# Illuminating the hidden challenges of data-driven CDNs

Theophilus A. Benson
Carnegie Mellon University

## ABSTRACT

While Data-driven CDNs have the potential to provide unparalleled performance and availability improvements, they open up an intricate and exciting tapestry of previously unaddressed problems. This paper highlights these problems, explores existing solutions, and identifies open research questions for each direction. We, also, present a strawman approach, Guard-Rails, that embodies preliminary techniques that can be used to help safeguard data-driven CDNs against the identified perils.

## CCS CONCEPTS

• **Networks** → **Network management**; **Network protocol design**; • **Computing methodologies** → **Ensemble methods**.

## KEYWORDS

Data driven networking, Safe machine learning.

**ACM Reference Format:**
Theophilus A. Benson. 2023. Illuminating the hidden challenges of data-driven CDNs. In *3rd Workshop on Machine Learning and Systems (EuroMLSys '23), May 8, 2023, Rome, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3578356.3592574

## 1 INTRODUCTION

There is a growing movement to adopt data-driven networking techniques to improve the performance and availability of traditional CDNs by Akamai [54], Facebook [27, 46], Microsoft [10], Yahoo [3], Baidu [39, 40] (i.e., data-driven CDNs [23]).

These Data-driven CDNs (DD-CDNs) replace traditional networking heuristics (e.g., cache eviction, ABR algorithms) and management techniques with data-driven algorithms and machine learning techniques (e.g., Multi-armed Bandit-based [49] bitrate selection [22, 36, 48] or DeepRL driven Congestion control [40]). A distinguishing feature of DD-CDNs is their ability to collect data from multiple users, perform analysis to generate inferences and use them to determine optimal behavior for individual users. This dependency on data results in new security, fairness, and convergence challenges within the CDN context (specifically DD-CDN) which *few have fully explored and solved*.

In this paper, we take the first step to better understand the emerging challenges that arise due to the growing adoption of these data-driven paradigms, identify practical and valuable designs to address these challenges, and explore holistic architectures for solving these challenges under practical deployment scenarios. Most importantly, we illustrate the limitations of existing approaches by applying them to a multi-armed bandit-based DD-CDN framework [38] and analyze the system from data traces captured at a large multi-billion user online social networking site. In particular, we observe the following challenges.

- **Security:** the use of a data-driven control loop opens the CDN to a new class of security attacks, e.g., data-pollution – where an attacker pollutes the model by sending malicious inputs which forces the CDN to behave in a suboptimal fashion;
- **Unfair Resource Allocation:** employing heterogeneous policies impacts connections competing for resources both locally within the same CDNs (e.g., a connection may starve out others [18]) or globally across the internet (e.g., a connection may unfairly dominate the bottleneck link [43, 52]);

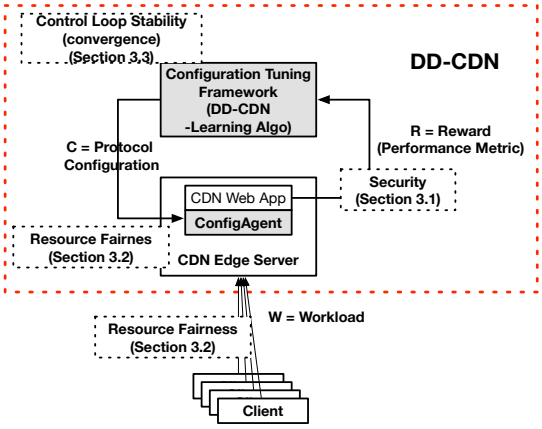


**Figure 1: Canonical DD-CDN architecture.**


*EuroMLSys '23, May 8, 2023, Rome, Italy*

ACM ISBN 979-8-4007-0084-2/23/05.
https://doi.org/10.1145/3578356.3592574

- **Convergence:** dynamically interacting and self-adaptive closed control loops introduce novel stability and convergence challenges (or lack thereof).

While security and bias in data-driven networking [16, 28, 33, 45, 55] has been previously identified, prior works often focus on problems that arise when analyzing one CDN. Similarly, stability and fairness have been identified in the broader multi-party setting but with limited success — either providing fairness [1] at the cost of significant performance [4, 5] or providing fairness and stability for a limited set of knobs [15] (e.g., transport – sending rate). They generally overlook broader issues arising from interactions between a broad range of control loops (both traditional and DD-CDN) across a broad set of actions (configuration knobs). More importantly, they do not present concrete and practically deployable solutions for the CDN space.

In exploring the design space of existing solutions, we discover that traditional machine learning techniques for addressing security, fair allocation, and convergence provide limited benefits. Specifically, ML-security techniques provide limited benefits because of the inherent noisiness and non-stationary within the CDN domain. Moreover, their minimal use of rich domain knowledge unfairly limits them. On the other hand, cutting-edge ML techniques for global and competitive scenarios require cooperation which is unrealistic in some CDN scenarios, e.g., Multi-agent Reinforcement learning or provided limited effectiveness (generative adversarial network (GAN)). The former requires joined learning between the CDNs, which is unreasonable in specific settings, given their inherent competition. In contrast, the latter assumes that performance metrics may be untenable in highly dynamic environments. We illustrate these issues by analyzing the impact of these design choices on a representative multi-armed bandit framework using data from a large global CDN.

Our key observation is that these problems arise due to unfettered trust in different entities (clients (security), local CDN algorithms (fairness), and competing CDN algorithms (stability)) and their interactions. We propose changes to DD-CDN frameworks to include techniques that protect the core DD-CDN-learning algorithm from attackers and convergence issues due to external control loops and support a more fair allocation of resources by dynamically constraining the learning algorithm's action space.

This work aims to identify challenges, highlight limitations of existing approaches, and sketch out potential solutions for various deployment scenarios. To this end, our goal is not to develop an entirely novel framework but to investigate appropriate components and illustrate directions for integrating them into a practical approach. Our key contributions are:

- We identify three deployment challenges that arise when data-driven control loops are used to manage large-scale CDNs. We demonstrate the impact of these challenges through empirical analysis of several data-driven frameworks with production traces.
- We then demonstrate that popular machine learning techniques, e.g., generative adversarial networks, provide minimal impact in addressing these issues.
- We conclude with a proof-of-concept sketch for a solution, Guard-Rails, that addresses these challenges by using domain-specific algorithms to place constraints on the inputs to the DDN technique and the set of available actions that the DDN technique can explore. We, also, identify open research challenges.

## 2 DD-CDN BACKGROUND

This section provides a brief overview of existing CDNs and approaches to designing DD-CDN.

### 2.1 CDN Background

To understand the potential role of data-driven enhancements, we provide a brief overview of the interactions between CDNs management techniques and networking protocols and the CDN's clients. Initially, a client needs to identify a CDN server to connect to. Clients are assigned to CDN servers based on latency, load, cache content, and many other factors [10, 13]. Next, the client initiates a connection to the server, and both parties agree on the network protocols (transport and HTTP layer) to use. The choice of protocol often depends on a client's networking conditions [3] and historical data [40]. Finally, the client retrieves content from the CDN. Content is either received from the CDN's local cache or retrieved from external servers (called *origin*). The CDN employs a sophisticated policy to determine what items to cache – such policies may take into account client latency [56], backend performance [9], content popularity [8]. In sending the data to the client, the CDN has several choices for its egress path. Recently, much effort has gone into engineering egress traffic routing for large hyperscalers [54].

In these CDNs, there are many opportunities for moving beyond existing heuristics to data-driven approaches, specifically in CDN server selection (Targeting), protocol selections (Transport, HTTP), caching policies (Caching), and egress traffic engineering (Routing). Unsurprisingly, we have seen many academic and recent industrial efforts shift towards data-driven approaches for each of these dimensions. In Table 1, we briefly summarize these efforts at the Transport and Application layers.

| Protocol Knobs | Learning Technique | |
|---|---|---|
| | Multi-Armed Bandit | DeepRL ( No-Regret) |
| Transport | [23, 38] | [20, 29, 40**? ? ?** ] |
| Application | [24**? ?** ] | [8, 32**?** ] |

**Table 1: A list of representative DD-CDN techniques with the type of learning models used highlighted.**

## 2.2 Data-Driven CDNs (DD-CDNs)

A DD-CDN consists of several key defining components (illustrated in Figure 1): (1) a learning framework, which infers the appropriate configuration parameters to configure for a client (or more specifically, the client's workload ($WC_N$)) – the learning framework is usually run globally, (2) a configuration agent, which runs on the CDN's servers or routers and provides a flexible, low-overhead interface for dynamically tuning various configuration parameters – the configuration agent is usually distributed with one agent on each CDN server or router, to provide control over local configuration parameters, and (3) CDN telemetry, which captures the state of the current CDN applications and the environment (e.g., the network or the server) on which the application runs. The CDN telemetry also captures the reward (or performance metric) from client responses – either implicitly from the TCP/HTTP level ACKs or explicitly through application signals.

The learning framework usually takes as input the system's current state and creates as output the set of configurations that optimize the performance metric. Most learning frameworks group users into clusters, $WC_N$ in Figure 2, to amortize and scale learning efforts. The learning algorithm can either be trained offline using modeled data or trained online using an exploitation/exploration-based system. The machine learning techniques are often run in a **centralized** fashion, and the clients are **trusted** to provide accurate results. Finally, each CDN will independently run its own control loop due to trust and economic reasons, potentially using different learning algorithms. As more CDNs adopt data-driven approaches (Figure 2), the learning algorithms for these systems will start to interact with each other and potentially compete with non-DD-CDNs (i.e., traditional CDNs using hand-tuned configurations, $Trad - CDN_B$ in Figure 2). Moreover, we expect that DD-CDNs will start to interact and compete with non-DD-CDNs (CDNs using hand-tuned configurations) at various bottleneck locations.

Several interesting challenges arise as a result of the growing adoption of machine learning techniques for managing CDNs. Next, we highlight three such challenges:

- **Security:** In many of these data-driven approaches, users are often grouped based on workloads or environmental characteristics, and the performance metric returned by a user impacts the model's behavior for other members in the group – while some techniques group users based on IP-prefixes [3, 40], others [22] group on more extensive features (e.g., device type and city). For example, in Figure 2, workload clusters, $WC_1$ and $WC_2$, both have malicious users in their groups, and these users can influence the CDN server's behavior, thus impacting server utilization or end-user experience.
- **Resource Allocation (Fairness):** While the use of tailored configurations enhances the CDN's performance, anecdotal evidence suggests that many configuration choices can lead to unfairness both locally within the CDN (e.g., unfair use of CPU) or globally on the internet (e.g., unfair bandwidth allocations). For example, in Figure 2, $Trad - CDN_B$ will not dynamically adapt to bottleneck conditions at $S_2$ while the DD-CDNs will, thus ensuring that the DD-CDNs get a higher share of the bandwidth.
- **Stability (and Equilibrium):** Finally, a key concern when deploying multiple interacting control loops is stability or convergence. This is also problematic in the DD-CDN scenario. Specifically, when multiple DD-CDNs interact over a bottleneck link (i.e., $CDN_A$ and $CDN_B$ over router $S_2$), it is unclear if their control loops will converge, under what conditions they will converge, and the general characteristics of the converged scenario. For example, in Figure 2, $DD - CDN_A$, and $DD - CDN_C$ will tune their configuration in response to changes in the environment and may also respond to each other's dynamic changes.

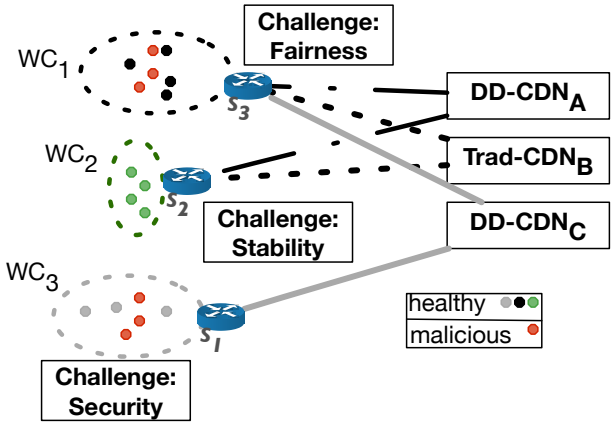


**Figure 2: Deployment Challenges for DD-CDN.**

# 3 OPEN CHALLENGES IN REALIZING DD-CDNS

Next, we discuss open challenges in protecting DD-CDN from data-poisoning (Section 3.2), discuss the implications

of DDNs for resource allocation and internet fairness (Section 3.3), and conclude by highlighting the impact of DD-CDNs on global internet health (e.g., stability or convergence) (Section 3.4).

## 3.1 DataSet, Methodology, and Setup

We empirically analyze the challenges within the context of a recently proposed Multi-Armed Bandit-based DD-CDN, Configanator [38] [NSDI'22], which has been slightly modified to specifically tune the transport layer. In particular, Configanator employs a contextual multi-armed bandit to learn configurations and uses k-means to group clients into clusters. To analyze Configanator, we re-use one of the traces from the original paper: traces from a large global CDN that hosts an online social networking site with over a billion users (GlobalCDN). We analyze our use-case on a day's worth of traces from said GlobalCDN, which constitutes several million requests across six continents. This data-set provides representative and heterogeneous user interactions regarding last-mile connections, end-user devices, and online service type. We note that while we focus on one use-case, our approach and insights readily generalize to alternative approaches based on variants of traditional reinforcement learning (i.e.,Multi-Armed Bandit [23, 24] and deep reinforcement learning [20, 29, 32, 40]).

## 3.2 Security: Poisoning of DD-CDNs

**Background:** Machine learning (ML) techniques are vulnerable to data-poisoning attacks, wherein an attacker can influence the decisions of the machine learning model by injecting carefully crafted data. Many studies [16, 28, 33, 45, 55] have shown that in settings such as DD-CDNs, data-poisoning can easily change the behavior of the ML-Model.

**Empirical Analysis:** To illustrate this point, in Figure 3, we present the results of attacking our DD-CDN framework [38, 40]. In this scenario, we assume that the attacker can compromise a subset of legitimate clients, e.g., via IoT compromises, and alter the spacing between ACK packets and the contents of the HTTP response messages, e.g., OnLoad information – these changes will alter the "reward" or metric given to the learning algorithm. We vary the number of endpoints an attacker can compromise from 0.3% to 10%. The figure shows that attacks can significantly impact the model's effectiveness by simply perturbing the inputs. More importantly, we observe that the greater the amount of attacker traffic, e.g., 0.3% compared with 10%, the higher its impact on the model. Most importantly, we observe that these attacks significantly impact the entire distribution with significant consequences for tail performance, a significant concern for modern CDNs [18].

Existing defenses [16, 50] build on the insight that: (1) attackers try to influence the ML-model by introducing data points that either have a distinctly different distribution than the underlying data [16] or (2) attackers are malicious clients [50]. Thus detection can be achieved by performing anomaly detection on the clients directly [50] or anomaly detection on the distributions of the data [16]. The DD-CDN scenario presents distinctly different constraints: Due to several sources, e.g., last-mile issues, honest clients can generate unrepresentative data points [22]. For example, a phone with a low battery may have a page load time that does not represent normal conditions. Thus, honest clients may occasionally but unknowingly poison the data set. Moreover, other more targeted techniques require special hardware, i.e., SGX [21], at the devices creating the data (i.e., clients); however, most CDN operators cannot change the software and hardware of the clients. Unfortunately, the attackers are given significantly more freedom because of the lack of control over the endpoints. Thus, practical solutions must detect poisoning and Sybils without client-side cooperation.

Next, we explore one of the most recently proposed and most promising approaches for improving the security of data-driven networking techniques, which uses Conditional Generative Adversarial Networks (CGANs) to detect adversarial attacks [31]. In Figure 3, we present results of using the more advanced CGANs and simpler GAN (Generative Adversarial Networks) to address these data-poisoning attacks; we observe that both techniques do improve performance but that they do not effectively eliminate the problem. Additionally, as highlighted by prior work, we observe the CGAN does much better than GANs.

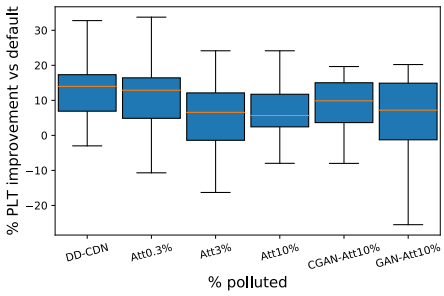


**Figure 3: Varying the level of data pollution (i.e., # of compromised clients) reduces the accuracy and performance of the DD-CDN framework (i.e., MAB+NC). The MAB+NC shows initial DD-CDN performance without any attacks. The other bars show how varying levels of attack impact the model's peformance.**

**Challenge #1:** The lack of control over client-end points introduces unique challenges because of the noise in the data and the potential for adversarial clients.

**Opportunity #1:** Unlike general Machine learning-based defensive techniques for data pollution, within the networking domain, there are side channels through which we can reliably capture "hints". *We can use domain knowledge about the network or active measurements from the network to identify potentially malicious behavior.* For example, we can use RTT measurements or traceroutes to confirm, detect, or label potentially suspicious behavior. At a high level, this approach of using "hints" is similar to existing work on problem diagnosis and localization, which aim to detect and pinpoint the location of a problem. However, our problem is sufficiently different; while diagnosis seeks to identify groups of misbehaving clients, our goal is to determine further if such misbehavior is malicious or not.

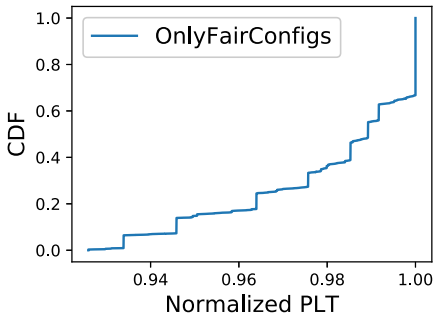


**Figure 4: performance of DD-CDN usecase without unfair Configuration knobs (e.g., BBR, high ICW values etc.).**

## 3.3 Resource Allocation: Global Fairness and Competing Objectives

**Background:** Today, most approaches to enabling DD-CDNs focus on largely solving a single objective: performance or availability. However, there are multiple unexplored consequences of using heterogeneous configurations, both locally on the servers and globally on the internet. Specifically, tuning the transport layer can lead to fairness issues in the WAN (Wide-Area Network). Where-as tuning any parameters, e.g., ABR or HTTP, can lead to different resource allocations on the server [18, 56]. More concretely, different parameters/options require different CPU/memory resources.

**Empirical Analysis:** To illustrate, we analyze the CPU overheads of different HTTP/TCP options for different connections. To do this, we set the workload (i.e., requests per second) and the network conditions (i.e., bandwidth, loss, RTT) to be fixed but vary the application's TCP/HTTP configuration knobs. We observe a 20-30% difference: intuitively, such differences are not surprising because the different configuration options will use different code paths, invariably leading to different resource footprints. Similarly, when emulating the wide-area network, we observe that the different configurations can lead to different bandwidth shares and unfairness. We do not quantify the network unfairness because significant work [11, 37, 43] has been done to explore, analyze, and quantify unfairness between different configurations.

A naive solution is to manually restrict and limit the configuration space statically. To better understand the implications of such a choice, in Figure 4, we explore the impact of re-running our DD-CDN use case but with only TCP-fair configuration knobs – as defined in [38]. *We observe that while there is some reduction in performance (2-7%), there is still a clear benefit to using these DD-CDN techniques, and the benefit arises because the top configurations being used are not necessarily the most aggressive.* In fact, prior work [19, 51] has shown that, counterintuitively, using newly recommended configuration knobs, e.g., BBR or higher ICW, can have negative consequences on performance (e.g., high ICW [19], spdy [51], BBR [11]).

**Challenge #2:** Unfortunately, statically restricting this configuration space is overly rigid and limits a DD-CDN's ability to dynamically adapt and evolve to the network's changing characteristics: we highlight that dynamically restricting this space and exploring suboptimal configurations can still yield significant benefits. However, there are several challenges to such a dynamic and automated approach. First, *visibility*, while a CDN can easily detect CPU or memory overheads, the CDN can not quantify the level of unfairness in the WAN because network level unfairness is a function of all flows sharing the bottleneck link: prior works have shown that the level of unfairness depends on the number of competing flows [11, 37, 52]. Second, *control*, while a CDN can dynamically alter the locally competing flows to ensure safe interactions, on a global scale, the CDN can not control other CDN's connections.

Both on the local and global scale, dynamically configuring and tuning to greedily maximize a performance objective is myopic and can have disastrous consequences. Additionally, extending existing reward maximization-based DD-CDN models to address this issue directly is non-trivial in the presence of competing DD-CDN models because these approaches do not operate effectively in non-stationary and adversarial environments created by competitive DD-CDN control loops.

**Opportunity #2:** Abstractly, the ideal solution limits the DD-CDN's configuration parameters and knobs to a subset that has been proven to be fair both locally and globally: for example, a subset of transport configuration parameters have been theoretically and empirically proven to be TCP-friendly [26, 52] (i.e., they are fair with TCP NewReno).

### 3.4 Stability: Global Interactions between CDNs

**Background:** We conclude by discussing the interactions between the various DD-CDN control loops. As discussed earlier, multiple CDNs are working towards different incarnations of DD-CDN control loops. A challenge that arises when multiple closed-looped networking systems [6, 17, 30] are interacting is convergence; the DD-CDN domain is no different. For example, the performance of a specific transport protocol (or transport configuration parameters) is a function of (1) the bottleneck link's network conditions and (2) the configurations of the other connections sharing the bottleneck link. This fact implies that multiple DD-CDN may analyze the current bottleneck and make the optimal decisions for this condition; however, if any DD-CDN changes its choices, this change may force other DD-CDNs to make changes accordingly. This cycle may continue *Ad nauseam*, with each CDN continually reacting to the other CDNs or to underlying changes in the network conditions or workloads.

**Empirical Analysis:** Motivated by these potential oscillations, next we attempt to understand if a "Nash equilibrium" exists when multiple DD-CDNs are interacting over the network. Specifically, we want to understand if they ever converge to some stable configurations, i.e., a Nash equilibrium, or if they are consistently oscillating between configurations. To do this, we re-use the simulator from Configanator [38] to simulate a scenario where two distinct CDNs with different websites (randomly selected from the top Alexa-top 1000) are employing our DD-CDN to tune the transport layer. We evaluate their interactions across many different networking conditions. To more directly characterize the equilibrium behavior, we generate and feed data from our simulator into a game-theoretic tool for characterizing nash-equilibrium [25]. In Figure 5, we present the results of this analysis: we observe that in a minority of situations (< 30%), there are situations where no equilibrium exists; however, in many situations, multiple different equilibria exist. *Also, we observe a correlation between the configuration in the equilibria and the set of fair configurations. Intuitively, equilibria consist of fairer configurations because there is less incentive to switch.*

**Challenge #3:** Understanding the equilibrium behavior of multiple interacting DD-CDN is an open challenge. The key issue lies in the distributed and decentralized nature in which each DD-CDN makes its decisions. Unfortunately, existing approaches to analyze and design DD-CDNs [2, 22, 48] often overlook this issue, and the simulation-based analysis do not include these interactions. Despite the lack of work, understanding equilibrium remains a significant problem, and more importantly, understanding the changes required to allow a

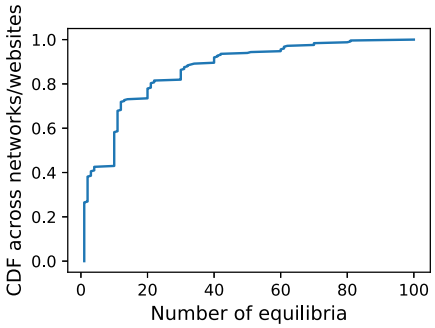


**Figure 5: Nash equilibrium analysis of two interacting DD-CDNs.**

DD-CDN model to converge quickly towards a safe equilibrium – where safety is defined as a function of the price of anarchy.

**Opportunity #3:** An essential step towards designing for equilibrium lies in understanding if a nash-equilibrium exists for modern DD-CDNs– lacking such an equilibrium, we would need to rethink the core algorithm underlying these DD-CDNs fundamentally. Motivated by the existence of such equilibria (Figure 5), we argue for exploring solutions inspired by prior work on routing and transport convergence [14, 44, 47]. Prior works have addressed this problem in one of two ways: first, by adding timers and by restricting the flexibility of the control loop [14, 44, 47] or second, my revisiting the model design to leverage regret minimization instead of reward maximization [7, 15, 41].

## 4 A PATH FORWARD: GUARD-RAILS

Our central argument is that the identified issues arise because of an overly open architecture that trusts clients (hence security issues), trusts the control loop (thus fairness issues), and trusts other CDNs (hence stability/convergence issues). We take the first step towards addressing these problems by introducing designs that limit the architecture's trust in clients, the learning algorithm's control loop, and other CDNs.

To address this, our system, Guard-Rails, retroactively improves a DD-CDN architecture by introducing a unique and practical data-driven ensemble that limits the implications of this over-trust. In designing our ensemble, our aim is to avoid overhauling existing DD-CDNs to use a new machine-learning algorithm but rather to propose a practical synthesis of existing techniques that allows DD-CDNs to continue to use their current machine-learning models. *Essentially, we provide the design of a CDN-specific data-driven pipeline that provides a unique synthesis of standard techniques to detect and prevent the problems under consideration (albeit in a non-conventional method). The key emphasis and motivation for building a pipeline atop standard techniques is*

*ensuring that the resulting pipeline addresses operational and deployment issues.*

## 4.1 Solution Design Space

Abstractly, we can address these challenges in a purely ML-driven fashion by using techniques like GAN to identify and filter attack traffic and by exploring recent work on multi-agent learning or regret minimization (no regret) models to address equilibrium and convergence. However, as shown in section 3 and supported by prior work [53], care must be taken when exploring purely ML-based solutions.

Alternatively, we can explore a fusion of an existing protocol and a data-driven approach. Recently, Orca [1] showed that such combinations could improve fairness and convergence. Unfortunately, such fusions have three drawbacks: first, such techniques inherit the rigidity and inflexibility of existing protocol (e.g., Orca inherits Cubic's limitations); second, such fusions can compromise the effectiveness of the ML techniques (e.g., Orca is less aggressive [4]); and third, to effectively address all challenges, we need to combine multiple protocols whose interactions are not clearly understood (e.g., Orca does not address security issues).

At the other end of the spectrum is a largely domain-driven approach, wherein domain-specific techniques protect the DD-CDN by limiting the inputs and restricting outputs while keeping the learning algorithm unchanged.

## 4.2 Guard-Rails' Architecture

Our approach explores the last design point wherein we leverage domain-specific techniques to limit and provide guardrails around the core learning framework. Our system, appropriately named Guard-Rails (Figure 6), limits the inputs and the set of actions/configuration options exposed to the learning algorithm. To do this, Guard-Rails introduces two key modules into an existing DD-CDN's ML-pipeline.

**security module:** The security module detects and filters out potentially malicious inputs from attackers using active measurements and passive data. In particular, Guard-Rails leverages the fact that modern CDNs collect measurements for availability reason [10] proactively. Similar to the core learning framework, we envision that the security module will group user sessions and perform anomaly detection and data-depollution independently for each group. We envision creating domain-specific predictive models that allow us to use low-level network telemetry (e.g., packet loss distribution and latency) to approximate and infer high-level application quality of experience metrics (e.g., GoodPut and page load times). Unlike existing domain-specific models, e.g., ECON [12], which try to provide accurate predictions, our goals are to provide general and approximate predictions of expected application-level metrics for arbitrary applications.

We note that designing unique models for each application across different conditions (as required by many existing domain-specific approaches) is prohibitively costly. Instead, we argue for exploring black-box models. We envision building on Bayesian Optimization, a non-parametric approach, which makes few assumptions about the underlying system. Interestingly, Bayesian optimization also provides a confidence interval that allows us to determine the likelihood that a sample is anomalous.

**stability module:** The stability module tries to limit the set of existing configurations to a set that promotes fairness and stability. The fundamental challenge with this module is that while fairness can be conservatively determined offline (TCPFriendliness), stability requires understanding the current dynamics and, thus, online exploration. Additionally, others [7, 15, 41] have successfully addressed stability by switching from reward maximizing to regret minimizing techniques; we note that in our experiments, regret minimization (e.g., exp3-bandit) provides stability but at the sacrifice of effectiveness. We observed, similar to others [15], that regret minimization schemes are slower to adapt and thus often have lower performance profiles.

Within our stability module, we note that equilibrium is easier to achieve because of the offline processing to eliminate unfair configurations. This provides two significant benefits that improve the performance of regret minimization schemes: (1) the search space is smaller, and (2) the equilibria discovered in Sec 3.4 consist of fair configurations.

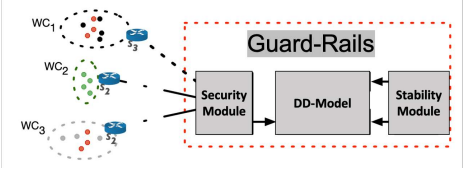


**Figure 6: Guard-Rails: novel framework for protecting DD-CDN.**

**Guard-Rails's Workflow:** Putting the components together, a DD-CDN control loop enhanced by Guard-Rails behaves as follows: First, the input data received from end-users go through a prefiltering step wherein the security module analyzes low-level metrics to determine the bound regions and eliminates data points outside this region as candidates for data poisoning attacks. Second, the "arms" or actions exposed to the learning model are limited by the stability module to the subset that improves fairness and promotes convergence.

# 5 DISCUSSION

Next, we discuss several open issues regarding Guard-Rails.

**Federated CDNs:** Our current work focuses on purely competitive scenarios; however, with the growth of over-the-top video and the desire for rich user experience, there are now CDN brokers [35] which control traffic and configurations across multiple CDNs. Such cooperative and collaborative settings introduce a new point in the design space. In particular, efforts to address stability and convergence can be made by exploring cooperative learning algorithms. A key question here is understanding the practicality and the fit of existing work on cooperative multi-agent learning to the DD-CDN domain.

**Scalability:** A central component of Guard-Rails involves active probing and the development of black box models for the security module and offline determination of fair-configs. This naturally raises questions about the scalability and overheads associated with building and maintaining such models at scale. We plan to explore system designs that minimize such overheads.

**Verifiable ML:** Our work focuses on keeping the machine learning core simple and developing guardrails around it. Others have explored techniques to introspect into the learning algorithm to verify [42] or interpret [34] it. We note that while we address orthogonal issues, such introspection could prove valuable in improving the scalability of our approach. For example, with Metis' [34] decision tree, we can focus on building Bayesian optimization models for each rule.

# 6 CONCLUSION

There is significant work on designing effective data-driven control loops for networks – a subset of this work focuses on CDNs. While most works focus on performance, only some address the practical challenges that arise when data-driven CDNs are productized. We highlight three key issues and provide empirical evidence of their impact: security, fairness, and stability. We show that such issues arise because of unfettered trust between the learning algorithms and the entire ecosystem. Although fairness and stability are often discussed, existing solutions are designed for specific scenarios [15] or provide limited effectiveness [1].

Motivated by our findings, we argue for a system that introduces guardrails to address these issues by limiting the trust placed on the ecosystem by the core learning frameworks within existing data-driven CDNs. In presenting our system, Guard-Rails, we aim to initiate a broader discussion on the management challenges underlying DD-CDN deployments.

# 7 ACKNOWLEDGMENTS

I thank the anonymous reviewers for their invaluable comments. I also thank Usama Naseer for providing access to his simulator, data-set and help getting experiments running. This work is supported by NSF grants CNS-1814285.

# REFERENCES

[1] Abbasloo, S., Yen, C.-Y., and Chao, H. J. Classic meets modern: A pragmatic learning-based congestion control for the internet. 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* (New York, NY, USA, 2020), SIGCOMM'20, Association for Computing Machinery, p. 632–647.

[2] Akhtar, Z., Nam, Y. S., Govindan, R., Rao, S., Chen, J., Katz-Bassett, E., Ribeiro, B., Zhan, J., and Zhang, H. Oboe: auto-tuning video abr algorithms to network conditions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (2018), ACM, pp. 44–58.

[3] Al-Fares, M., Elmeleegy, K., Reed, B., and Gashinsky, I. Overclocking the yahoo!: Cdn for faster web page loads. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (2011), ACM, pp. 569–584.

[4] Alan-JW. Competitiveness issues for orca. https://github.com/Soheil-ab/Orca/issues/10.

[5] Alan-JW. Three questions in model testing. https://github.com/Soheil-ab/Orca/issues/8.

[6] Alizadeh, M., Javanmard, A., and Prabhakar, B. Analysis of dctcp: Stability, convergence, and fairness. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, 2011), SIGMETRICS '11, Association for Computing Machinery, p. 73–84.

[7] Avramopoulos, I., Rexford, J., and Schapire, R. From optimization to regret minimization and back again. In *Proceedings of the Third Conference on Tackling Computer Systems Problems with Machine Learning Techniques* (USA, 2008), SysML'08, USENIX Association, p. 7.

[8] Berger, D. S. Towards lightweight and robust machine learning for cdn caching. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2018), HotNets '18, Association for Computing Machinery, p. 134–140.

[9] Berger, D. S., Berg, B., Zhu, T., Sen, S., and Harchol-Balter, M. Robinhood: Tail latency aware caching – dynamic reallocation from cache-rich to cache-poor. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)* (Carlsbad, CA, Oct. 2018), USENIX Association, pp. 195–212.

[10] Calder, M., Gao, R., Schröder, M., Stewart, R., Padhye, J., Mahajan, R., Ananthanarayanan, G., and Katz-Bassett, E. Odin: Microsoft's scalable fault-tolerant CDN measurement system. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)* (Renton, WA, Apr. 2018), USENIX Association, pp. 501–517.

[11] Cao, Y., Jain, A., Sharma, K., Balasubramanian, A., and Gandhi, A. When to use and when not to use bbr: An empirical analysis and evaluation study. In *Proceedings of the Internet Measurement Conference* (New York, NY, USA, 2019), IMC '19, ACM, pp. 130–136.

[12] Cao, Y., Nejati, J., Balasubramanian, A., and Gandhi, A. Econ: Modeling the network to improve application performance. In *Proceedings of the Internet Measurement Conference* (New York, NY, USA, 2019), IMC '19, Association for Computing Machinery, p. 365–378.

[13] Chen, F., Sitaraman, R. K., and Torres, M. End-user mapping: Next generation request routing for content delivery. *ACM SIGCOMM Computer Communication Review 45*, 4 (2015), 167–181.

[14] Cittadini, L., Battista, G. D., and Rimondini, M. On the stability of interdomain routing. *ACM Comput. Surv. 44*, 4 (Sept. 2012).

[15] Dong, M., Li, Q., Zarchy, D., Godfrey, P. B., and Schapira,

M. Pcc: Re-architecting congestion control for consistent high performance. In *NSDI* (2015), vol. 1, p. 2.

[16] FUNG, C., YOON, C. J. M., AND BESCHASTNIKH, I. Mitigating sybils in federated learning poisoning. *CoRR abs/1808.04866* (2018).

[17] GRIFFIN, T. G., AND WILFONG, G. An analysis of bgp convergence properties. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (New York, NY, USA, 1999), SIGCOMM '99, Association for Computing Machinery, p. 277–288.

[18] HELT, J., FENG, G., SESHAN, S., AND SEKAR, V. Sandpaper: Mitigating performance interference in cdn edge proxies. In *Proceedings of SEC* (2019).

[19] IETF. Rfc 6928. https://tools.ietf.org/html/rfc6928.

[20] JAY, N., ROTMAN, N. H., GODFREY, P., SCHAPIRA, M., AND TAMAR, A. Internet congestion control via deep reinforcement learning. *arXiv preprint arXiv:1810.03259* (2018).

[21] JIA, Y., TOPLE, S., MOATAZ, T., GONG, D., SAXENA, P., AND LIANG, Z. Robust synchronous P2P primitives using SGX enclaves. *IACR Cryptology ePrint Archive 2017* (2017), 180.

[22] JIANG, J., SEKAR, V., MILNER, H., SHEPHERD, D., STOICA, I., AND ZHANG, H. Cfa: A practical prediction system for video qoe optimization. In *NSDI* (2016), pp. 137–150.

[23] JIANG, J., SEKAR, V., STOICA, I., AND ZHANG, H. Unleashing the potential of data-driven networking. In *Communication Systems and Networks* (Cham, 2017), N. Sastry and S. Chakraborty, Eds., Springer International Publishing, pp. 110–126.

[24] JIANG, J., SUN, S., SEKAR, V., AND ZHANG, H. Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)* (Boston, MA, 2017), USENIX Association, pp. 393–406.

[25] KNIGHT, V., AND CAMPBELL, J. Nashpy: A python library for the computation of nash equilibria. *Journal of Open Source Software 3*, 30 (2018), 904.

[26] LEGOUT, A., AND BIERSACK, E. W. Beyond TCP-friendliness: a new paradigm for end-to-end congestion control. Tech. Rep. EURECOM+390, Eurecom, 11 1999.

[27] LETHAM, B., KARRER, B., OTTONI, G., AND BAKSHY, E. *Efficient tuning of online systems using Bayesian optimization*. https://research.fb.com/efficient-tuning-of-online-systems-using-bayesian-optimization/.

[28] LETTNER, S., AND BLENK, A. Adversarial network algorithm benchmarking. In *Proceedings of the 15th International Conference on Emerging Networking EXperiments and Technologies* (New York, NY, USA, 2019), CoNEXT '19, Association for Computing Machinery, p. 31–33.

[29] LI, W., ZHOU, F., CHOWDHURY, K. R., AND MELEIS, W. M. Qtcp: Adaptive congestion control with reinforcement learning. *IEEE Transactions on Network Science and Engineering* (2018).

[30] LI, Y., LEITH, D., AND SHORTEN, R. N. Experimental evaluation of tcp protocols for high-speed networks. *IEEE/ACM Transactions on Networking 15*, 5 (2007), 1109–1122.

[31] LIN, Z., MOON, S.-J., ZARATE, C. M., MULAGALAPALLI, R., KULANDAIVEL, S., FANTI, G., AND SEKAR, V. Towards oblivious network analysis using generative adversarial networks. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2019), HotNets '19, Association for Computing Machinery, p. 43–51.

[32] MAO, H., NETRAVALI, R., AND ALIZADEH, M. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (2017), SIGCOMM'17, ACM, pp. 197–210.

[33] MEIER, R., HOLTERBACH, T., KECK, S., STÄHLI, M., LENDERS, V., SINGLA, A., AND VANBEVER, L. (self) driving under the influence: Intoxicating adversarial network inputs. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2019), HotNets '19, Association for Computing Machinery, p. 34–42.

[34] MENG, Z., WANG, M., BAI, J., XU, M., MAO, H., AND HU, H. Interpreting deep learning-based networking systems. In *Proc. ACM SIGCOMM* (2020).

[35] MUKERJEE, M. K., BOZKURT, I. N., MAGGS, B., SESHAN, S., AND ZHANG, H. The impact of brokers on the future of content delivery. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2016), HotNets '16, Association for Computing Machinery, p. 127–133.

[36] MUKERJEE, M. K., NAYLOR, D., JIANG, J., HAN, D., SESHAN, S., AND ZHANG, H. Practical, real-time centralized control for cdn-based live video delivery. *ACM SIGCOMM Computer Communication Review 45*, 4 (2015), 311–324.

[37] NASEER, U., AND BENSON, T. Inspectorgadget: Inferring network protocol configuration for web services. pp. 1624–1629.

[38] NASEER, U., AND BENSON, T. A. Configanator: A data-driven approach to improving CDN performance. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)* (Renton, WA, Apr. 2022), USENIX Association, pp. 1135–1158.

[39] NIE, X., ZHAO, Y., CHEN, G., SUI, K., CHEN, Y., PEI, D., ZHANG, M., AND ZHANG, J. Tcp wise: One initial congestion window is not enough. In *Performance Computing and Communications Conference (IPCCC), 2017 IEEE 36th International* (2017), IEEE, pp. 1–8.

[40] NIE, X., ZHAO, Y., LI, Z., CHEN, G., SUI, K., ZHANG, J., YE, Z., AND PEI, D. Dynamic tcp initial windows and congestion control schemes through reinforcement learning. *IEEE Journal on Selected Areas in Communications 37*, 6 (June 2019), 1231–1247.

[41] RAMOS, G. D. O., DA SILVA, B. C., AND BAZZAN, A. L. Learning to minimise regret in route choice. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems* (Richland, SC, 2017), AAMAS '17, International Foundation for Autonomous Agents and Multiagent Systems, p. 846–855.

[42] ROTMAN, N. H., SCHAPIRA, M., AND TAMAR, A. Online safety assurance for learning-augmented systems. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2020), HotNets '20, Association for Computing Machinery, p. 88–95.

[43] RÜTH, J., KUNZE, I., AND HOHLFELD, O. An empirical view on content provider fairness. *arXiv preprint arXiv:1905.07152* (2019).

[44] SCHAPIRA, M., ZHU, Y., AND REXFORD, J. Putting bgp on the right path: A case for next-hop routing. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks* (New York, NY, USA, 2010), Hotnets-IX, Association for Computing Machinery.

[45] SHEN, S., TOPLE, S., AND SAXENA, P. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32Nd Annual Conference on Computer Security Applications* (New York, NY, USA, 2016), ACSAC '16, ACM, pp. 508–519.

[46] SHUFF, P. Building a billion user load balancer. USENIX Association.

[47] SIDDIQI, A., AND NANDY, B. Improving network convergence time and network stability of an ospf-routed ip network. In *Proceedings of the 4th IFIP-TC6 International Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems* (Berlin, Heidelberg, 2005), NETWORKING'05, Springer-Verlag, p. 469–485.

[48] SUN, Y., YIN, X., JIANG, J., SEKAR, V., LIN, F., WANG, N., LIU, T., AND SINOPOLI, B. Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), ACM, pp. 272–285.

[49] VERMOREL, J., AND MOHRI, M. Multi-armed bandit algorithms and

empirical evaluation. In *European conference on machine learning* (2005), Springer, pp. 437–448.

[50] WANG, G., WANG, B., WANG, T., NIKA, A., ZHENG, H., AND ZHAO, B. Y. Defending against sybil devices in crowdsourced mapping services. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2016), MobiSys '16, ACM, pp. 179–191.

[51] WANG, X. S., BALASUBRAMANIAN, A., KRISHNAMURTHY, A., AND WETHERALL, D. How speedy is spdy? In *NSDI* (2014), pp. 387–399.

[52] WARE, R., MUKERJEE, M. K., SESHAN, S., AND SHERRY, J. Beyond jain's fairness index: Setting the bar for the deployment of congestion control algorithms. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2019), HotNets '19, Association for Computing Machinery, p. 17–24.

[53] YAN, F. Y., AYERS, H., ZHU, C., FOULADI, S., HONG, J., ZHANG, K., LEVIS, P., AND WINSTEIN, K. Learning in situ: a randomized experiment in video streaming. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (Santa Clara, CA, Feb. 2020), USENIX Association, pp. 495–511.

[54] YAP, K.-K., MOTIWALA, M., RAHE, J., PADGETT, S., HOLLIMAN, M., BALDUS, G., HINES, M., KIM, T., NARAYANAN, A., JAIN, A., ET AL. Taking the edge off with espresso: Scale, reliability and programmability for global internet peering. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (2017), pp. 432–445.

[55] ZERWAS, J., KALMBACH, P., HENKEL, L., RÉTVÁRI, G., KELLERER, W., BLENK, A., AND SCHMID, S. Netboa: Self-driving network benchmarking. In *Proceedings of the 2019 Workshop on Network Meets AI & ML* (New York, NY, USA, 2019), NetAI'19, Association for Computing Machinery, p. 8–14.

[56] ZHANG, X., SEN, S., KURNIAWAN, D., GUNAWI, H., AND JIANG, J. E2e: Embracing user heterogeneity to improve quality of experience on the web. In *Proceedings of the ACM Special Interest Group on Data Communication* (New York, NY, USA, 2019), SIGCOMM '19, Association for Computing Machinery, p. 289–302.