Perceiving Internet Anomalies via CDN Replica Shifts

Yihao Jia
Tsinghua University
jiayh14@mails.tsinghua.edu.cn

Aleksandar Kuzmanovic
Northwestern University
akuzma@northwestern.edu

***Abstract*—Anomalies are a ubiquitous and inevitable phenomenon associated with a complex and large-scale system such as the Internet. While measuring and analyzing network anomalies is as old as the Internet itself, comprehensively detecting anomalies at a global scale is a challenging task that requires a significant measurement infrastructure. In this paper, we demonstrate that the production Content Distribution Networks (CDNs), and their pervasive network infrastructure, could be effectively utilized to detect Internet anomalies. Our approach avoids direct network measurements and instead relies on "abnormal" spatial and temporal CDN replica shifts to indirectly sense anomalies. We measure replica shifts for five CDNs (Google, Amazon, Akamai, Fastly, and Incapsula) for two months. Contrary to our expectations, we find that (i) Google's and Amazon's CDNs, which are characterized by rich connectivity and infrastructure, are *not* best suited for our method because they effectively mask anomalies; (ii) Akamai is the most "sophisticated" of all evaluated CDNs, yet again not best suited to detect anomalies because it reacts exceptionally to much smaller network performance variations; (iii) Fastly's and Incapsula's replica shifts strongly correlate with network anomalies, making them viable anomaly predictors.**

***Index Terms*—anomalies prediction, CDN, DNS mapping**

I. Introduction

By constantly and increasingly providing information and services to the users, the Internet has already evolved into one of the most crucial infrastructures in the world. It is no exaggeration to say that the entire world's economy nowadays critically depends on the Internet and its reliability. However, Internet anomalies are technically inevitable, and even a tiny connection degradation may induce poor user experience, and thus a significant revenue reduction [1], [2]. To deal with anomalies, network operators and administrators consider malfunction detection and recovery as one of their vital tasks. While detecting anomalies within a single network is relatively feasible, timely reporting could be extremely costly involving abundant continuous measurement [3], [4]. Moreover, measuring and analyzing the Internet anomalies at a global scale requires not only continuous measurements, but also a significant infrastructure, which necessarily increases the operational costs [5].

To improve the quality of service and reliability, Content Distribution Networks (CDNs) distribute online content closer to end-users by deploying hundreds of thousands of servers worldwide [6]. Additionally, to minimize the client-server latency, such systems perform extensive network and server measurements and use them to redirect clients to different servers. Since the "best" replica match is constantly changing due to the highly dynamic network conditions, *including* network anomalies, the question is if and how we can infer network anomalies via CDN replica shifts?

In this paper, we aim to *detect network anomalies by monitoring CDN replica shifts.* By "detecting anomalies via CDN replica shifts," we mean to detect anomaly events in *statistical terms*, not in a deterministic sense. Indeed, our methodology provides "anomaly alerts" that are likely to correlate with real network anomalies, as we demonstrate via a large-scale measurement study. As such, a system utilizing our methodology is envisioned as a reliable "whistle-blower" to trigger other, more sophisticated, systems to direct their measurements to locations flagged by us. Our method thus has a potential to significantly reduce the measurement overhead and operational costs of such systems.

The key challenge with our approach lies in the fact that CDN replica shifts are frequent events, often happening at time scales of seconds [7]. Replica shifts can occur for various reasons, including CDN load balancing policies, internal maintenance issues, network-performance optimization (*e.g.*, latency towards one replica smaller by 1 *ms* than the latency to another replica), and certainly due to Internet anomalies. However, anomalies typically occur over longer time scales [8], *i.e.*, far more rarely. The question is thus which CDN shifts are triggered by network anomalies?

Our key hypothesis is simple and intuitive: given that anomalies are exceptional network events (manifested by communication disruption, link failures, high packet loss rates or significantly increased network latency), we suppose that such events trigger exceptional CDN reactions. For example, redirecting users to a spatially distant CDN replica implies exceptional CDN behavior, given that its main purpose is to localize Internet traffic. We call such events *regional shifts*. Likewise, redirecting users to an "alien" replica, still within the same region yet not previously associated with the given users, is another potential indicator of anomalies. Given that such shifts are often short-lived, we call them *occasional shifts*.

We conduct a large-scale measurement study by continually querying five CDNs — Akamai, Google, Amazon, Fastly, and Incapsula — for two months. In addition, in order to establish a "ground truth" for anomalies, we utilize the RIPE Atlas infrastructure to conduct triggered measurements and detect latency inflation for select networks that our CDN-based system flags as anomaly-prone.

We find that Google frequently shift replicas, far more than the remaining three CDNs. Still, such shifts are apparently heavily induced by its internal load balancing mechanisms – not the Internet anomalies. As an example, we didn't detect a single regional shift for Google and for Amazon over a two-month measurement. Next, we show that Akamai is apparently the most "sophisticated" of all evaluated CDNs as it considers the Internet traffic and network properties more than other CDNs. However, this further means that Akamai is not well suited to predict anomalies, given that the vast majority of shifts are performance-, yet not necessarily anomaly-related. We find that while Fastly and Incapsula CDNs experience the smallest number of replica shifts of all CDNs, regional shifts in their cases strongly correlate with network anomalies, making them viable anomaly predictors. In all cases, regional shifts are far more reliable indicators of anomalies than occasional shifts. Finally, concurrent regional shifts for multiple CDNs are the strongest indicator of anomalies.

Contributions. We list the key contributions below.

1) (First to) propose a methodology to perceive anomalies via CDN replica shifts.
2) Explore the factors and ingredients that contribute to the shifts and anomalies detection.
3) Study the accuracy and promptness of the most popular CDNs in anomaly resilience.
4) Validate the feasibility of the proposed methodology by predicting anomalies in an authentic network environment.

Roadmap. The rest of this paper is organized as follows. Section II presents the motivation and necessary background of CDN replica shifts, and proposes a CDN-based anomaly detection methodology. Then, in Section III, we elaborate on the characteristics of shift behaviours for five distinguished CDNs, and detail the experiments in analyzing real Internet anomalies. In Section IV, we discuss possible system optimizations. Finally, we present related work in Section V, and conclude in Section VI.

II. Motivation and Principle

Here, we first provide the necessary background on CDNs and their mechanisms. Then, we outline the problem and provide the rational and approach to addressing the problem.

A. Background

CDNs attempt to improve web and streaming performance by delivering content to end users from multiple, geographically distributed servers typically located at the edge of the network [9]. CDN providers usually build far-ranging points-of-presences/replicas around the world. When a user is requesting content that is hosted on a CDN, ideally, the CDN aims to direct the user to a replica that provides the best performance. Indeed, it has been demonstrated that in most cases the user will be mapped to a proximate replica [10]. In addition to network proximity, CDNs often consider other features (*e.g.*, replica server load) when deciding where to direct a user, as we explain in more detail below. Given that both the network conditions and server load dynamically change over time, this leads to *replica shifts*.

Currently, CDNs deploy at least one of the following two methods for directing users to the nearby replicas.

Anycast-based CDNs [11] use a globally-unique address — anycast address — to serve the content. As a result, the user direction is naturally handled by the routing protocol, *i.e.*, Border Gateway Protocol (BGP), without a separate mapping system involved. Although anycast significantly reduces the difficulty in establishing a CDN, the downside is that CDNs providers largely lose direct control over user the direction, and outsource it to BGP. From our perspective, anycast-based CDNs aren't well suited for helping detect anomalies for several reasons. BGP alone is often a source of Internet anomalies. Even severe anomalies might not lead to replica shifts in anycast [12], as the underlying BGP protocol masks such effects. We thus refrain from utilizing CDNs that are anycast based only, *e.g.*, Cloudflare, in our work.

Domain Name System (DNS)-based CDNs [13] use DNS to direct users. This approach gives much more control to the CDN providers. In particular, by setting a short DNS time-to-live parameter, they are capable of redirecting users to different replicas over short time scales, *e.g.*, seconds [7]. This, however, requires the CDN to utilize a large-scale, often proprietary, mapping system. Such a system necessarily involves continuous network and server measurements. From our perspective, given that DNS-based CDNs often react quickly to changes in network performance, and given that such changes incur replica shifts, makes them ideal for our method. We provide an example scenario below.

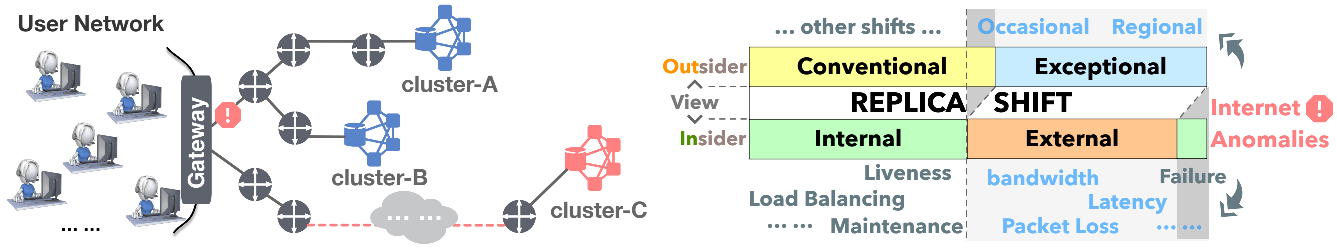
B. Example Scenario

Consider a scenario shown in Figure 1a. For users that reside in a given network, in absence of any anomalies, a CDN will direct users to replicas in clusters A or B. At some point, consider an anomaly on a gateway link in the network, as shown in the figure. Assuming continuous CDN measurements, it is expected that a CDN detects this event, and quickly redirects users to a different replica. In the example shown in the figure, it is a replica in cluster C. While a replica in cluster C may be a distant and suboptimal choice, it still avoids the anomaly and provides better performance. Later in the paper, we demonstrate that events as the one shown here are frequent, and that our methodology can detect them.

While CDN redirections are a common phenomenon, network anomalies (typically) are not. The question is thus how do we single out CDN replica shifts that likely *do* correspond to anomalies. To answer this question, we first provide a "deeper dive" into CDN replica shifts and different reasons that cause it, and then outline our approach to address the problem.

C. Rationale and Approach

Here, we outline the reasons that cause CDNs to shift replicas. First, we provide an insider view, *i.e.*, from the CDN



(a) Replica shift reflects Internet anomalies

(b) Rationale of replica shift

Fig. 1. Anomalies behind the replica shift: (a) replicas in cluster-C are relatively remote and suboptimal (compared to replicas in cluster-A/B) for serving users in this network; when an anomaly happens, users will be served from replicas in cluster-C, instead of the regular ones in cluster-A/B, to maintain a desirable QoS. (b) CDN shifts replicas for either internal or external reasons; shifts for external reasons include Internet anomalies; anomaly-induced shifts usually behave exceptionally in spatial and temporal domains, and thus indicate the anomalies.

perspective. Then, we provide an outsider view, *i.e.*, how do CDN replica shifts look like to an external observer.

1) An insider view: According to [14], numerous factors are taken into account by a CDN, *i.e.*, Akamai in this particular case. Denote by *Domain* the domain name requested by a user, by $IP_{\text{requester}}$ the IP address which determines the network location of the requester, by $\sum_{Internet}$ the external (outside the CDN) network conditions, and by $\sum_{CDN}$ the internal (within the CDN) network or server status. Then, the mapping-system logic can be abstracted as follows.

$$IP_{\text{replica}} \Leftarrow Map(Domain * IP_{\text{requester}} * \sum\nolimits_{Internet} * \sum\nolimits_{CDN})$$

Thus, for a particular domain name and a particular user, either the internal network or server status, or the external network status will cause replica shifts.

Figure 1b summarizes typical reasons for replica shifts caused by internal and external status. In particular, load balancing, server availability, and maintenance are reasons that can affect the internal status, and cause replica shifts. On the other hand, variation in Internet latency, packet loss, and reduced bandwidth can affect the external status, and cause replica shifts. Necessarily, network anomalies can certainly cause replica shifts, given that they are often characterized by significant latency or packet loss inflation, that leads to bandwidth degradations or outages. The question remains — how do we pinpoint replica shifts that likely correspond to network anomalies?

2) An outsider view: Our key hypothesis is the following: given that anomalies are exceptional network events, we suppose that such events trigger *exceptional* CDN redirections, as we describe below. For example, as shown in Figure 1a, replica shifts inside cluster A, or shifts from cluster A to cluster B, could be regarded as a conventional shift, while a shift to a remote cluster C might be more likely considered an exceptional shift.

Spatially exceptional shifts. We call a CDN replica shift a *regional shift* if the users are mapped to a different region than the one they reside in. By different region, we mean a different *continent*, as we explain in more detail later in the text. Indeed, CDNs are designed to localize the Internet traffic and provide high performance to users. Directing a user to another continent is an event that *might* be caused by a network anomaly on a route between a user and a previous replica.

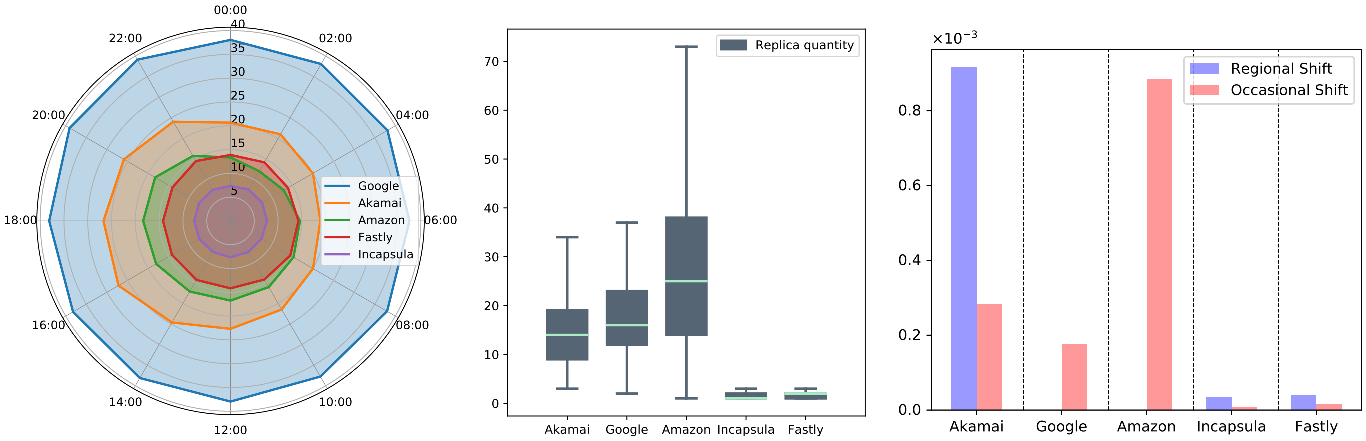
Temporally exceptional shifts. We call a CDN replica shift an *occasional shift* if the users are mapped to an "alien" replica that rarely, if ever, showed up for the particular users. Such an event also might be caused by a network anomaly. Given that such shifts are often short-lived, we call them occasional shifts.

We emphasize that the above heuristics are necessarily error-prone and statistical in nature. Indeed, we use them only as good *hints* for flagging potential network anomalies. Numerous other events could lead to exceptional CDN shifts. For example, CDN-level failures, maintenance, *etc*. Still, we demonstrate below, via a large-scale measurement study, that exceptional shifts for select CDNs are indeed largely correlated with Internet anomalies.

III. In What We Trust?

Here, we need to decide which CDNs are we going to select for our analysis. Our first criteria is that a CDN is DNS-based because they are expected to be compatible with our methodology. Given that most of the commercial CDNs are DNS-based, this do not significantly constrain our selection. Our second desired criteria is that a CDN supports the EDNS0 client subnet option (ECS) [15]. Since ECS provides adopters accurate user locations, CDNs that enable ECS in practice are more likely to concern QoS and mapping accuracy [16], and thus, are more likely to conduct constant and sophisticated network measuring. We provide some brief context below.

Historically, one of the key reasons for systematic CDN imperfections was the distance between clients and their local DNS resolvers [17]. This issue was further dramatically amplified in recent years with the proliferation of public DNS resolvers [18], *e.g.*, [19], [20]. In an attempt to remedy poor server selection resulting from local DNS resolvers, there has been a recent push, spearheaded by public DNS providers, to adopt the ECS [21]. With ECS, the clients IP address (truncated to a /24 or /20 subnet for privacy) is passed through the recursive steps of DNS resolution as opposed to passing the local DNS resolvers address. Thus, by putting any desired IP as $IP_{\text{requester}}$, this option itself provides another significant



(a) Replica shift frequency.

(b) Regular replica quantity across networks.

(c) Frequency of occurrence: exceptional shifts.

Fig. 2. Exploring (exceptional) replica shifts: (a) Frequency: Google>Akamai>Amazon>Fastly>Incapsula; Akamai and Amazon shift more frequently during daytime than overnight. (b) Google, Akamai, Amazon show an affluent quantity in assigning replicas to networks, far beyond that in Incapsula and Fastly. (c) Both regional shifts and occasional shifts are rare among all CDNs; Google and Amazon show no regional shifts; Incapsula and Fastly rarely show regional and occasional shifts relative to others.

feature for us — The behaviors of CDNs can be *remotely* monitored from a *single* machine (*e.g.*, see [22]).

Based on the above criteria, we selected Amazon Cloudfront, Google Cloud, Fastly, and Incapsula. While the largest CDN, Akamai, supports ECS, it does not accept a user-defined ECS option [23], *i.e.*, an arbitrary IP is replaced by the actual requester IP. To still include Akamai in our study, we proceed as follows. We select 1,000 open and stable DNS resolvers in the US, and utilize them in our measurement study.[1] While this limits our measurements to US, we still find it valuable to include Akamai in our study.

One final question is which domains to select for each of the CDNs. We opt for popular domains that constantly serve abundant traffic, such that even a slightly inappropriate mapping may cause significant problems and performance degradation for users. Based on this, we select the following domains: *apple.com* (Akamai), *android.com* (Google), *zillow.com* (Amazon), *t-mobile.com* (Incapsula), and *imgur.com* (Fastly).

A. The Shift Primer

CDNs shift their replicas continuously for a user. Here, we aim to understand how frequently these CDNs are likely to shift their replicas over time. To that end, we query the 1,000 recursive DNS resolvers once every 20 seconds for the domains outlined above, and record the responses. Besides, we consider a shift to be a change of a replica to another replica beyond the $/24$ address space. In particular, a change from $5.5.5.5$ to $5.5.5.10$ is not regarded as a replica shift, yet a change from $5.5.5.5$ to $5.5.10.5$ is. Since $/24$ is the longest prefix that can be routed by BGP, shifts within the same $/24$ reside in the same cluster, and share the same routes. This, to a large extent, helps us to filtering the shifts residing in the same cluster, which is probably not triggered by anomalies.

Figure 2a depicts the number of CDN replica shifts (on the radius) for every 2 hours of the day. For example, if the replica shift happened every time we measured it, the radius would have 2 (hours) * 60 (minutes) * 3 (20 seconds) = 360 samples. The result is averaged over all $1,000$ resolvers and the days across the entire June of 2018.

Figure 2a shows that Google, of all the evaluated CDNs, deploys most replica shifts. On average, one replica shift every 200 seconds. Still, the figure also shows that Google has almost identical and deterministic number of replica shifts over every two hours of a day. While it is certainly expected that Google's CDN optimizes performance for its users, the time-independent profile (same during day and night) of the number of replica shifts implies that apparently a deterministic internal mechanism leads to an almost identical number of shifts over time.

On the contrary, an ellipsoid-like curves for Akamai and Amazon show that replica shifts in their cases are more likely to happen during daytime than overnight.[2] This is despite the fact that we average over 1,000 networks over a month. Given that the users are more active during daytime, we hypothesize that this dominantly drives the observed behavior. Finally, Fastly and Incapsula have a smaller number of replica shifts relative to other CDNs. This implies that their behavior is fairly stable. We will show that this is essential for our method.

B. The Frequency of Exceptional Shifts

Here, we explore how frequently do the exceptional — regional and occasional — shifts occur. Initially, we first re-introduce these types of replica shifts.

Regional shifts: Given that our resolvers are in US, we define the regional shifts as follows. For a specific network/resolver, if the returned replica(s) is located outside US,

[1] We verify that the DNS resolvers we use do *not* support ECS. Hence, the DNS requests are resolved based on the IP addresses of 1,000 geographically distributed DNS resolvers.

[2] There are four time-zones in US spanning three hours. Figure 2a corresponds to the US Central Time Zone.

Canada, or Mexico[3], we denote it as a regional shift. For this purpose, we utilize geolocation databases and additional active, yet simple, measurements that confirm that a replica is outside these three countries. While geolocation databases are known to often have poor quality [24], [25], verifying that an IP is on the other continent is rather straight-forward. We recognize that our definition of a regional shift is rather simple. Still, it serves our purpose well here.

Occasional shifts: To establish a baseline for replicas that are regularly shown to a network/resolver, we proceed as follows. For a specific network/resolver, we collect all replicas that were showed up in June as "regular replicas" with the following exception: we exclude replicas that showed up less than 0.1% of cases. We remove such replicas because we want to filter the replicas that may have been assigned temporarily, as an example, to avoid anomalies. Then, for every replica that showed up in July and is *outside the base* established in June, we denote it as an occasional shift. We are also conservative when judging whether a replica is outside the base: if the IP address of this replica does not overlap with any of the addresses from the base at the $/24$ level, we denote it as an occasional shift.

Figure 2b shows the distribution of the size of conventional replica groups (established in June), in the context of occasional shifts. The figure shows that Akamai, Google, and Amazon expectedly feature relatively large groups, given that these are the world's largest CDNs. In particular, Akamai and Google usually assign a client a set of 10-20 replicas, five at least. This appears rational and certainly improves the reliability of content delivery in these networks. Amazon is providing even a larger number of replicas. On the contrary, Incapsula and Fastly usually assign only a small set of replicas for a specific resolver, at *most* five. This is again expected, as their infrastructure is much smaller relative to the above CDNs. Despite the fact the number of replica groups is smaller for Incapsula and Fastly, we will demonstrate later that this is actually *good* for our methodology, which relies on exceptional shifts.

Going back to the main topic — the frequency of exceptional shifts — we next provide the measurement results for such shifts. We captured the exceptional shifts within a 1-week period from July 15th to 22nd.

Figure 2c shows the results, *i.e.*, depicts the frequency of occurrence of exceptional replica shifts. The first insight is that regional and occasional shifts are rather rare events. Indeed, for most CDNs, less than one *permille* of all shifts are exceptional shifts. For regional shifts, Akamai leads the pack. We will show later that not all such shifts are associated with anomalies, and hypothesize that Akamai regularly applies such shifts for internal reasons, unknown to us. Amazon, on the other hand, leads the pack in terms of occasional shifts, given that it is fairly active in assigning new replica to users, beyond stable replica groups. Finally, Incapsula and Fastly have the smallest number of shifts. Nonetheless, we will demonstrate that such shifts are the most useful ones from our perspective. The underlying logic is simple: Incapsula and Fastly rarely utilize exceptional shifts, yet when they do, they do for a good reason — a fairly severe Internet anomaly.

C. Active Measurements Triggered by Exceptional Shifts

Our system is envisioned to be used as a trigger for other, more advanced, monitoring systems to explore the root causes behind potential anomalies. Still, we need some "ground truth" to understand when the actual anomalies are happening in a network, so that we can analyze the performance of our system. To address this problem, we conduct a simple active measurement methodology to help us reach our desired goal. In short, whenever a regional or occasional shift is triggered by a CDN, we probe the given resolver by sending active probes from 100 geographically distributed nodes in US, and measure the round-trip time. Then, we aim to understand if the round-trip time has significantly increased on any of the 100 paths towards the resolver, indicating an anomaly. While this is an ad-hoc and simplistic method, we demonstrate that it does provide sufficiently reliable data to fortify our claims.

In particular, we utilize the RIPE Atlas platform [26] and randomly select 100 nodes located in US. A single probe from a measurement node involves sending three packets, and we select the *minimal* Round-Trip Time (RTT) (out of three) as the current RTT. This is mainly to reduce the interference of the network jitter on the measurement results. To establish a good minimum RTT results, we continuously probe the 1,000 resolvers from the fixed 100 RIPE Atlas nodes over 3 days.

Formally, denote by N the number of measuring nodes, *i.e.*, $N = 100$. For a given resolver j, we denote the RTT value from the node i to this resolver j by rtt_i^j, and denote the minimal (best) RTT value we ever seen by Rtt_i^j. Then, once there is an anomaly triggered by our system, we conduct active measurements towards that particular resolver (as explained above, each measurement consists of 3 probes and we record the minimum). Then, denote by Δ_j^{value} the "summary measurement" after an anomaly happens, emphasizing the severity of an anomaly around resolver j,

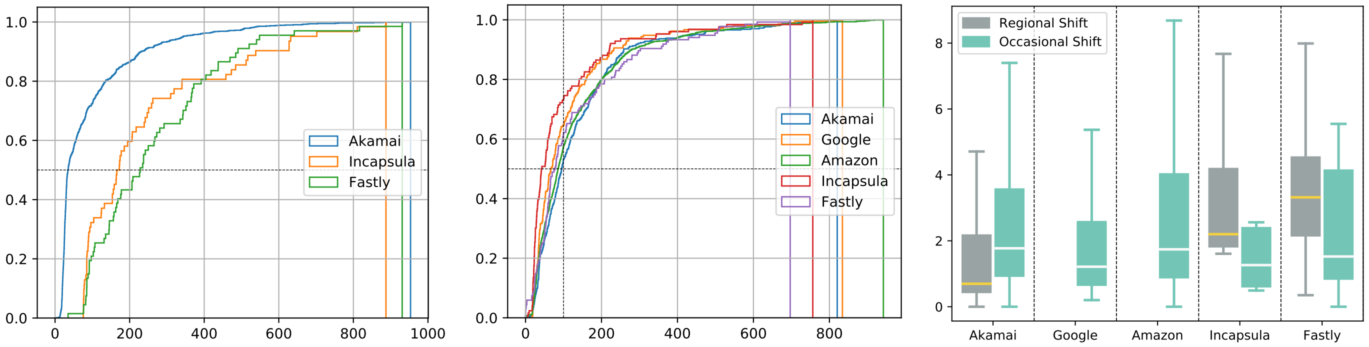
$$\Delta_j^{value} = \max_{i \in N}(rtt_i^j - Rtt_i^j).$$

Effectively, we collect all N measurements towards a resolver, and select the one that shows the largest increase relative to its minimum value, Rtt_i^j. Almost exclusively, the value will be positive because it is highly unlikely that all of the 100 probes will experience the minimal RTT. If the largest RTT increase on 100 paths isn't significant, then it is unlikely that any anomalies are present around the given resolver. Likewise, if the latency does increase significantly, this implies potential anomalies around the resolver.[4]

Next, define Δ_j^{ratio} as follows,

[3] None of the replicas in our data set was located in Mexico.

[4] Severe packet losses are another strong signature for anomalies. We, however, do not utilize it at this point. Still, we note that almost any QoS degradation, such as bandwidth reduction or increased packet loss rates, are often accompanied by inflated latency, which is what we measure.



(a) *cdf* of regional shifts on Δ_j^{value}

(b) *cdf* of occasional shifts on Δ_j^{value}

(c) Overall distributions on Δ_j^{ratio}

Fig. 3. Insights behind exceptional shifts: (a) Akamai shows $35ms$ latency inflation in the median case – a weak indication for anomalies; Incapsula and Fastly present around $200ms$ in the median case – a clear indication for anomalies. (b) All CDNs present 40-100ms in the median case – a relatively weak indication compared to 3a. (c) Akamai has Δ_j^{ratio} less than 1 for regional shifts in the median case – a weak indication for anomalies. This result corresponds to insights in 3a; Incapsula and Fastly feature Δ_j^{ratio} of 2 and 3 for regional shifts in the median case – a strong indication for anomalies. This result corresponds to insights in 3a; Incapsula and Fastly have Δ_j^{value} less than 2 for occasional shifts in the median case – weaker than for their regional shifts.

$$\Delta_j^{ratio} = \max_{i \in N}((rtt_i^j - Rtt_i^j)/Rtt_i^j).$$

Δ_j^{ratio} measures the severity of an anomaly in the vicinity of a resolver j in *relative* terms. In particular, it maximizes the *ratio* of the RTT inflation relative to the minimal RTT value among all 100 probes. For example, this measure may help us understand the potential severity of an anomaly when the rtt_i^j reaches $40ms$ from it's best $Rtt_j^{value} = 10ms$, given that $30ms$ does not appear significant, while Δ_j^{ratio} is already 3. Below, we utilize Δ_j^{value} and Δ_j^{ratio} to quantify the severity of potential anomalies in the vicinity of the resolver networks.

D. Insights Behind Exceptional Shifts

Here, we would like to have a deep understanding of exceptional shifts above, and uncover (i) how likely is it for an anomaly to arise when a CDN shows an exceptional shift, and (ii) which CDNs present a better indicator for a real anomaly.

1) Regional shifts: Figure 3a depicts the Cumulative Distribution Function (CDF) of Δ_j^{value} for regional shifts, where j denotes the particular resolver that detected the particular regional shift. The figure aggregates data from all resolvers. Necessarily, the figure shows results for only three CDNs (Akamai, Incapsula, and Fastly), given that that Amazon and Google show no regional shifts, as we established in Figure 2c. As a first insight, we see that the median value of Δ_j^{value} for Akamai is as low as $35ms$. While the tail of the distribution grows to as high as above $900ms$, the relatively low median value implies that Akamai deploys regional shifts even in absence of anomalies in the vicinity of the resolvers. We assume that this happens due to load-balancing or other internal, non-Internet performance related, reasons.

On the other hand, *the key insight* from Figure 3a is that Incapsula's and Fastly's regional shifts are indeed strongly correlated with high Δ_j^{value} values. Indeed, in the median case, Δ_j^{value} is as high as approximately $200ms$, which is rather significant. This implies that while the Incapsula's and Fastly's regional shifts are rare (Figure 2c), they are highly likely to reflect real Internet anomalies, as shown in Figure 3a.

2) Occasional shifts: Figure 3b shows the results for occasional shifts. In summary, we see that occasional shifts, which are less exceptional events in nature, correspond to less exceptional anomalies. For example, in the median case, occasional shifts correspond to the increase in RTT between 40 and $100ms$. While this is not insignificant, this is still less severe than for regional shifts. Finally, we do see that for about 20% of occasional shifts, the latency inflation is above $200ms$. Thus, we conclude that occasional shifts are reasonable indicators of anomalies. However, they are blurred by conventional shifts. We discuss this in more detail in Section IV.

3) Relative Latency Inflation Ratios: Here, we evaluate the effect of the relative latency inflation ratio, Δ_j^{ratio}, which captures the maximum *relative* RTT inflation for a particular resolver j.

Figure 3c shows the results for the five CDNs, for regional and occasional shifts. First we note that the relative inflation ratios are particularly pronounced for Incapsula and Fastly for regional shifts. Indeed, the inflation ratio increases to between 2 and 3 in the median case, and up to around 8 times in edge scenarios. This confirms our above insights that regional shifts for these two CDNs are reliable indicators of anomalies. Second, the figure shows that Akamai's inflation ratio for regional shifts is rather small, *below* 1 in the median case. This confirms that indeed Akamai deploys regional shifts in absence of any anomalies. Finally, with respect to occasional shifts, we can see that they are much weaker for Incapsula and Fastly relative to regional shifts. At the same time, we note that *variance* of inflation ratio is rather high for Akamai, Google, and Amazon. This implies that substantial portions of the shifts correspond to anomalies, while the remaining shifts

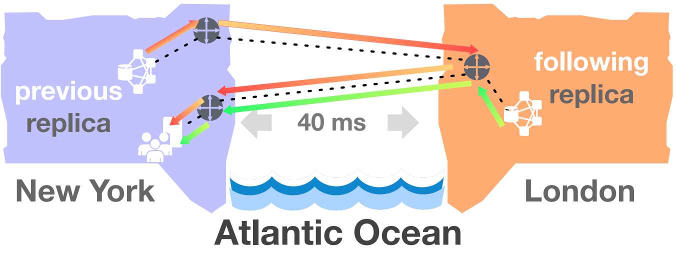


Fig. 4. Case illustration: The traffic detours to London before it arrives New York. Thus, traffic will delay a round-trip journey (40ms*2) before arrive to New York users from New York replica(s). While it take only one-way journey (40ms*1) to New York users if contents can be served from London.

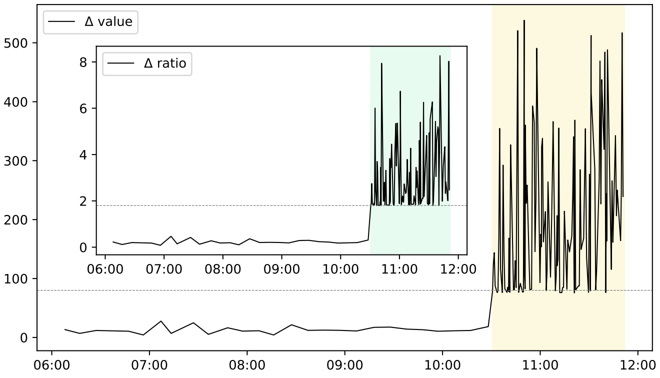


Fig. 5. Anomaly indication: latency inflation around affected resolvers. Δ_j^{ratio} of this affected resolver increase to at least 1.8, and Δ_j^{value} increase to at least $80ms$ (j points to resolver 69.9.160.191). The $80ms$ Δ_j^{value} corresponds to the round-trip delay ($40ms * 2$) that depicted in Figure. 4, showing a decent indication of such anomaly case.

do not. Hence, more sophisticated methods (than our simple ones) are needed to identify anomalies within occasional shifts, as we discuss in Section IV.

E. Multi-CDN Anomaly Detection: A Case Study

Here, we evaluate the scenarios when regional shifts are *concurrently* triggered by multiple CDNs for the same resolver. Given that a regional shift for a single CDN is already a strong indicator of an anomaly, having multiple CDNs experience regional shifts at the same time should be a very strong signature of an anomaly. Necessarily, whenever a regional shift happens, we trigger the 100 monitors to measure the absolute and relative latency inflation. In addition, here we analyze the entire scene in an attempt to uncover the underlying root causes of the anomaly. We discovered several such incidents, flagged by concurrent regional shifts at 3 CDNs — Akamai, Incapsula and Fastly. One incident particularly stands out, both in terms of frequency and longevity. We evaluate it in detail below.

1) Scene Restoration: The anomaly is a detour of traffic from New York to London, and then back to New York. This case is first reported by our system at UTC-time *10:30:08* July 27th. An important insight is that all three CDNs, Akamai, Incapsula, and Fastly alarmed it simultaneously in a "second" granularity. These alarms all affect resolvers 69.9.160.191 and 69.9.191.4. We check these 2 addresses and find that both of them belong to the IP block 69.9.160.0/19, which is held by Autonomous System Number (ASN) 29791 and *Voxel Dot Net, Inc*. We confirm that these 2 IPs are located around New York, U.S. We further inspect the routes generated by the 100 RIPE Atlas probes, and we confirm that 98 out of the 100 probes, generated from U.S. to these 2 resolvers, will first be detoured to the IP 64.95.159.38, which is located in London and belong to ASN 29791 and *Voxel Dot Net, Inc* as well.

Figure 4 depicts the scene when the anomaly happened. Before the anomaly arose, Akamai, Incapsula, and Fastly all assign replica(s) that are located in the New York area. However, all these 3 CDNs shift their replica(s) to London simultaneously after the anomaly emerged. The one-way delay between New York and London is around $40ms$, hence the round trip time is about $80ms$. Thus, contents (of *apple.com* (Akamai), *t-mobile.com* (Incapsula), and *imgur.com* (Fastly)) that deliver to users behind these resolvers will be delayed for at least $80ms$ if the CDNs remained with the previous replica. Thus, redirecting traffic to London in this case is a *better* decision made by the 3 CDNs, than sticking with the previous replica. The decision is better because when the replica is in London, the latency is induced by going over the Atlantic ocean *once* (*NYC – London – NYC*). With the original replica, it would go over the Atlantic Ocean *twice*, *i.e.*, *NYC – London – NYC* to get to the replica in New York, and then *NYC – London – NYC* to retrieve content from the replica. The bottom line is that this set of events show the utility of our system to pinpoint real-world Internet anomalies.

Interestingly, we find no regional shifts by Google and Amazon in this case. We hypothesize that they utilize different, much richer, connectivity in this area, which does not affect their performance as dramatically. Indeed, our own measurements show that 2 of the 100 monitoring probes do *not* detour to London. Hence, shorter paths are indeed available. Since Amazon and Google serve users from all around the world, they are actively seeking for exclusive and direct connections to user networks to reduce latency and improve the QoS [27], [28]. By building comprehensive direct connections around the world, traffic between end-users and Google or Amazon are highly accelerated by their rich connections. This necessarily reduces the possibility of experiencing anomalies as well. A recent measurement study of Google's and Amazon's CDNs show that 66% of the Google's traffic take *only 1 AS hop* before reaching the destination user networks, while 95% of them takes less than 2 hops [29]. This number is as high as 80% for Amazon, far beyond the other cloud platforms.

2) Triggered Measurement Results: Figure 5 shows how our triggered measurement system detected this event. We retrieve the data from RIPE Atlas and analyze Δ_j^{value} and Δ_j^{ratio}. Δ_j^{ratio} and Δ_j^{value} are collected every $10min$ before the anomaly happened. At *10:30:08*, our system was alarmed by Akamai, Incapsula, Fastly, which automatically triggers the frequency of our measurements to one every $30s$. Figure 5 shows that Δ_j^{value} increases for about $80ms$ *at least* for every data point during the anomaly period, which corresponds to the trans-Atlantic RTT inflation explained above. Likewise, Δ_j^{value} climbed by at least 1.8 times as well, showing a clear indication of a real anomaly.

IV. Possible Optimizations

Our main goal was to demonstrate that it is possible to utilize CDNs, together with their significant infrastructure and Internet measurements they conduct, to help us detect Internet anomalies. While successfully achieving this major task, we do recognize that our methodology, in its current form, is fairly rudimentary. Below, we outline potential ways to improve it so that it becomes widely deployed and achieves better alarming accuracy. Necessarily, we leave all such improvements for future work.

First, our system currently involves an infrastructure of 1,000 DNS resolvers. As explained above, this is *not* a must for our system, and we used in order to evaluate Akamai's CDN. Otherwise, we can rely on ECS to remotely query DNS for a random IP. As such, our system can be made widely available to anyone who has interest in a low-cost and effective method to flag Internet anomalies. Such alarms could then trigger other, more sophisticated, systems to explore the root causes behind the anomalies.

Second, we argue that the accuracy of detecting anomalies is highly dependent on the spatial and temporal base we made. For example, re-directing users to another continent is indeed an exceptional event, but more sophisticated methods could be deployed. Next, our methodologies could be tuned specifically for individual CDNs and for different networks on a case-by-case basis. We further note that statistical analysis techniques could be applied to this problem, and we plan to do so. Nonetheless, we argue that despite the wide space we left to optimize our system, our key contribution is to show that this approach is feasible.

V. Related Work

Measuring Internet anomalies has been an active research topic for decades. Here, we provide a necessarily not comprehensive overview of related work, which provides only a small subset of representative systems. In [3], it has been shown that delay and forwarding anomalies could be pinpointed using large-scale traceroute measurements, while in [30] and [31], machine learning and sophisticated statistical techniques were shown to be effective in the network anomaly detection. For such insights, it is often necessary to use direct traffic volume measurement at Internet links [32]. Recently, it has been shown that detecting network anomalies is a relevant problem in software-defined networks [33]. Common for the above systems and methods is that large-scale monitoring requires a large-scale infrastructure. Our system avoids direct measurements by utilizing CDNs' infrastructure. As such, it could be utilized as a trigger mechanism to dramatically reduce the measurement overhead associated with the above systems.

The performance of a CDN is highly influenced by the accuracy of mapping users to replica(s) which provides the best QoS to them. Thus, CDNs actively keep updating their mapping systems. In [14], the authors disclose the Akamai's mapping architecture, which depicts a valuable content delivering framework. In [34], the authors reveal a core algorithm used for user mapping. Meanwhile, it has been confirmed in [16] that when utilizing the ECS option, user mapping for public resolvers results in an extraordinary accuracy, and helps accelerate the time-to-first-byte delivery by at least 30%. Motivated by the benefits of replica-user mapping, the authors of [12] effectively balance the load in an anycast CDN by introducing the mapping system. An increased accuracy and sophistication of a mapping system gives us increased confidence in flagging an anomaly when CDNs exceptionally shift their replicas.

Our paper is not the first to utilize CDNs to achieve goals beyond CDNs' main purpose. In particular, given the significant expansion of Google's CDN, it has been shown that it can be effectively used to map the Internet [35]. In [7], CDN measurements are used to reduce the measurement overhead in overlay routing, while in [36] CDNs are used to estimate network distance between arbitrary Internet hosts. Finally, multi-CDN systems are utilized to build an affordable DDoS defense mechanism [37]. To the best of our knowledge, we are the first to propose a method to use CDNs to detect Internet anomalies.

VI. Conclusions

In this paper, we demonstrated the feasibility of perceiving Internet anomalies via CDN replica shifts. We first introduced the principles behind replica shifts, and then presented a methodology that can be used to indirectly sense anomalies. In particular, our key idea is that given that network anomalies are exceptional events, CDNs' reaction to them should be exceptional as well. We thus introduced regional (spatially exceptional) and occasional (temporally exceptional) shifts and analyzed our hypothesis that such shifts are correlated with Internet anomalies. To that end, we measured five CDNs — Akamai, Google, Amazon, Incapsula, and Fastly — across the US over two months.

Our key insights are the following.

1) Regional shifts are more correlated with anomalies than occasional shifts are. This is not a surprise, given that regional shifts are more severe in nature, *i.e.*, route traffic to another continent, than occasional ones.
2) Akamai, Google, and Amazon are powered by an extensive infrastructure to serve the vast traffic to users with a high reliability. This is further accompanied by very close proximity to the users and fine-grained re-direction mechanisms. As such, counter-intuitively, these CDNs are *not* well suited for our anomaly detection methodology, because they are too sophisticated.
3) Unexpectedly, we find that Incapsula and Fastly are almost perfectly suited for our methodology. This is despite the fact that they provide the least number

of replica shifts (Figure 2a), have the lowest quantity of replica in their base (Figure 2b), and exhibit the smallest frequency of both regional and exceptional shifts (Figure 2c). Yet, their shifts are most strongly correlated with real network anomalies (Figure 3a). The bottom line is that Incapsula and Fastly shift replicas infrequently, but when they do, it happens for a good reason — they are likely to flag an Internet anomaly.

4) Scenarios when multiple CDNs concurrently exhibit an exceptional shift almost surely imply Internet anomalies.

We conclude by noting that our method is very simple and practical, enabling anyone to remotely flag potential network anomalies, thus dramatically reducing measurement overhead and operational costs of existing monitoring systems.

Acknowledgment

We sincerely thank Alexander Azimov, Marc Anthony Warrior, Baojun Liu and Yaji Xing for their valuable discussion and feedback. We also thank all anonymous reviewers for their helpful suggestions and comments to improve the paper.

This research is funded by the National Science Foundation (NSF, grant No. 1810582, 1615837, 1526052), National Natural Science Foundation of China (NSFC, grant No. 61772307, 61402257), National Key R&D Program of China (grant No. 2017YFB0801700) and China Scholarship Council.

References

[1] R. Bhagwan, R. Kumar, R. Ramjee, G. Varghese, S. Mohapatra, H. Manoharan, and P. Shah, "Adtributor: Revenue debugging in advertising systems." in *NSDI*, 2014, pp. 43–55.

[2] C. Luo, J.-G. Lou, Q. Lin, Q. Fu, R. Ding, D. Zhang, and Z. Wang, "Correlating events with time series for incident diagnosis," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1583–1592.

[3] R. Fontugne, C. Pelsser, E. Aben, and R. Bush, "Pinpointing delay and forwarding anomalies using large-scale traceroute measurements," in *Proceedings of the 2017 Internet Measurement Conference (IMC)*. ACM, 2017, pp. 15–28.

[4] T. Yang, J. Jiang, P. Liu, Q. Huang, J. Gong, Y. Zhou, R. Miao, X. Li, and S. Uhlig, "Elastic sketch: Adaptive and fast network-wide measurements," in *Proceedings of the 2018 Conference of the ACM SIGCOMM*. ACM, 2018, pp. 561–575.

[5] M. Luckie and R. Beverly, "The impact of router outages on the AS-level internet," in *Proceedings of the Conference of the ACM SIGCOMM*. ACM, 2017, pp. 488–501.

[6] A.-M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report*, vol. 4, 2007.

[7] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting behind akamai (travelocity-based detouring)," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 435–446.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[9] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end path information to optimize CDN performance," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 2009, pp. 190–201.

[10] M. A. Warrior, U. Klarman, M. Flores, and A. Kuzmanovic, "Drongo: Speeding up CDNs with subnet assimilation from the client," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies(CoNEXT)*. ACM, 2017, pp. 41–54.

[11] Z. Li, D. Levin, N. Spring, and B. Bhattacharjee, "Internet anycast: performance, problems, & potential," in *Proceedings of the 2018 Conference of the ACM SIGCOMM*. ACM, 2018, pp. 59–73.

[12] A. Flavel, P. Mani, D. A. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev, "Fastroute: A scalable load-aware anycast routing architecture for modern CDNs," *connections*, vol. 27, p. 19, 2015.

[13] J. Pan, Y. T. Hou, and B. Li, "An overview of dns-based server selections in content distribution networks," *Computer Networks*, vol. 43, no. 6, pp. 695–711, 2003.

[14] E. Nygren, R. K. Sitaraman, and J. S. Sun, "The Akamai network: a platform for high-performance internet applications," *Operating Systems Review*, vol. 44, pp. 2–19, 2010.

[15] C. Contavalli, W. van der Gaast, D. C. Lawrence, and W. Kumari, "Client Subnet in DNS Queries," RFC 7871, May 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7871.txt

[16] F. Chen, R. K. Sitaraman, and M. Torres, "End-user mapping: Next generation request routing for content delivery," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 167–181.

[17] C. Huang, I. Batanov, and J. Li, "A practical solution to the client-LDNS mismatch problem," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 2, pp. 35–41, 2012.

[18] J. S. Otto, M. A. Sánchez, J. P. Rula, and F. E. Bustamante, "Content delivery and the natural evolution of DNS: remote DNS trends, performance issues and alternative solutions," in *Proceedings of the 2012 Internet Measurement Conference*. ACM, 2012, pp. 523–536.

[19] "Quad9," https://www.quad9.net, 2017.

[20] "Cloudflare DNS," https://1.1.1.1, 2018.

[21] "A Faster Internet: The Global Internet Speedup," http://afasterinternet.com, 2016.

[22] F. Streibelt, J. Böttger, N. Chatzis, G. Smaragdakis, and A. Feldmann, "Exploring edns-client-subnet adopters in your free time," in *Proceedings of the 2013 conference on Internet measurement conference(IMC)*. ACM, 2013, pp. 305–312.

[23] F. U. Sudrajat, "The state of adoption of DNS ECS extension on the internet," Ph.D. dissertation, Case Western Reserve University, 2017.

[24] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP geolocation databases: Unreliable?" *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 53–56, 2011.

[25] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang, "Towards street-level client-independent IP geolocation." in *NSDI*, vol. 11, 2011, pp. 27–27.

[26] R. Staff, "RIPE Atlas: A global internet measurement network," *Internet Protocol Journal*, vol. 18, no. 3, 2015.

[27] "Direct Peering with Google," https://peering.google.com/.

[28] "AWS Peering," https://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide.

[29] Y.-C. Chiu, B. Schlinker, A. B. Radhakrishnan, E. Katz-Bassett, and R. Govindan, "Are we one hop away from a better internet?" in *Proceedings of the 2015 Internet Measurement Conference*. ACM, 2015, pp. 523–529.

[30] K. Limthong, "Real-time computer network anomaly detection using machine learning techniques," *Journal of Advances in Computer Networks*, vol. 1, no. 1, 2013.

[31] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Transactions on signal processing*, vol. 51, no. 8, pp. 2191–2204, 2003.

[32] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 219–230.

[33] A. Khurshid, W. Zhou, M. Caesar, and P. Godfrey, "Veriflow: Verifying network-wide invariants in real time," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 49–54.

[34] B. M. Maggs and R. K. Sitaraman, "Algorithmic nuggets in content delivery," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 3, pp. 52–66, 2015.

[35] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan, "Mapping the expansion of google's serving infrastructure," in *Proceedings of the 2013 conference on Internet measurement conference(IMC)*. ACM, 2013, pp. 313–326.

[36] M. Flores, A. Wenzel, K. Chen, and A. Kuzmanovic, "Fury route: Leveraging CDNs to remotely measure network distance," in *International Conference on Passive and Active Network Measurement(PAM)*. Springer, 2018, pp. 87–99.

[37] Y. Gilad, A. Herzberg, M. Sudkovitch, and M. Goberman, "CDN-on-Demand: An affordable DDoS defense via untrusted clouds." in *NDSS*, 2016.