Quantifying the Transient Performance of Congestion Control Algorithms

Yixin Shen1,3, Zili Meng1,3, Jing Chen1,3, Mingwei Xu1,2,3
1Institute for Network Sciences and Cyberspace, 2Department of Computer Science and Technology, Tsinghua University
3Beijing National Research Center for Information Science and Technology (BNRist)

CCS CONCEPTS
• Networks → Network performance analysis.

ACM Reference Format:

1 INTRODUCTION

A plethora of Congestion Control Algorithms (CCA) [1–4] have emerged for decades along with ever-evolving network technologies and rapidly changing application requirements. To quantitatively compare different CCAs, researchers have proposed different metrics from various aspects, e.g., throughput and delay for performance, Jain’s index [6] for fairness. In practice, these metrics are mostly measured in a long-term manner (e.g., everyday [13]).

However, with the rising demand of the performance from applications, the transient performance of CCAs during network fluctuations is also critical [9]. For example, for low-latency applications such as videoconferencing, several 100ms stalls due to transient network fluctuations could severely degrade the users’ experience. Therefore, whether CCAs could responsively converge to the network condition itself is complex and changeable, making the convergence ability of CCAs to provide better understandings for operators.

Yet, it is challenging to quantify the transient reaction, i.e. the convergence ability, of CCAs, which can be defined as the pattern between network status change and the steady state. On one hand, the network condition itself is complex and changeable, making the input of CCA complicated. Inspired by Signal Processing, as the CCA’s output will not change without the change of input, we can treat the input as a combination of steps at different moments. On the other hand, due to the complex nature of CCAs (usually piecewise and nonlinear), it is hard to mathematically formulate

1There are recent efforts trying to formulate CCAs, which remains preliminary and might have performance gaps for quantitative comparisons [8].

The research is supported by the National Key R&D Program of China under Grant 2018YFB1800302, the NSFC under Grant 92038302 and 61625203.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8629-6/21/08...$15.00
https://doi.org/10.1145/3472716.3472861

Figure 1: Transient reaction subjects to damped oscillation.
next network change. When a CCA’s transient reaction is characterized as critically-damped (the red curve), it means the CCA can react to the network condition changes fast with minimal cost and enter the steady state most quickly, which is apparently the optimal case. Hence, we can evaluate a CCA’s transient performance by capturing the feature of damped oscillation of the system.

Based on this observation, we propose a framework to give an effective analysis on transient performance based on the damped oscillation hypothesis. To quantify the transient reaction, we measure the gap between the actual oscillation and the critically damped oscillation. However, we are confronted with 2 challenges: (1) How can we detect when the CCA enters the steady state and demarcate the short-term reaction period? (2) How can we quantify the behavior gap? In response, we use a steady-detect window to find when a CCA enters the steady state based on the assumption that a practical CCA should have limited fluctuations in the steady state. To quantify the reaction performance, we use Normalized Area (i.e., the accumulated gap between the theoretical throughput and the experimental throughput before CCA’s convergence) as the metric and a smaller area means better performance. Our framework could evaluate how close a CCA is to the optimal (critically damped) transient performance given a network condition. Therefore, with a bunch of CCAs at hand, network operators can adopt our framework to select the most well-behaved CCA (in the sense of transient performance) under different network scenarios.

We use the framework to quantify and compare the transient performance of three CCAs: BBR[3], Cubic [4] and Reno [5], giving a preliminary and reasonable evaluation on transient performance.

2 FRAMEWORK

Our framework consists of 2 parts: (1) Steady State Detection to confirm when CCA enters the steady state, and (2) Reaction Quantification to quantify the transient performance.

**Steady State Detection.** As mentioned in §1, transient reaction is the pattern between network status change and the steady state. Thus, before measuring the reactive tendency, we must confirm when the CCA enters the steady state (the convergence point). The convergence points may vary greatly among different CCAs, so we need to find out the specific convergence point case-by-case. If we determine the convergence point earlier than its actual time, the reaction performance measured will be worse than practice. If we determine the convergence point later than the actual time, the reaction performance measured will be worse than practice. Hence, we use a set of steady-detect windows (consisting of several consecutive windows of the same size and uniform spacing) to accurately position the convergence point. When the mean and variance are almost the same among these windows, we consider the CCA has already entered the steady state. We choose three 10-RTT-sized windows, and the beginning of each window is separated by 4-RTT. This does not bring much computation overhead and is also enough to precisely detect the steady state in our experiments.

**Reaction Quantification.** Transient reaction is captured with random noise, disabling us to fit the curve and derive an accurate mathematical expression of the CCA’s throughput for the classification of oscillations. In response, we calculate Normalized Area to quantify the transient performance. Area represents the accumulated gap between the throughput of CCAs and the ground-truth available bandwidth from the network change to the convergence point (e.g., for critically damped oscillation, it is the area between the red and pink dashed curve in Figure 1(a)), which is indicative of the oscillation tendency. Obviously, critically damped oscillation has the smallest area and the areas of both under-damped and over-damped largen with deviation of the damping. Additionally, the impact of converging to a non-optimal value is covered by this method. We normalize the area by reaction time and bandwidth change, so the relative area value among different scenarios (e.g., with different bandwidth) is referable. Note that, by taking the Normalized Area as our metric, we make no distinction between under-damped oscillation and over-damped oscillation, simply examining the gap between them and the optimal situation.

3 EXPERIMENTS AND FUTURE WORK

We implement our framework on the Congestion Control Plane [10], with some of its already deployed CCAs2: Reno [5], Cubic [4] and BBR [3]. We use Mahimahi [11] to emulate the delay and the bandwidth change, whose noise subjects to 3σ-truncated normal distribution \( N(0, \sigma) \). We measure the throughput over the course of 100 60-second experiments under each set of network conditions using Iperf [12]. BDP-related-sized window is used to smooth the raw throughput data recorded at packet-level on milliseconds-scale by Mahimahi. The result in Figure 1 shows that the throughput of different CCAs subjects to different damped oscillations.

To illustrate the effectiveness of our framework, we set a step change of link capacity from 0 to 12Mbps and RTT as 50ms with different buffer size as an example. Figure 2(a) shows a small buffer case while Figure 2(b) shows a larger buffer case. From the results, we can compare the transient performance of different CCAs in different cases, which is reasonable and consistent with theory. When the buffer is smaller, loss-based CCAs have better reaction performance and BBR has a long tail. This is because loss-based CCAs increase their congestion window more quickly than BBR. When the buffer is larger, loss-based CCAs degrade their performance.

Additionally, the results give us the suggestion when considering the transient performance: we can choose loss-based CCAs with small buffer while choose BBR with large buffer.

In the future, we plan to adopt our framework to learning-based CCAs in the hope of providing more insight on them.

---

2https://github.com/BobAnkh/TFCCA
3We use the CCA’s official implementations in https://tcp-project.github.io
REFERENCES


