearity in the delay penalty.

4.1 Equilibria beyond social concavity

We have demonstrated that social concavity is violated in realistic networking utilities. Because prior work relies on social concavity to tractably analyze equilibria, this might suggest that it may be intractable to characterize the behavior of network agents with more realistic utilities.

In this section, we show that this is not the case by proving an initial result in a setting with more general utilities. In the case of symmetric utilities (i.e. when all users share the same delay and loss sensitivity), we can show that more complex networking utilities will still have a single unique symmetric Nash equilibrium. Demonstrating the uniqueness of the symmetric equilibrium is a step towards ensuring that the equilibria reached by agents with general utilities are *fair*: if an asymmetric equilibrium exists, there is potential for agents to end up in an unfair equilibrium state. We show that a generic monotonicity condition excludes asymmetric equilibria.

Proposition 2. *Let* $\alpha := \alpha_1 = \ldots = \alpha_n$ *and* $\beta := \beta_1 = \ldots = \beta_n$*. If*

$$0 < \alpha g'\left(\frac{\sum_j x_j - C}{C}\right)$$

then there are no asymmetric equilibria.

We can strengthen this result further and demonstrate that there is only a single unique symmetric equilibrium for delay sensitivities that are sufficiently convex.

Proposition 3. *Under the condition in Proposition 2, if additionally for all* x,

$$\frac{2+2\beta C}{C} < \alpha_i g''\left(\frac{nx - C}{C}\right)$$

then there is a unique symmetric Nash equilibrium for the n*-player symmetric game.*

4.2 Implications

Consider the utility functions discussed in Section 3. These are highly nonlinear and the penalty for delay is convex, increasing sharply when delays grow beyond a small threshold. As we have shown, social concavity requires nearly linear penalties for delay, so a game with real-world utilities for these interactive applications would not be amenable to analysis via the theory of socially concave games. However, as we showed in Section 4.1, as long as the delay is monotone increasing (an intuitive requirement) and its second derivative is larger than a small constant, a game with symmetric utilities will have a unique symmetric Nash equilibrium. This suggests that a more general class of utilities can be well-behaved in the congestion control setting.

5 Utilities for congestion control

In the previous section, we made steps towards analytically characterizing utilities that better model real-world user preferences. In this section, we posit concrete utility functions and evaluate their behavior in simulation, both in the symmetric case that we can characterize as well as an asymmetric setting. Although these utility functions do not lend themselves to straightforward analysis using standard techniques, simulations show that they behave predictably, suggesting that a broader class of utilities may be safe in practice.

5.1 Modified utility functions

We investigate utility functions that set $g(z) = z^2, g(z) = z^4$, and $g(z) = \log(1+z)$. For example, in the case of $g(z) = z^2$, the utility in Equation 1 becomes

$$u_i(x) = \min(C, \sum_j x_j) \frac{x_i}{\sum_j x_j} - \alpha_i x_i \left(\frac{\sum_j x_j - C}{C} \right)^2 - \beta_i x_i \left(1 - \frac{C}{\sum_j x_j} \right) \tag{2}$$

This function is not socially concave because g is convex with a second derivative of 2, violating the constraints in Section 4. However, it satisfies the unique Nash conditions for symmetric equilibria and more closely matches the real-world degradation in user experience described in Section 3.

5.2 Simulated behavior

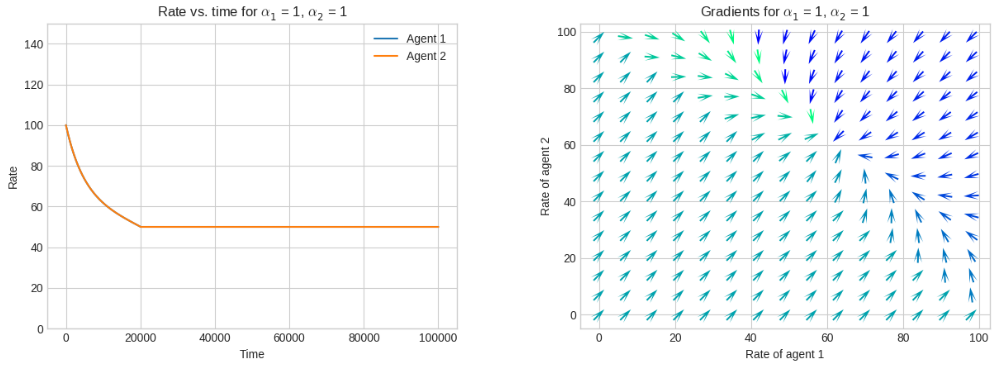
In order to understand the equilibrium behavior of regret-minimizing agents computationally, we simulate agents performing gradient descent on their local utilities in a manner similar to the regret-minimizing gradient descent algorithm of [27]. While we cannot analytically reason about equilibria in asymmetric utility settings, this computational approach can help us understand whether multiple equilibria are likely to occur and where.

We first simulate two agents running gradient descent using the above utility function for 100000 time steps in a network with a capacity of 100 packets per time step using the function $g(z) = z^2$. In all experiments, β is set to 0.1 (which differs from the recommended value in the evaluation of [11] because the reward in our utility is the allocated bandwidth rather than the input rate). We run one experiment in the symmetric case, where both agents have a weight of $\alpha = 1$ on delay, and another experiment in the asymmetric case, where $\alpha_1 = 10$ and $\alpha_2 = 1$. For both cases, we also plot the vector field of gradients for each pair of rates.

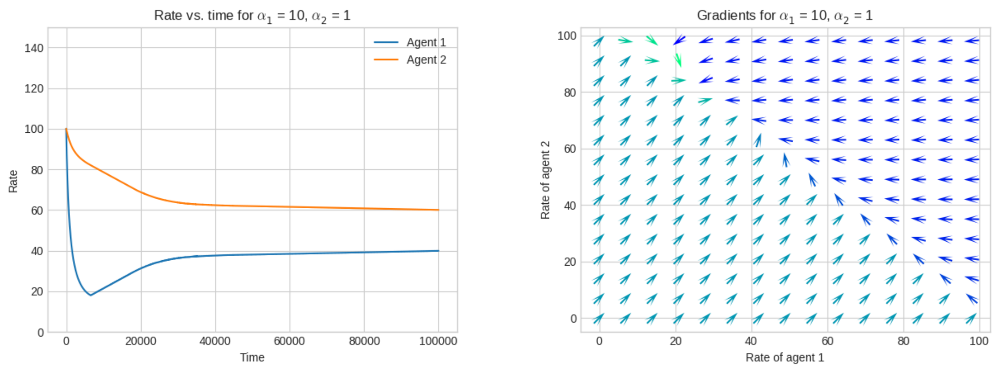
The results are in Figure 2. In the symmetric case (Figure 2a), our experiments show that there exists one symmetric equilibrium for the agents. Both agents quickly converge to an equal share of the available bandwidth. This is consistent with the unique, symmetric equilibrium predicted by our analysis in Section 4.1.

In the asymmetric case (Figure 2b), gradient descent again converges to a single equilibrium at which the more delay-sensitive agent receives a smaller share of the bandwidth.

In addition, we investigate the behavior of agents under other functional forms for g that may also be reasonable in a real setting. For example, $g(z) = z^4$ may be a better model of the sharp decrease in utility shown in Figure 1. Alternatively, recent proposals such as Copa [4] and Remy [24] have adopted utilities with logarithmic forms similar to $g(z) = \log(1+z)$. We plot the corresponding



(a) Gradient descent convergence for symmetric utilities. The agents quickly converge to equal rates, each sharing half of the link. At equilibrium, each agent sends at approximately 61 packets per time step, resulting in a total rate higher than the link capacity.



(b) Gradient descent convergence for asymmetric utilities. Agent 1, which has a higher value of α and therefore a higher penalty for delay, converges to a smaller share of the link than agent 2. At equilibrium, agent 1 sends at about 81 packets per time step and agent 2 at 29 packets per time step.

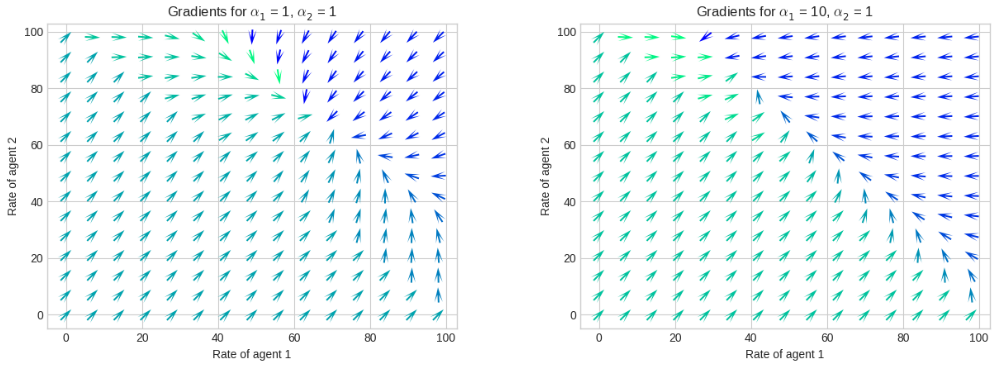
Figure 2: Convergence of gradient descent for two agents with symmetric and asymmetric utility functions, where $g(z) = z^2$. Gradient vectors only represent direction and are of equal length. The color of the gradient vector varies with the direction. When $\sum_j x_j < C$, the gradient is 1 for both agents, as there is no loss or delay.

vector fields in Figure 3. While the location of the equilibrium changes based on the functional form and the coefficients involved, all of these utilities result in a single equilibrium.

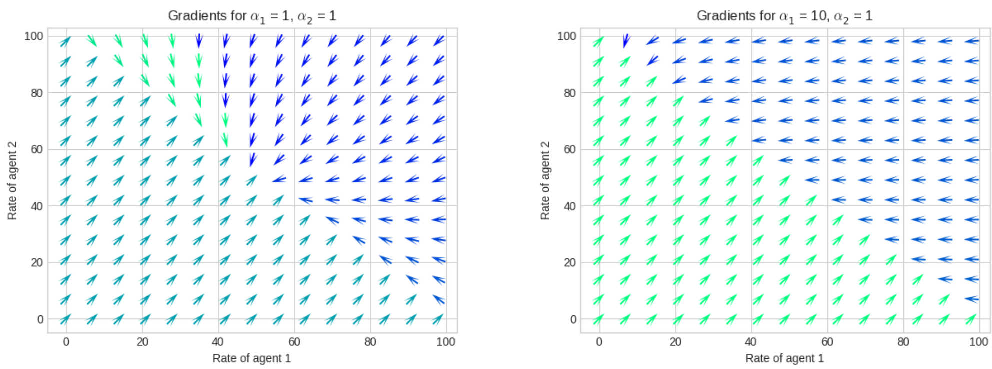
As discussed in Section 4.1, the uniqueness of equilibria helps us understand whether agents will reach a fair equilibrium independent of their initial conditions (for example, if two agents enter the network at different times). To verify this computationally, we used a grid search over the space of rates (for two agents) for $1 \leq \alpha_i \leq 10$ (setting the maximum rate at twice the network capacity) and $g(z) = z^2$ as well as $g(z) = \log(1 + z)$ to confirm that a unique pair of bids minimizes the utility gradients for the agents under these conditions. These results suggest that showing more general unique equilibrium properties may be analytically tractable for a broader class of utilities.

6 Related Work

Game-theoretic models of congestion control. PCC-Vivace [11] introduces a practical congestion-control algorithm based on prior work of Even-Dar et al. [13]. The latter paper shows that, for linear delay penalties, regret-minimizing senders converge to a Nash equilibrium. [11] adds a loss term to the utility while maintaining the social concavity condition, and provides empirical evidence that these senders perform well in practical settings. In [11], Dong et al. suggest that agents' utilities



(a) Gradient convergence in symmetric and asymmetric settings where $g(z) = z^4$.



(b) Gradient convergence in symmetric and asymmetric settings where $g(z) = \log(1 + z)$.

Figure 3: Gradient vector fields for symmetric and asymmetric utilities for two agents with varying nonlinear functional forms for $g(x)$. In each case, the agents converge to a unique equilibrium.

can be engineered so that favorable conditions, such as low latency and a fair bandwidth allocation, emerge in equilibrium. In our work, we take the perspective that agents enter the network with their own utility functions and make distributed decisions about sending, and we must understand the conditions under which agents will behave well or poorly together. This is the typical setting in wide-area networks, in which agents can change the algorithms they run locally to suit their preferences. PCC Proteus [22] expands the set of utility functions considered but is still limited to socially concave utilities.

Our work expands on these papers by analyzing a more general functional form for the utility that allows for nonlinear delay penalties that do not satisfy social concavity. We also make progress toward addressing questions about the behavior of regret minimizers in settings with existing agents that use heuristic algorithms, which [11] and [22] evaluate empirically.

Prior work [3, 9] also addresses the congestion control problem from a game-theoretic perspective. These papers assume agents running the TCP NewReno algorithm (which, roughly, linearly increases the sending rate and then divides it by a constant factor if a packet loss is observed) with variable increase and decrease coefficients, and show that for a throughput-based utility, NewReno-like agents will reach the Nash equilibrium utility values. However, this work does not answer questions regarding performance in the face of other types of agents. We argue that a model that encompasses a broad range of behavior is necessary as new algorithms are introduced into the Internet. Other classic work that models the congestion control problem as a game [19] assumes a centralized platform that can modify agent allocations in order to optimize for a global objective.

Learning and optimization in congestion control. Recent work has investigated algorithms that learn and adapt in various ways to network conditions. Remy [24] uses an offline Monte Carlo optimization procedure to find an algorithm that performs well on average in a specified set of training scenarios, given a utility function and a piecewise-linear functional form for the algorithm. PCC [10] takes an experimental approach to online optimization, adjusting sending rates by periodically testing higher and lower rates and adapting its rate if a new rate might give better utility.

A number of recent papers investigate the feasibility of using reinforcement learning in congestion control. Some of this work [5, 21, 17] operates in the single-agent video streaming setting, while other papers [26, 14, 12, 25, 20] consider the distributed congestion control setting.

7 Conclusion

Recent work [13, 11, 22] makes progress toward applying tools from online learning and regret minimization to the problem of distributed congestion control. However, these results depend on the social concavity condition on utilities, which we have demonstrated is too restrictive to describe real-world utility functions. In this work, we made progress towards understanding a more general and realistic setting by studying a broader class of utilities that is closer to the utilities of users of real applications. Although these utilities are not socially concave, we demonstrated that a unique symmetric equilibrium exists when utilities are symmetric, and empirically showed that agents running gradient descent converge to an equilibrium in both symmetric and asymmetric settings. These results suggest that agents with realistic preferences can achieve stable outcomes in the distributed network setting.

References

[1] ITU-T Recommendation G.107: The E-model: a computational model for use in transmission planning. Technical report, ITU, 2003.

[2] ITU-T Recommendation G.114: One-way transmission time. Technical report, ITU, 2003.

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

[4] Venkat Arun and Hari Balakrishnan. Copa: Practical delay-based congestion control for the internet. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. USENIX Association, 2018.

[5] Rajarshi Bhattacharyya, Archana Bura, Desik Rengarajan, Mason Rumuly, Srinivas Shakkottai, Dileep Kalathil, Ricky KP Mok, and Amogh Dhamdhere. Qflow: A reinforcement learning approach to high qoe video streaming over wireless networks. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 251–260, 2019.

[6] L. S. Brakmo and L. L. Peterson. Tcp vegas: End to end congestion avoidance on a global internet. *Selected Areas in Communications, IEEE Journal on*, 13(8):1465–1480, 1995.

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

[8] Kuan-Ta Chen, Polly Huang, and Chin-Laung Lei. How sensitive are online gamers to network quality? *Communications of the ACM*, 49(11):34–38, 2006.

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

[10] Mo Dong, Qingxi Li, Doron Zarchy, Philip Brighten Godfrey, and Michael Schapira. PCC: Re-architecting congestion control for consistent high performance. In *NSDI*, volume 1, page 2, 2015.

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

[12] Salma Emara, Baochun Li, and Yanjiao Chen. Eagle: Refining congestion control by learning from the experts. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 676–685. IEEE, 2020.

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

[14] Joyce Fang, Martin Ellis, Bin Li, Siyao Liu, Yasaman Hosseinkashi, Michael Revow, Albert Sadovnikov, Ziyuan Liu, Peng Cheng, Sachin Ashok, et al. Reinforcement learning for bandwidth estimation and congestion control in real-time communications. *arXiv preprint arXiv:1912.02222*, 2019.

[15] Andreas Hefti. Equilibria in symmetric games: Theory and applications. *Theoretical Economics*, 12(3):979–1002, 2017.

[16] J.C. Hoe. Improving the start-up behavior of a congestion control scheme for tcp. In *ACM SIGCOMM Computer Communication Review*, volume 26, pages 270–280. ACM, 1996.

[17] Tianchi Huang, Rui-Xiao Zhang, Chao Zhou, and Lifeng Sun. Qarc: Video quality aware rate control for real-time video streaming based on deep reinforcement learning. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1208–1216, 2018.

[18] Nathan Jay, Noga Rotman, Brighten Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. In *International Conference on Machine Learning*, pages 3050–3059, 2019.

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

[20] Wei Li, Fan Zhou, Kaushik Roy Chowdhury, and Waleed Meleis. Qtcp: Adaptive congestion control with reinforcement learning. *IEEE Transactions on Network Science and Engineering*, 6(3):445–458, 2018.

[21] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 197–210, 2017.

[22] Tong Meng, Neta Rozen Schiff, P Brighten Godfrey, and Michael Schapira. 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*, pages 615–631, 2020.

[23] Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

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

[25] Zhiyuan Xu, Jian Tang, Chengxiang Yin, Yanzhi Wang, and Guoliang Xue. Experience-driven congestion control: When multi-path tcp meets deep reinforcement learning. *IEEE Journal on Selected Areas in Communications*, 37(6):1325–1336, 2019.

[26] Francis Y Yan, Jestin Ma, Greg D Hill, Deepti Raghavan, Riad S Wahby, Philip Levis, and Keith Winstein. Pantheon: the training ground for internet congestion-control research. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 731–743, 2018.

[27] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

A Social concavity requires near-linear delay

Proof of Proposition 1. We prove the lemma by evaluating the social concavity conditions: first, that the utility should be concave in x_i, and second, that it should be convex in x_{-i}.

Recall that the utility can be written

$$u_i(x) = \min(C, \sum_i x_i)\frac{x_i}{\sum_i x_i} - \alpha_i x_i g\left(\left[\frac{\sum_i x_i - C}{C}\right]_+\right) - \beta_i x_i\left(\max\left(0, 1 - \frac{C}{\sum_i x_i}\right)\right)$$

First note that if $\sum_j x_j \le C$ then $\min(C, \sum_j x_j)\frac{x_i}{\sum_j x_j} = x_i^t$, which is clearly linear in x_i^t, and the delay and loss terms are 0. Hence, for the remainder we assume $\sum_j x_j > C$.

For the first condition, we take derivatives with respect to x_i:

$$\begin{aligned}
\frac{\partial u_i}{\partial x_i} &= C\frac{\sum_j x_j - x_i}{(\sum_j x_j)^2} - \frac{\alpha_i x_i}{C} g'\left(\frac{\sum_j x_j - C}{C}\right) - \alpha_i g\left(\frac{\sum_j x_j - C}{C}\right) - \beta_i + \beta_i C\frac{\sum_j x_j - x_i}{(\sum_j x_j)^2} \\
&= C(1+\beta_i)\frac{\sum_j x_j - x_i}{(\sum_j x_j)^2} - \frac{\alpha_i x_i}{C} g'\left(\frac{\sum_j x_j - C}{C}\right) - \alpha_i g\left(\frac{\sum_j x_j - C}{C}\right) - \beta_i \\
\frac{\partial^2 u_i}{\partial {x_i}^2} &= 2C(1+\beta_i)\frac{x_i - \sum_j x_j}{(\sum_j x_j)^3} - \frac{\alpha_i x_i}{C^2} g''\left(\frac{\sum_j x_j - C}{C}\right) - \frac{2\alpha_i}{C} g'\left(\frac{\sum_j x_j - C}{C}\right).
\end{aligned}$$

Then we can find the conditions for g'' in order for the utility to be concave in x_i:

$$\begin{aligned}
2C(1+\beta_i)\frac{x_i - \sum_j x_j}{(\sum_j x_j)^3} - \frac{\alpha_i x_i}{C^2} g''\left(\frac{\sum_j x_j - C}{C}\right) - \frac{2\alpha_i}{C} g'\left(\frac{\sum_j x_j - C}{C}\right) &\le 0 \\
-\frac{\alpha_i x_i}{C^2} g''\left(\frac{\sum_j x_j - C}{C}\right) &\le \frac{2\alpha_i}{C} g'\left(\frac{\sum_j x_j - C}{C}\right) - 2C(1+\beta_i)\frac{x_i - \sum_j x_j}{(\sum_j x_j)^3} \\
g''\left(\frac{\sum_j x_j - C}{C}\right) &\ge -\frac{2C}{x_i} g'\left(\frac{\sum_j x_j - C}{C}\right) + \frac{2C^3(1+\beta_i)}{\alpha_i x_i}\frac{x_i - \sum_j x_j}{(\sum_j x_j)^3}
\end{aligned}$$

Note that as $x_i \to \infty$, this lower bound goes to 0.

For the second condition, we take derivatives with respect to x_{-i}:

$$\begin{aligned}
\frac{\partial u_i}{\partial x_{-i}} &= -C\frac{x_i}{(\sum_j x_j)^2} - \frac{\alpha_i x_i}{C} g'\left(\frac{\sum_j x_j - C}{C}\right) + \beta_i C\frac{x_i}{(\sum_j x_j)^2} \\
\frac{\partial^2 u_i}{\partial {x_{-i}}^2} &= 2C\frac{x_i}{(\sum_j x_j)^3} - \frac{\alpha_i x_i}{C^2} g''\left(\frac{\sum_j x_j - C}{C}\right) + 2\beta_i C\frac{x_i}{(\sum_j x_j)^3} \\
&= (2C)(1+\beta_i)\frac{x_i}{(\sum_j x_j)^3} - \frac{\alpha_i x_i}{C^2} g''\left(\frac{\sum_j x_j - C}{C}\right).
\end{aligned}$$

For convexity, we need

$$\begin{aligned}
(2C)(1+\beta_i)\frac{x_i}{(\sum_j x_j)^3} - \frac{\alpha_i x_i}{C^2} g''\left(\frac{\sum_j x_j - C}{C}\right) &\ge 0 \\
-\frac{\alpha_i x_i}{C^2} g''\left(\frac{\sum_j x_j - C}{C}\right) &\ge (2C)(1+\beta_i)\frac{x_i}{(\sum_j x_j)^3} \\
g''\left(\frac{\sum_j x_j - C}{C}\right) &\le \frac{(2C^3)(1+\beta_i)}{\alpha_i}\frac{1}{(\sum_j x_j)^3}
\end{aligned}$$

□

B Equilibria beyond social concavity

Proof of Proposition 2. Our statement follows from verifying the inequality in Proposition 2, condition 1 of [15]. To do so, we first write our individual utilities in terms of the action x_i and the aggregate bandwidth $S = \sum_j x_j$.

$$u_i(x, S) := x_i\left(\frac{\min(C,S)}{S} - \alpha g\left(\frac{S-C}{C}\right) - \beta\left(1 - \frac{\min(C,S)}{S}\right)\right).$$

The condition is satisfied if the following partial derivative inequality holds uniformly in x and S. While proposition 2 provides more general conditions that restrict the inequality to a smaller subset of (x, S). We ignore this for simplicity, as this uniform bound approach provides tight characterizations in the closely related Cournot game.

$$\frac{\partial^2 u_i(x,S)}{\partial x^2} + \frac{\partial^2 u_i(x,S)}{\partial x \partial S} < 0.$$

Note that the second partial derivative with respect to x is zero, and thus we have the stated result,

$$\frac{\partial^2 u_i(x,S)}{\partial x \partial S} = \frac{-1-\beta C}{S^2}\mathbf{1}_{S>C} - \alpha g'\left(\frac{S-C}{C}\right)\frac{1}{C} < 0$$

where $\mathbf{1}_{S>C}$ is an indicator that is one whenever $S > C$ and zero otherwise. □

Proof of Proposition 3. We verify the inequality in proposition 2, condition 2 of [15] which is fulfiled for a n-player game if

$$\left.\frac{\partial^2 u_i(x,S)}{\partial x^2}\right|_{S=nx} + (n+1)\left.\frac{\partial^2 u_i(x,S)}{\partial x \partial S}\right|_{S=nx} + n\left.\frac{\partial^2 u_i(x,S)}{\partial S^2}\right|_{S=nx} < 0.$$

From Proposition 2 we know $\frac{\partial^2 u_i(x,S)}{\partial x^2}\big|_{S=nx} = 0$ and $\frac{\partial^2 u_i(x,S)}{\partial x \partial S}\big|_{S=nx} < 0$ implying that it suffices to show $\frac{\partial^2 u_i(x,S)}{\partial S^2}\big|_{S=nx} < 0$. Now note that

$$\frac{\partial^2 u_i(x,S)}{\partial S^2} = x_i\left(\frac{2+2BC}{S^3}\mathbf{1}_{S>C} - \frac{\alpha}{C^2}g''\left(\frac{S-C}{C}\right)\right).$$

Since $2 + 2BC \geq 0$ and $S > 0$ we can upper bound this partial derivative using $\frac{2+2BC}{S^3}\mathbf{1}_{S>C} \leq \frac{2+2BC}{C^3}$. Using the fact that $x_i > 0$, it suffices to have

$$\frac{2+2BC}{C^3} - \frac{\alpha}{C^2}g''\left(\frac{S-C}{C}\right) < 0.$$

to show $\frac{\partial^2 u_i(x,S)}{\partial S^2}\big|_{S=nx} < 0$. Evaluating at $S = nx$ and rearranging terms gives the stated result. □