How I Learned to Stop Worrying About CCA Contention

Lloyd Brown[♡], Yash Kothari[♠], Akshay Narayan[♡], Arvind Krishnamurthy[◇], Aurojit Panda[♣], Justine Sherry[♠], Scott Shenker[♡†]

♡ UC Berkeley ♠ CMU ◇ University of Washington ♣ NYU † ICSI

Abstract

This paper asks whether inter-flow contention between congestion control algorithms (CCAs) is a dominant factor in determining a flow's bandwidth allocation in today's Internet. We hypothesize that CCA contention typically does not determine a flow's bandwidth allocation, present an initial analysis in support of this hypothesis, propose a measurement technique and study to settle this question, and discuss the implications should the hypothesis prove true.

CCS Concepts

• **Networks** → **Public Internet**; **Transport protocols**.

Keywords

Congestion Control

1 Introduction

Contention between Internet flows has long been considered a core factor in determining bandwidth allocations. As a result, the community has invested substantial effort into analyzing the long-term bandwidth allocation between competing flows. For example, TFRC [1] was designed to guarantee that UDP flows would share bandwidth evenly with TCP NewReno during congestion avoidance, and BBR has been shown [2] to take more than its long-term fair share of bandwidth when competing against NewReno and Cubic. Further, the research community has proposed prescribing how CCAs should interact with each other, from Floyd and Fall's TCP Friendliness [3], to Jain's fairness index for throughput [4], to Ware et al.'s proposal to center "harm" over a set of metrics.

In this position paper, we hypothesize that **there do not remain common scenarios in the modern Internet in which CCA contention is the dominant factor in determining flows' bandwidth allocations.** While additional measurement data is required to conclusively resolve this hypothesis, we argue that it is plausible. In other words, if our hypothesis holds, then for most flows on the Internet, contention between CCAs doesn't matter. Instead, we argue that a second set of factors, including in-network bandwidth management techniques (e.g., fair queueing [5], traffic engineering [6–8], and capacity planning [9]), along with application traffic limitations, primarily determine bandwidth allocations on the Internet today.

Why is resolving this hypothesis important? First, if our hypothesis is true, this has implications for CCA design: designers of new CCAs should no longer concern themselves with contention properties, and analysis of new and old CCAs should de-emphasize this property. Second, if CCA dynamics do not determine bandwidth allocations on the Internet and outcomes are instead a function of in-network and host rate shaping, how do these mechanisms work in the wild? Are they principled, equitable, and efficient?

In our analysis, it is important to clarify the difference between the terms "contention" and "congestion." Congestion, and even congestion collapse, can occur without CCAs interacting with each other. For example, consider a peering link overloaded with web traffic (consisting of short flows). This link will experience queueing, and the traffic will experience signs of congestion (packet loss and high delays), but the individual flows' CCAs will not interact. Instead, situations such as this one are usually managed by traffic engineering (TE) systems [6–8]. Contention, on the other hand, occurs when two long-running flows share the same bottleneck link, and their allocated bandwidth is some function of their CCA dynamics. We discuss the precise prerequisites for CCA contention in §2.

In the rest of this paper, we present:

(1) A discussion of modern Internet traffic dynamics in which we argue that our hypothesis is plausible (§2).
(2) Initial measurement results using data from a traditional measurement source (M-Lab [10]) and a proposal for a new active measurement technique that seeks to directly measure CCA contention (§3).
(3) A discussion of the implications of our hypothesis should it prove true, both for CCA design and analysis as well as the Internet architecture (§5).

2 Where is the Contention?

Why do we suspect that flows no longer contend for bandwidth on the Internet? Recall that for contention to occur, at least two flows must: (i) share a path segment, (ii) experience a bottleneck in that path segment, and (iii) use the same queue at the bottleneck link. In this section, we argue that the confluence of these three conditions is rare in the modern Internet for two reasons, summarized in Figure 1:

HotNets '23, November 28–29, 2023, Cambridge, MA, USA

ACM ISBN 979-8-4007-0415-4/23/11.
https://doi.org/10.1145/3626111.3628204

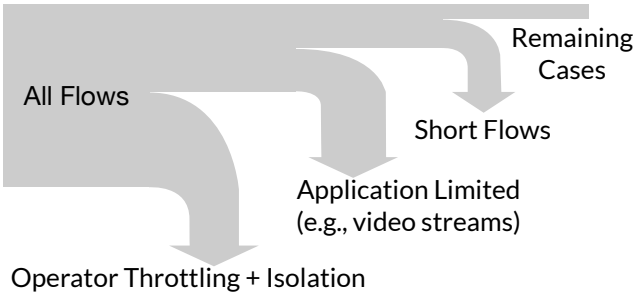


Figure 1: Operator policies driven by the economics of the commercial Internet combined with traffic dynamics combine to reduce or eliminate inter-flow contention.

Economics (§2.1) Users pay for access bandwidth, and providers provision and configure their networks to provide users with acceptably good network capacity. Furthermore, providers limit how much data a user can send, ensuring that a user's sending rate does not exceed the bandwidth for which they have paid. Note that while operators' scheduling mechanisms, such as fair queueing, cannot change the total amount of traffic in a queue, they can enforce a bandwidth allocation (i.e., max-min fairness) as well as isolation between flows. Thus, a universal deployment of fair queueing (for example) would entirely eliminate the role of CCA dynamics in determining bandwidth allocations.

Access Bottlenecks and App Limitations (§2.2) While the economic factors described above limit contention between flows from different users, they do not eliminate contention between flows from a single user (since most isolation mechanisms operate on a per-user, not per-flow, basis). However, increases in the amount of available bandwidth and application limitations on offered load limit even contention between a single user's flows. This trend is likely to continue with Moore's law of compute scaling having ended, but Edholm's law of bandwidth scaling continuing to the present day. Further, application bandwidth demands have not grown as much as available bandwidth, so increasingly, an application's bottleneck is not in the network but rather its own offered load.

These factors do not eliminate contention in all cases, and inter-flow CCA contention can still, in some cases, determine bandwidth allocations. We discuss these cases and how they might easily evolve to eliminate CCA contention in §2.3.

2.1 Economics

The Internet began as a shared communication medium between academic institutions (ARPANet, NSFNet, etc). As a result, Internet users developed techniques for dealing with traffic contention as a set of shared protocols that governed how good Internet citizens would behave when accessing the common infrastructure. After a series of "congestion collapses" in 1986 [11], Jacobson and Karels developed the first congestion control algorithm [12] based on Chiu and Jain's AIMD technique [13]. As the Internet grew to encompass commercial traffic, over time, it evolved from a government-funded research network to today's commercial Internet.

This transition had a number of effects, including exponential growth in usage, but also operator measures to isolate their customers' traffic. ISPs today have a financial incentive to provision sufficient bandwidth for their customers' usage lest those customers switch to competing ISPs[1]. Further, ISPs advertise services based on a promised service quality (e.g., 4K video streaming), and customers are unlikely to be impressed with protestations that this service quality was not met not due to access link provisioning but rather contention with other customers. As a result, many ISPs aggressively upgrade their inter-domain link capacities to maintain link utilization below 60-70% [9]. As a recent example, in response to increased traffic volume during the first year of the COVID-19 pandemic, IXP members in Europe and the US increased their peering link capacities by up to 20% [14].

While capacity planning limits the amount of congestion an operator's traffic might encounter in aggregate, operators also use a bevy of traffic management techniques to ensure that they limit users to the level of bandwidth the users paid for, isolate users' traffic, and implement other policies. ISPs commonly offer multiple service levels—Paul et al. [15] report that US ISPs provide between 3 and 11 unique plans with monthly prices ranging from $20 to $120—so throttling user traffic incentivizes users to upgrade. For example, Flach et al. found [16] that traffic policing (in which the ISP's router drops a user's traffic above a configured rate) occurs on 7% of measured paths. They note that directly detecting the presence of shaping (in which the router queues the user's excess traffic rather than dropping it) is challenging. Similarly, Li et al. [17] found that traffic differentiation (i.e., throttling) is also common and identified 77 ISP-App throttling pairs. The prevalence of differentiation implies that bandwidth shaping (a super-set of differentiation) must be at least as prevalent.

At the largest scale, hyperscalers deploy private WANs over leased-lines [18, 19] to carry their traffic. Since only one organization controls such a WAN, it can deploy host-based bandwidth allocation [20] or priority queueing [21] to eliminate inter-flow contention. For example, Google uses BwE [22] to allocate bandwidth in its private WAN. BwE integrates with applications that report their bandwidth demand to centrally determine bandwidth allocations across the entire network. This isolates applications from each other and eliminates inter-flow contention across applications.

Commercial entities appear to have responded to this increasing level of isolation by developing proprietary, more

[1]Even in areas lacking ISP competition, ISPs are incentivized to provide advertised service levels to avoid customer complaints and encourage customers to upgrade to pricier plans.

aggressive, and application-specific CCAs. This is a significant departure from the initial introduction of congestion control mechanisms as *protocols* implemented and deployed with a near-universal agreement to today's custom *algorithms*. For example, Akamai's FastTCP remains proprietary, and Google developed its BBR and BBRv2 without disclosing details about those CCAs. With user-space CCA implementations gaining popularity [23, 24], we can expect this trend to continue.

2.2 Access Bottlenecks and App Limitations

Internet measurement studies have observed three properties from which we infer a minimal role for CCA dynamics in determining bandwidth allocations: (a) most paths are short [25], (b) most flows are short [26], and (c) most of the bytes are from video streaming traffic which has bounded throughput demand [14].

First, most paths on the Internet are short, and thus, in most cases, access links—i.e., links within a home network or a device's cellular link—are the only candidates for inter-flow contention. Recent surveys [25, 27–29] have found that most ISPs host CDN nodes (from which they can respond to requests) and peer with cloud providers (e.g., Google directly connects to networks representing more than 60% of end-user prefixes) through carefully provisioned and managed links [6–8]. Furthermore, cloud providers host a significant fraction of the services accessed by most users. Consequently, access links are the only potential location for inter-flow contention for a significant portion of the traffic. On access links, inter-flow contention can affect bandwidth allocation only if a user's applications simultaneously offer enough load to exceed the access link's capacity. Otherwise, each application would simply receive a bandwidth allocation corresponding to its offered load.

Second, most application flows are short [14, 26, 30]. These flows fit within the initial congestion window or, in the case of a persistent connection, within the available congestion window. Since these flows are not limited by any flow-level transmission mechanism (except, perhaps, a rate limit), per-flow CCA dynamics cannot affect their bandwidth allocations. Further, only in-network mechanisms can affect these flows; intermediate routers might drop their component packets, or a cellular base station might delay their transmission.

Finally, for inter-flow contention to occur on access links, there must be enough application traffic to exceed the link's provisioned capacity. However, most flows are either app-limited or (as discussed above) too short to contend for bandwidth. For example, most transmitted bytes are from video streaming traffic, and even the most aggressive form of video streaming—real-time video game streaming—consumes only 20–30 Mbit/s at the highest bitrates [31]. This remains less than the available user bandwidth at broadband speeds in many regions of the world [32]. Even if the video stream saturated its access link, rather than inter-flow dynamics determining bandwidth allocations, adaptive bitrate (ABR) algorithms would reduce video streams' throughput demand.

Past measurement studies corroborate these observations. For example, Araújo et al. find that "flow rates are not typically dictated by TCP congestion control alone;" in their measurement dataset, less than 40% of traffic was neither application-, host-, nor receiver-limited [33]. Further, Yang et al. found [32] that deployed home Wi-Fi routers commonly reflect the bandwidth local ISPs offer; in other words, the Wi-Fi link itself is as much of a bottleneck as the home access link. This further reduces the likelihood that two flows will contend for bandwidth at the access link.

Of course, there remain plausible scenarios where flows are not application- or receiver-limited and are long enough to contend for bandwidth with other flows. We discuss these below in §2.3.

2.3 What about…?

We do not claim that inter-flow bandwidth contention does not occur *at all*; it can occur in some cases, e.g., persistently backlogged flows (software updates, etc) on access links. Rather, we merely claim that it is not the primary determinant for flows' bandwidth allocations. For the rest of this section, we discuss scenarios in which contention might still occur.

Couldn't contention still happen at access links? Contention can still happen at access links in some cases; we can create these conditions by starting two persistently backlogged connections from behind an access link. However, we note that the user (or their operating system) could use an end-host traffic shaper (e.g., Linux qdisc) to implement isolation. Software implementations are sufficient for most access links, and faster access links are unlikely to be congested in the first place. At the access link itself, recent work has proposed to bring bandwidth shaping to the edge even in cases where the true bottleneck is elsewhere [34, 35]. Overall, fair queueing and isolation is cheap and easy to implement, and these solve the per-flow contention problem more effectively than any flow-level CCA mechanism can.

What about the developing world? Chen et al. showed that on certain links where the bandwidth-delay product (BDP) is less than one packet, congestion control mechanisms can unfairly allocate bandwidth over short (~20 seconds) timescales [36]. This is primarily due to timeout mechanisms that starve an arbitrary set of flows. Users in rural areas and the developing world are the most likely to experience such conditions [37].

We note two factors that minimize the role of flow contention for bandwidth in these scenarios. First, ISPs in both rural regions and in the developing world follow the same economics as those in other parts of the Internet and use similar traffic management techniques to isolate and throttle user traffic. Further, Internet growth in the developing

world is dominated by wireless links, whether cellular (most commonly) [37, 38], or using commercial aircraft [39] or satellites [40, 41]. Of course, cellular links already provide flow isolation. Given satellite networks also experience bandwidth and RTT fluctuations [42], they similarly adopt flow isolation and bandwidth management techniques [43], as is already common in other wireless ISP deployments [44, 45].

While these factors provide an initial indication that flow contention may not happen in low-bandwidth or sub-packet scenarios, they do not provide a conclusive answer, and studying such scenarios remains an open question.

What about hypergiants? One ostensible source of contention is large content providers overwhelming peering links with large traffic aggregates. One popular example is the 2014 peering dispute between Netflix and Comcast [46]. In this case, the dispute's source was not flow-level contention but rather the need for extra link capacity. Since Netflix is a video streaming service and video traffic is generally application-limited, individual flows' CCA dynamics could not have played a significant part in determining bandwidth allocations; those CCAs would control only the allocations at the unloaded edge bottleneck links. Rather, the issue was the aggregate demand over the peering link, which no individual flow's CCA controls. In this example, Netflix and Comcast resolved the peering dispute with a capacity upgrade corresponding to a commercial agreement.

What about datacenters? Some proposals for datacenter-specific flow control mechanisms use CCA mechanisms to allocate bandwidth [47–51], but other more recent works propose active in-network mechanisms to enforce isolation [52, 53]. Overall, since a single entity–a cloud provider– manages a datacenter, it can choose the bandwidth allocation mechanism that works best for its needs.

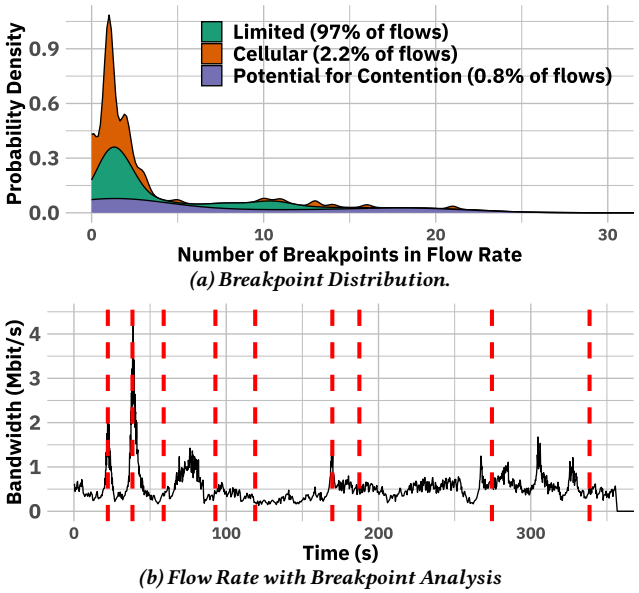
3 Measuring CCA Contention

How can we determine whether this paper's hypothesis is true? We discuss past approaches to measuring CCA dynamics and present an initial analysis using M-Lab [10] data, but we observe that passive measurement approaches cannot conclusively determine the presence (or absence) of CCA contention. Instead, in §3.2, we propose an *active* measurement approach based on a recently proposed CCA, Nimbus [54].

3.1 Passive Measurement

Existing work in measuring CCA behavior takes one of two common approaches: controlled testbed experiments or speedtest-style user-driven bandwidth testing (e.g., from a CDN or dedicated speedtest service). Unfortunately, neither approach is sufficient to conclusively determine CCA contention's impact on flows' bandwidth allocations.

Testbed Experiments The goal of testbed congestion control analysis work is to gain an understanding of the behavior



(a) Breakpoint Distribution.

(b) Flow Rate with Breakpoint Analysis

Figure 2: Breakpoint Analysis

of the algorithm by observing its reactions to different inputs. These works [55, 56] use controlled environments where researchers can carefully manipulate network conditions to get a comprehensive view of the CCA's actions. This approach is useful for understanding *how* CCAs interact but cannot determine *whether* they interact.

Crowdsourced Bandwidth Testing Users who wish to test their connection quality often make use of speedtests. Clients initiate speedtest requests (or requests for content) to specific servers or CDN nodes, respectively. Then, the speedtest operator can measure connection telemetry and store it for future analysis [10, 32, 57]. How much can this data tell us? To find out, we analyzed data from the M-Lab project, which publishes data from user-initiated network data test (NDT) [58, 59] measurements. These measurements contain TCPInfo statistics from each data transfer, including the source and destination IP addresses, achieved throughput and RTT, the amount of time the connection was application-limited, etc. Since the M-Lab data contains snapshots of this data over each flow's lifetime, we can attempt to identify cases of CCA contention by searching for cases where a flow's achieved throughput level changed during its lifetime. This might indicate that it was contending for bandwidth with another flow, and the flow's CCA adjusted its sending rate in response.

To perform this analysis, we attempt to remove flows from the dataset that we know were unlikely to have experienced contention: application- or receiver-limited flows and flows we infer to use cellular links. We queried the M-Lab NDT dataset for one month (June 2023) of flow data (9,984 flows). We categorized flows as application-limited if the `AppLimited` field was greater than zero, and similarly we categorized a flow as receiver-limited if the `RWndLimited` field was greater

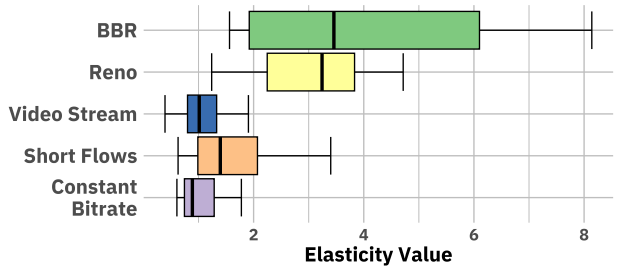


Figure 3: Active measurement allows directly measuring elasticity, a measure of the level of contention a flow experienced.

than zero. Since the data does not identify whether a flow traversed a cellular link, we use a heuristic to identify these cases: if the maximum RTT and minimum RTT the flow experienced differ by at least 100ms, we classify it as cellular. After categorizing flows, we used the `ruptures` [60] library's binary segmentation breakpoint detection algorithm to identify periods of time when each flow experienced a change in its achieved throughput. Since the breakpoint algorithm requires a prior on the ideal number of breakpoints (and we have no such prior), we employed it in rounds. In each round, we use binary segmentation to find the next breakpoint, and we stop once the average standard deviation of each segment is less than one-third of the standard deviation across all segments. We show an example in Figure 2b, where we mark the breakpoints with a vertical dotted red line. The breakpoints track large changes in the flow's sending rate. Of the 9,894 flows, we remove 5,761 for having too few data points ($\leq$ 12) for breakpoint analysis. We show results for the remaining flows in Figure 2a. First, corroborating our analysis in §2.2, most flows are application- or receiver-limited (4,008) or traverse cellular links (90); only 35 flows (fewer than 1%) of flows remain afterwards. Of these, there is a clear skew to a small number of breakpoints; i.e., flows generally experience a static bandwidth allocation throughout their lifetime. This provides some initial evidence that most flows do not experience CCA contention. However, this data cannot conclusively determine the absence of CCA contention (indeed, many flows experienced at least one breakpoint) even if it covered all Internet paths. This is because speedtest data uses passive telemetry using fixed CCAs (Cubic and BBR), and thus cannot show what *would have* happened had the CCA competed for bandwidth more aggressively.

3.2 Actively Measuring Elasticity

Instead, we propose an *active* measurement technique. If we run a speedtest-style experiment but use a CCA which actively determines whether cross traffic on the links it traverses is actively competing for bandwidth, we could use the CCA's report of the cross traffic's aggressiveness as an indicator of bandwidth contention. If the CCA reported no contention on most paths, we could conclude that CCA contention is rare. To this end, we observe that recent CCAs propose *mode-switching* techniques to determine the nature of cross traffic on their path and enable delay-based congestion control techniques in the absence of such cross traffic. While one such work, Copa [61], simply checks whether a path's cross-traffic adheres to Copa's delay oscillation dynamics, another, Nimbus [54] actively probes for cross traffic that responds to short-term changes in available bandwidth (the authors refer to such traffic as *elastic* cross traffic). Nimbus estimates a path's *elasticity*, i.e., the degree to which cross traffic on a path responds to fluctuations in available bandwidth, and uses a delay-based CCA mode when elasticity is low and a loss-based CCA otherwise.

We propose to use this CCA as a measurement tool to determine whether CCA contention occurs. If, for a probe flow, we use Nimbus but disable mode-switching, maintain the bandwidth oscillations, and measure the reported elasticity of Internet paths, we can determine whether cross traffic on those paths contended for bandwidth with that probe flow. We demonstrate a proof-of-concept of this technique in Figure 3. We run five types of traffic for 45 seconds each on a 48Mbps, 100ms emulated Mahimahi [62] link: along with a probe flow using Nimbus, we run (in sequence) two persistently backlogged flows that contend for bandwidth using CCAs Reno and BBR, as well as a video stream, a set of short flows with poisson arrivals, and constant bitrate (CBR) UDP traffic. We can clearly observe higher values for the elasticity metric for the flows that contend for bandwidth. We refer readers to Nimbus [54] for a full description and evaluation of the elasticity measurement algorithm.

4 Related Work

In addition to measurement and analysis work discussed previously, we call attention to a few additional related works.

Measuring Congestion Dhamdhere et al. [63] and related work [64] use a measurement technique called "time-series latency probes (TSLP)." With TSLP, the observation point sends TTL-limited latency probes to routers just before and after a given link and measures that link's latency differential. By measuring this value longitudinally, researchers using TSLP can identify time periods where individual links experienced inflated queueing delays; the technique infers congestion from the presence of such inflated queueing delays. While this approach is effective at identifying underprovisioned links in a lightweight manner, it cannot discriminate between cases where individual flows contend for bandwidth and cases where aggregates consisting of shorter and application-limited flows overwhelm a given link. Even so, we note Dhamdhere et al.'s finding that "…we did not find evidence of widespread endemic congestion on interdomain links between access ISPs and directly connected transit and content providers."

Evolving Congestion Control Recent work has proposed extending congestion control mechanisms to implement sharing information or state amongst the flows of a host [65], rack [66], site [34, 35], or organization [67]. These proposals argue that sharing information is helpful for using the network efficiently as well as prioritizing certain traffic. Effectively, they propose supplanting CCA contention as a mechanism for bandwidth allocation and replacing it with a more centralized mechanism, whether fair queueing or prioritization.

Critiquing Throughput-Centricity In a recent HotNets paper, Ware et al. argued against "throughput centricity" as a means of analyzing CCA contention, since this ignores other critical application metrics such as latency [68]. We discuss how CCAs might evolve to fairly contend on metrics other than throughput in §5.2, but we note that our goal in this paper is to analyze whether CCA contention occurs in the first place.

5 Implications

While further study is needed to test this paper's core hypothesis that contention between CCAs is no longer the dominant factor in determining flows' bandwidth allocations in today's Internet, we have presented initial evidence in its support. Therefore, we conclude by discussing how this hypothesis being true should affect future networking research and protocol design.

5.1 Future of Congestion Control

In recent years, development in congestion control has shifted towards providing three goals simultaneously: high throughput, low delay, and fairness [54, 61, 69]. In fact, recent analysis has shown that delay-minimizing methods cannot always avoid starvation, an extreme form of unfairness [70]. Another line of congestion control proposals, focusing on congestion control in cellular networks, has sought to implement CCAs that can cope with high variability in link capacities [71, 72] while providing low delays. Since these proposals target cellular networks, which implement per-user isolation, they are not concerned with fairness.

If this paper's hypothesis proves to be true and most flows today are isolated from each other, CCA designers should deprioritize concerns about starvation and fairness. Instead, future CCAs should resemble the second group of CCAs, which focus on coping with bandwidth variability while navigating the trade-off between self-inflicted delay and link underutilization. Indeed, as discussed in §2.3, links with variable bandwidth are likely to become more common as the growth of mobile networks continues; e.g., even future fiber networks could embrace bandwidth variability [73].

This is not to say that CCAs would become simpler or irrelevant. For example, prior analytical work has shown [74] that even without considering fairness, a given CCA still cannot simultaneously satisfy efficiency and loss-minimization goals. Further, Schapira and Winstein argued [75] that the fundamental tenets of congestion control remain unresolved, e.g., whether CCAs should attempt to model the network, use hill-climbing techniques, etc.

5.2 Contention on Alternate Metrics

Even if CCAs no longer determine flows' bandwidth allocations in the Internet today, they might contend on other metrics such as latency and jitter. For example, bursty traffic can vary the instantaneous bandwidth and delay other flows on the same link observe, even if the link uses fair queueing to isolate flows from each other. These variations in delay, i.e., jitter, can degrade QoE for applications such as live video streaming that perform best with consistently low delay. The precise mechanism the operator uses to perform bandwidth shaping affects the way flows contend for low jitter. For example, one popular method of bandwidth shaping is the token-bucket filter, in which a flow accrues "tokens" to send bytes at a fixed rate but can consume these tokens arbitrarily quickly once they are granted; the resulting bursty transmission can cause jitter. We leave for future work the question of designing CCAs that are friendly on jitter and other non-throughput metrics.

5.3 How Should We Model the Internet?

Internet users have long operated under a model of the Internet that specifies best-effort packet delivery combined with statistical multiplexing mediated by fair interaction dynamics. Building effective CCAs benefits from such a useful network model [75]. Indeed, a 30-year-strong line of research work has sought to understand and analyze these interactions, starting with Chiu and Jain's AIMD and ongoing with theoretical analyses and controlled experiments [2, 70, 74]. If this paper's hypothesis holds, opaque traffic shaping mechanisms governed by operator policies and economic arrangements, rather than CCA fairness dynamics, govern bandwidth allocations today. As an analogy, the Gao-Rexford model [76] of interdomain routing practices has proved useful despite its practical inaccuracy for *parts* of the Internet; however, if it proved not representative of *most* of the Internet, it would have to evolve to remain useful.

Thus, how should our model of congestion control evolve to incorporate the observation that CCA dynamics do not affect bandwidth allocations? Clearly, it should incorporate some notion of flow isolation and bandwidth shaping. Second, the unit of bandwidth contention would no longer be an individual flow but rather an economic arrangement that determines a network's bandwidth-shaping policy. A recent HotNets paper proposed one potential model, "Recursive Congestion Shares" [77], rooted in the Internet's existing economic arrangements. Unfortunately, we cannot know how accurate this model or any other model is without data from network operators. We leave the development and validation of such models to future work.

Acknowledgements

We thank the anonymous HotNets reviewers for their comments. This work was supported by NSF grants 1704941, 2201489, 2212390, and 2242502, and by funding from Intel, IBM, VMware, Ericsson, and Google.

References

[1] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation-Based Congestion Control for Unicast Applications. *ACM SIGCOMM CCR*, 30(4):43–56, 2000. 1

[2] Ranysha Ware, Matthew K. Mukerjee, Srinivasan Seshan, and Justine Sherry. Modeling BBR's Interactions with Loss-Based Congestion Control. In *IMC*, 2019. 1, 5.3

[3] Sally Floyd. Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-Way Traffic. In *SIGCOMM*, 1991. 1

[4] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. A Quantitative Measure of Fairness and Discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 21, 1984. 1

[5] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *SIGCOMM*, 1989. doi: 10.1145/75246.75248. 1

[6] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew J. Holliman, Gary Baldus, Marcus Hines, Taeeun Kim, Ashok Narayanan, Ankur Jain, Victor Lin, Colin Rice, Brian Rogan, Arjun Singh, Bert Tanaka, Manish Verma, Puneet Sood, Muhammad Mukarram Bin Tariq, Matt Tierney, Dzevad Trumic, Vytautas Valancius, Calvin Ying, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *SIGCOMM*, 2017. 1, 2.2

[7] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan Katz-Bassett, Harsha V. Madhyastha, Ítalo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In *SIGCOMM*, 2017.

[8] Rachee Singh, Sharad Agarwal, Matt Calder, and Victor Bahl. Cost-Effective Cloud Edge Traffic Engineering with CASCARA. In *NSDI*, 2021. 1, 2.2

[9] Cardona Restrepo, Juan Camilo and Stanojevic, Rade. A History of an Internet Exchange Point. *SIGCOMM CCR*, 42(2):58–64, 2012. 1, 2.1

[10] Christophe Diot, Lai Yi Ohlsen, Matt Mathis, and Phillipa Gill. M-Lab: User Initiated Internet Data for the Research Community. *SIGCOMM CCR*, 2022. 2, 3, 3.1

[11] John Nagle. Congestion Control in IP/TCP Internetworks. RFC 896, 1984. 2.1

[12] Van Jacobson and Michael J. Karels. Congestion Avoidance and Control. In *SIGCOMM*, 1988. 2.1

[13] Dah-Ming Chiu and Raj Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Comput. Netw. ISDN Syst.*, 17(1), 1989. doi: 10.1016/0169-7552(89)90019-6. 2.1

[14] Feldmann, Anja and Gasser, Oliver and Lichtblau, Franziska and Pujol, Enric and Poese, Ingmar and Dietzel, Christoph and Wagner, Daniel and Wichtlhuber, Matthias and Tapiador, Juan and Vallina-Rodriguez, Narseo and Hohlfeld, Oliver and Smaragdakis, Georgios. The Lockdown Effect: Implications of the COVID-19 Pandemic on Internet Traffic. In *IMC*, 2020. 2.1, 2.2

[15] Udit Paul, Vinothini Gunasekaran, Jiamo Liu, Tejas N. Narechania, Arpit Gupta, and Elizabeth Belding. Decoding the Divide: Analyzing Disparities in Broadband Plans Offered by Major US ISPs. https://arxiv.org/abs/2302.14216, 2023. 2.1

[16] Flach, Tobias and Papageorge, Pavlos and Terzis, Andreas and Pedrosa, Luis and Cheng, Yuchung and Karim, Tayeb and Katz-Bassett, Ethan and Govindan, Ramesh. An Internet-Wide Analysis of Traffic Policing. In *SIGCOMM*, 2016. 2.1

[17] Li, Fangfan and Niaki, Arian Akhavan and Choffnes, David and Gill, Phillipa and Mislove, Alan. A Large-Scale Analysis of Deployed Traffic Differentiation Practices. In *SIGCOMM*, 2019. 2.1

[18] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. Achieving High Utilization with Software-Driven WAN. In *SIGCOMM*, 2013. 2.1

[19] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. B4: Experience With a Globally-Deployed Software Defined WAN. In *SIGCOMM*, 2013. 2.1

[20] Vimalkumar Jeyakumar, Mohammad Alizadeh, David Mazières, Balaji Prabhakar, Albert Greenberg, and Changhoon Kim. EyeQ: Practical network performance isolation at the edge. In *NSDI*, 2013. 2.1

[21] Umesh Krishnaswamy, Rachee Singh, Paul Mattes, Paul-Andre C Bissonnette, Nikolaj Bjørner, Zahira Nasrin, Sonal Kothari, Prabhakar Reddy, John Abeln, Srikanth Kandula, Himanshu Raj, Luis Irun-Briz, Jamie Gaudette, and Erica Lan. OneWAN is Better Than Two: Unifying a Split WAN Architecture. In *NSDI*, 2023. 2.1

[22] Alok Kumar, Sushant Jain, Uday Naik, Nikhil Kasinadhuni, Enrique Cauich Zermeno, C. Stephen Gunn, Jing Ai, Björn Carlin, Mihai Amarandei-Stavila, Mathieu Robin, Aspi Siganporia, Stephen Stuart, and Amin Vahdat. BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing. In *SIGCOMM*, 2015. 2.1

[23] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, et al. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *SIGCOMM*, 2017. 2.1

[24] Akshay Narayan, Frank Cangialosi, Deepti Raghavan, Prateesh Goyal, Srinivas Narayana, Radhika Mittal, Mohammad Alizadeh, and Hari Balakrishnan. Restructuring Endpoint Congestion Control. In *SIGCOMM*, 2018. 2.1

[25] Chiu, Yi-Ching and Schlinker, Brandon and Radhakrishnan, Abhishek Balaji and Katz-Bassett, Ethan and Govindan, Ramesh. Are We One Hop Away from a Better Internet? In *IMC*, 2015. 2.2

[26] Vern Paxson. End-to-End Internet Packet Dynamics. In *SIGCOMM*, 1997. 2.2

[27] Todd Arnold, Jia He, Weifan Jiang, Matt Calder, Italo Cunha, Vasileios Giotsas, and Ethan Katz-Bassett. Cloud Provider Connectivity in the Flat Internet. In *IMC*, 2020. 2.2

[28] T. Böttger, G. Antichi, E.L. Fernandes, R. Lallo, M. Bruyere, S. Uhlig, and I. Castro. The Elusive Internet Flattening: 10 Years of IXP Growth. In *RIPE 78*, 2018.

[29] Petros Gigis, Matt Calder, Lefteris Manassakis, George Nomikos, Vasileios Kotronis, Xenofontas Dimitropoulos, Ethan Katz-Bassett, and Georgios Smaragdakis. Seven Years in the Life of Hypergiants' Off-Nets. In *SIGCOMM*, 2021. 2.2

[30] Yin Zhang, Lee Breslau, Vern Paxson, and Scott Shenker. On the Characteristics and Origins of Internet Flow Rates. In *SIGCOMM*, 2002. 2.2

[31] Iqbal, Hassan and Khalid, Ayesha and Shahzad, Muhammad. Dissecting Cloud Gaming Performance with DECAF. 5(3), 2021. 2.2

[32] Xinlei Yang, Hao Lin, Zhenhua Li, Feng Qian, Xingyao Li, Zhiming He, Xudong Wu, Xianlong Wang, Yunhao Liu, Zhi Liao, Daqiang Hu, and Tianyin Xu. Mobile Access Bandwidth in Practice: Measurement, Analysis, and Implications. In *SIGCOMM*, 2022. 2.2, 3.1

[33] João Taveira Araújo, Raúl Landa, Richard G. Clegg, George Pavlou, and Kensuke Fukuda. A Longitudinal Analysis of Internet Rate Limitations. In *INFOCOM*, 2014. 2.2

[34] Frank Cangialosi, Akshay Narayan, Prateesh Goyal, Radhika Mittal, Mohammad Alizadeh, and Hari Balakrishnan. Site-to-Site Internet Traffic Control. In *EuroSys*, 2021. 2.3, 4

[35] Ammar Tahir and Radhika Mittal. Enabling Users to Control their Internet. In *NSDI*, 2023. 2.3, 4

[36] Jay Chen, Janardhan Iyengar, Lakshminarayanan Subramanian, and Bryan Ford. TCP Behavior in Sub-Packet Regimes. In *SIGMETRICS*, 2011. 2.3

[37] Thomas Pötsch, Paul Schmitt, Jay Chen, and Barath Raghavan. Helping the Lone Operator in the Vast Frontier. In *HotNets*, 2016. 2.3

[38] Hasan, Shaddi and Barela, Mary Claire and Johnson, Matthew and Brewer, Eric and Heimerl, Kurtis. Scaling Community Cellular Networks with Community Cellular Manager. In *NSDI*, 2019. 2.3

[39] Talal Ahmad, Ranveer Chandra, Ashish Kapoor, Michael Daum, and Eric Horvitz. Wi-Fly: Widespread Opportunistic Connectivity via Commercial Air Transport. In *HotNets*, 2017. 2.3

[40] Daniel Perdices, Gianluca Perna, Martino Trevisan, Danilo Giordano, and Marco Mellia. When Satellite is All You Have: Watching the Internet from 550 Ms. In *IMC*, 2022. 2.3

[41] Debopam Bhattacherjee, Waqar Aqeel, Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. Gearing up for the 21st Century Space Race. In *HotNets*, 2018. 2.3

[42] Simon Kassing, Debopam Bhattacherjee, André Baptista Águas, Jens Eirik Saethre, and Ankit Singla. Exploring the "Internet from Space" with Hypatia. In *IMC*, 2020. 2.3

[43] Starlink. Starlink Fair Use Policy. https://www.starlink.com/legal/documents/DOC-1134-82708-70, 2023. 2.3

[44] Shaddi Hasan, Yahel Ben-David, Max Bittman, and Barath Raghavan. The Challenges of Scaling WISPs. In *DEV*, 2015. 2.3

[45] Bilal Saleem, Paul Schmitt, Jay Chen, and Barath Raghavan. Beyond the Trees: Resilient Multipath for Last-Mile WISP Networks. In *arXiv*, volume abs/2002.12473, 2020. URL https://arxiv.org/abs/2002.12473. 2.3

[46] Matthew Luckie, Amogh Dhamdhere, David Clark, Bradley Huffaker, and kc claffy. Challenges in Inferring Internet Interdomain Congestion. In *IMC*, 2014. 2.3

[47] Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. Data Center TCP (DCTCP). In *SIGCOMM*, 2010. 2.3

[48] Radhika Mittal, Vinh The Lam, Nandita Dukkipati, Emily Blem, Hassan Wassel, Monia Ghobadi, Amin Vahdat, Yaogong Wang, David Wetherall, and David Zats. TIMELY: RTT-Based Congestion Control for the Datacenter. In *SIGCOMM*, 2015.

[49] Gautam Kumar, Nandita Dukkipati, Keon Jang, Hassan M. G. Wassel, Xian Wu, Behnam Montazeri, Yaogong Wang, Kevin Springborn, Christopher Alfeld, Michael Ryan, David Wetherall, and Amin Vahdat. Swift: Delay is Simple and Effective for Congestion Control in the Datacenter. In *SIGCOMM*, 2020.

[50] Yibo Zhu, Haggai Eran, Daniel Firestone, Chuanxiong Guo, Marina Lipshteyn, Yehonatan Liron, Jitendra Padhye, Shachar Raindel, Mohamad Haj Yahia, and Ming Zhang. Congestion Control for Large-Scale RDMA Deployments. In *SIGCOMM*, 2015.

[51] Yuliang Li, Rui Miao, Hongqiang Harry Liu, Yan Zhuang, Fei Feng, Lingbo Tang, Zheng Cao, Ming Zhang, Frank Kelly, Mohammad Alizadeh, and Minlan Yu. HPCC: High Precision Congestion Control. In *SIGCOMM*, 2019. 2.3

[52] Vladimir Olteanu, Haggai Eran, Dragos Dumitrescu, Adrian Popa, Cristi Baciu, Mark Silberstein, Georgios Nikolaidis, Mark Handley, and Costin Raiciu. An Edge-Queued Datagram Service for All Datacenter Traffic. In *NSDI*, 2022. 2.3

[53] Prateesh Goyal, Preey Shah, Kevin Zhao, Georgios Nikolaidis, Mohammad Alizadeh, and Thomas E. Anderson. Backpressure Flow Control. In *NSDI*, 2022. 2.3

[54] Prateesh Goyal, Akshay Narayan, Frank Cangialosi, Srinivas Narayana, Mohammad Alizadeh, and Hari Balakrishnan. Elasticity Detection: A Building Block for Internet Congestion Control. In *SIGCOMM*, 2022. URL https://doi.org/10.1145/3544216.3544221. 3, 3.2, 5.1

[55] Ayush Mishra, Xiangpeng Sun, Atishya Jain, Sameer Pande, Raj Joshi, and Ben Leong. The Great Internet TCP Congestion Control Census. *SIGMETRICS*, 2019. 3.1

[56] Adithya Abraham Philip, Ranysha Ware, Rukshani Athapathu, Justine Sherry, and Vyas Sekar. Revisiting TCP Congestion Control Throughput Models & Fairness Properties At Scale. In *IMC*, 2021. 3.1

[57] Xinlei Yang, Xianlong Wang, Zhenhua Li, Feng Qian, Liangyi Gong, Rui Miao, Yunhao Liu, and Tianyin Xu. Fast and Light Bandwidth Testing for Internet Users. In *NSDI*, 2021. 3.1

[58] Matt Mathis and Mark Allman. A Framework for Defining Empirical Bulk Transfer Capacity Metrics. RFC 3148, 2001. 3.1

[59] Measurement Lab. The M-Lab NDT data set. https://measurementlab.net/tests/ndt, (2009-02-11 – 2015-12-21). 3.1

[60] Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective Review of Offline Change Point Detection Methods. *Signal Processing*, 167: 107299, 2020. 3.1

[61] Venkat Arun and Hari Balakrishnan. Copa: Practical Delay-Based Congestion Control for the Internet. In *NSDI*, 2018. 3.2, 5.1

[62] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. Mahimahi: Accurate Record-and-Replay for HTTP. In *USENIX ATC*, 2015. 3.2

[63] Amogh Dhamdhere, David D. Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky K. P. Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C. Snoeren, and Kc Claffy. Inferring Persistent Interdomain Congestion. In *SIGCOMM*, 2018. 4

[64] Rodérick Fanou, Francisco Valera, and Amogh Dhamdhere. Investigating the Causes of Congestion on the African IXP Substrate. In *IMC*, 2017. 4

[65] H. Balakrishnan, H. S. Rahul, and S. Seshan. An Integrated Congestion Management Architecture for Internet Hosts. In *SIGCOMM*, 1999. 4

[66] Danyang Zhuo, Qiao Zhang, Vincent Liu, Arvind Krishnamurthy, and Thomas Anderson. Rack-Level Congestion Control. In *HotNets*, 2016. 4

[67] Sundararajan Renganathan, Venkata N. Padmanabhan, and Akshay Uttama Nambi. Rethinking Networking for "Five Computers". In *HotNets*, 2018. 4

[68] Ranysha Ware, Matthew K. Mukerjee, Srinivasan Seshan, and Justine Sherry. Beyond Jain's Fairness Index: Setting the Bar For The Deployment of Congestion Control Algorithms. In *HotNets*, 2019. 4

[69] Mo Dong, Tong Meng, Doron Zarchy, Engin Arslan, Yossi Gilad, Brighten Godfrey, and Michael Schapira. PCC Vivace: Online-Learning Congestion Control. In *NSDI*, 2018. 5.1

[70] Venkat Arun, Mohammad Alizadeh, and Hari Balakrishnan. Starvation in End-to-End Congestion Control. In *SIGCOMM*, 2022. 5.1, 5.3

[71] Keith Winstein, Anirudh Sivaraman, and Hari Balakrishnan. Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks. In *NSDI*, 2013. 5.1

[72] Yasir Zaki, Thomas Pötsch, Jay Chen, Lakshminarayanan Subramanian, and Carmelita Görg. Adaptive Congestion Control for Unpredictable Cellular Networks. In *SIGCOMM*, 2015. 5.1

[73] Rachee Singh, Manya Ghobadi, Klaus-Tycho Foerster, Mark Filer, and Phillipa Gill. RADWAN: Rate Adaptive Wide Area Network. In *SIGCOMM*, 2018. 5.1

[74] Doron Zarchy, Radhika Mittal, Michael Schapira, and Scott Shenker. Axiomatizing Congestion Control. In *SIGMETRICS*, 2019. 5.1, 5.3

[75] Michael Schapira and Keith Winstein. Congestion-Control Throwdown. In *HotNets*, 2017. 5.1, 5.3

[76] Lixin Gao and J. Rexford. Stable Internet Routing Without Global Coordination. *IEEE/ACM Trans. Netw.*, 2001. 5.3

[77] Lloyd Brown, Ganesh Ananthanarayanan, Ethan Katz-Bassett, Arvind Krishnamurthy, Sylvia Ratnasamy, Michael Schapira, and Scott Shenker. On the Future of Congestion Control for the Public Internet. In *HotNets*, 2020. 5.3