Pathfinder: Exploring Path Diversity for Assessing Internet Censorship Inconsistency

(A Technical Report of Internet Freedom Fund from Open Technology Fund, May 2023.)

Xiaoqin Liang
Old Dominion University
Norfolk, Virginia, USA

Guannan Liu
Virginia Tech
Arlington, Virginia, USA

Lin Jin
University of Delaware
Newark, Delaware, USA

Shuai Hao
Old Dominion University
Norfolk, Virginia, USA

Haining Wang
Virginia Tech
Arlington, Virginia, USA

ABSTRACT

Internet censorship is typically enforced by authorities to achieve information control for a certain group of Internet users. So far existing censorship studies have primarily focused on country-level characterization because (1) in many cases, censorship is enabled by governments with nationwide policies and (2) it is usually hard to control how the probing packets are routed to trigger censorship in different networks inside a country. However, the deployment and implementation of censorship could be highly diverse at the ISP (Internet Service Provider) level. In this paper, we investigate the Internet censorship from a different perspective by scrutinizing the diverse censorship deployment inside a country. Specifically, by leveraging an end-to-end measurement framework, we deploy multiple geo-distributed back-end control servers to explore various paths from one single vantage point. The generated traffic with the same domain but different control servers' IPs could be forced to traverse different transit networks, thereby being examined by different censorship devices if present. Through our large-scale experiments and in-depth investigation, we reveal that the diversity of Internet censorship caused by different routing paths inside a country is prevalent, implying that (1) the implementations of centralized censorship are commonly incomplete or flawed and (2) decentralized censorship is also common. Moreover, we identify that different hosting platforms also result in inconsistent censorship activities due to different peering relationships with the ISPs in a country. Finally, we present extensive case studies in detail to illustrate the configurations that lead to censorship inconsistency and explore the causes.

1 INTRODUCTION

Internet censorship has become increasingly pervasive as more governments and authorities rely on it to restrict users from accessing undesired content. In many cases, censorship policies are typically enforced by nations' authorities, resulting in that nationwide censorship could be largely consistent. Thus, most censorship studies aggregate the observed activities at the country level [30, 31, 40, 43, 44, 48, 50, 55]. However, the severity of censorship may be highly diverse at the ISP level, due to the difference of deployments and implementations, which makes the country-level characterization too coarse to draw reasonable conclusions for examining censorship deployment. Limited by the measurement methods, such diversity is mostly underestimated in previous work as researchers have no control on which transit networks or gateways the experiment packets will reach and thus cannot attribute inconsistent censorship behaviors inside a country to diverse censorship deployment.

Although some previous studies reported decentralized information control in certain specific countries [22, 51, 61], little research has examined the inconsistency of censorship enforcement in a systematic manner and at a global scale. One of the closest studies is [14], which leverages BGP churn to identify the Autonomous Systems (ASes) responsible for inducing censorship activities. However, it entirely relies on the measurement data from ICLab [40] that uses VPNs only, so it has limited coverage and may miss some key observations [31]. Moreover, the scale of network-level path diversity caused by BGP churn is limited and the occurrence of a path churn is random and cannot be controlled by the experiments. BreadCrumb [9] studies the DNS censorship changes due to router load balancing based on different packet parameters, *i.e.*, ephemeral source port and local bits of the source IP address. However, it is dedicated to investigating the impact of DNS censorship, and its routing changes relying on router load balancing only explore a small-scale routing variation inside a network.

In this paper, we perform an in-depth investigation on Internet censorship from a different perspective that is largely overlooked in previous studies, by which we scrutinize the inconsistency of censorship deployments and implementations inside a country. Specifically, we design and deploy a measurement framework called *Pathfinder* to identify inconsistent censorship activities experienced on different network paths from vantage points inside a country. Pathfinder leverages multiple geo-distributed back-end control servers as different destinations of probing packets to explore potential diverse routing paths. As such, the probing packets issued from the vantage points in the same country could be routed to different transit networks when using different destination IPs of control servers, resulting in diverse censorship behaviors due to inconsistent policies on different networks. In addition, our system design is also inspired by Disguiser [31], which achieves accurate censorship detection by providing the control server a static payload as the ground truth of server responses. We set up Pathfinder's control servers in the same way to accurately identify censorship activities without manual inspection.

Through Pathfinder, we perform a large-scale measurement study to understand the censorship inconsistency caused by diverse routing paths. We conduct an eight-month experiment (from

May 2022 to December 2022) by acquiring 88,347 vantage points from 120 countries with detected censorship activities, and reveal that such a phenomenon is quite prevalent, where 91.7% of countries (110) experience various extents of inconsistency and many of them have been commonly considered as having centralized censorship control. In particular, we observe that some paths can have a lower percentage of censorship activities than others, indicating that potential censorship circumvention could be explored. For instance, with the vantage points from India, the packets routed to the paths toward the control servers deployed in the Middle East experience much less censorship than other paths (8% vs. 40-60%).

Moreover, we uncover that different cloud hosting platforms also contribute to inconsistent censorship behaviors, due to their various peering connections or preference, which could cause the packets to be routed to different transit networks that have different censorship deployments. On the other hand, direct peering between cloud platforms and the eyeball networks inside a country could allow certain censorship circumvention, because the packets would enter cloud providers' private networks before reaching the censorship devices that are typically deployed at the nation's border networks. To illustrate an in-depth investigation for understanding this phenomenon, we perform extensive case studies of South Korea and India by leveraging application traceroutes to examine detailed censorship deployments.

The main contributions of this work are summarized as follows:

- Developing and deploying Pathfinder, an end-to-end framework to simultaneously explore multiple potential routing paths for identifying censorship activities when probing packets from one vantage point are routed to different ISPs or transit networks.
- Conducting extensive measurements and uncovering that the censorship deployments inside a country are largely inconsistent, even for many countries that are considered to have centralized censorship controls. We further show that certain paths could experience far less censorship, which can be exploited for potential censorship circumvention.
- Discovering that large cloud platforms can affect the occurrence of censorship due to peering connections. We leverage application traceroutes to perform case studies in two countries (South Korea and India) to collect detailed network paths and examine different peering configurations that lead to inconsistent censorship.

2 BACKGROUND

2.1 Censorship Techniques

Internet censorship can be achieved by various techniques [23], such as IP-layer censorship and application-layer censorship. IP-based censorship examines the destination IP address of a packet, and thus it could be easily evaded by the changes of service IP addresses. Besides, with the wide adoption of cloud and CDNs, the IP resources become increasingly shared and dynamic so that the IP-based blocking often results in collateral damage for legitimate services [20, 63]. Thus, application-layer censorship that can accurately block undesired content has been attracting more attention from both censorship regimes and research community, and is the focus of this work.

Application-layer Censorship. Application-layer censorship involves inspecting the information carried in the application-layer protocols, *e.g.*, domain-name-based blocking in DNS, HTTP, and HTTPS, and keyword-based filtering in packet streams.

Domain names explicitly reveal the Internet resources a client intends to access, which allows a censorship device to prevent users from accessing online content that is prohibited by authorities. When a user device resolves a domain name to its web server's IP address through DNS, the censor can directly learn the accessed domain as DNS traffic is typically unencrypted. In addition, the HTTP Host header presents the domain name that a client is visiting. Since the HTTP protocol is unencrypted, the censors can know exactly the requested domain. HTTPS encrypts all the HTTP packets after a TLS handshake so that the Host header is no longer visible to the censors, but commonly an SNI (Server Name Indication) extension is sent in plaintext in the Client Hello message to indicate which domain the client intends to visit.

In order to achieve finer control to prohibit undesired content, keyword-based censorship searches the sensitive keywords in unencrypted packet streams and disrupts the traffic when those predefined, forbidden keywords are detected. To do so, the censorship device intercepts and parses all HTTP traffic, and searches the keywords in certain locations of HTTP requests or responses such as request line, headers, or payload body [57].

Interference Techniques. The mechanisms used for blocking or filtering undesired Internet resources also vary. To perform DNS manipulation for denying a domain access, the censorship devices usually inject forged DNS responses that redirect the user's requests to a censor-controlled address (*e.g.*, that shows a *blockpage* indicating the domain being prohibited), a non-routable private IP address, or a public IP address that will be prohibited by the IP-layer blocking [25]. Additionally, in the case of TCP-based DNS, a censor typically injects an RST/FIN packet to tear down the TCP connection [31].

In the HTTP/HTTPS blocking, when a prohibited domain is detected (*i.e.*, from the HOST header in unencrypted HTTP messages or the plaintext Client Hello message in the initial HTTPS handshake), the censor can simply drop the HTTP/HTTPS request, causing a timeout on the client side. Alternatively, the censors could also tear down the connection by sending RST/FIN packets to both the client and the server of the requested domain, or respond to the client with a dedicated blockpage.

Inbound and Outbound Censorship. Censorship devices can be deployed to examine different directional traffic, *i.e.*, inbound and outbound censorship [55]. Inbound censorship refers to that censorship devices monitor and intercept the traffic in the inbound direction where the traffic originates from the networks outside the country/region. On the contrary, in outbound censorship, censorship devices inspect the outbound traffic that originates internally and traverses toward the destination outside their networks. Compared to blocking undesired content detected inbound to the network, outbound censorship is more common as censorship policies typically aim to control how sensitive content can be accessed by the users within the censoring regions.

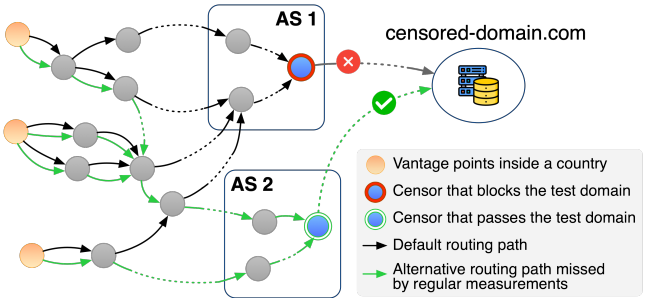

Figure 1: Illustration of inconsistent censorship in different paths due to different censorship implementations.

2.2 Application Traceroute

To better understand censorship, application traceroute has been explored to pinpoint the location of censorship devices and examine their behaviors [31, 50]. In particular, application traceroute increments the TTL field in the probing packets, while the application payload is set to trigger the censorship. Before reaching the censor, the probing packets will be dropped and an ICMP Time Exceeded message would be returned. As the TTL increases and the probing packet reaches the censorship device, the sign of interference (*e.g.*, RST/FIN packet) indicates the exact hop where the censorship device is located.

3 PATHFINDER

In this section, we introduce our methodology and framework, Pathfinder, to systematically investigate censorship inconsistency by measuring censorship activities from diverse paths simultaneously. We first discuss the phenomenon of censorship inconsistency and the challenges of examining this problem through existing censorship measurement platforms. We then describe the design of Pathfinder, as well as special design considerations that eliminate potential noise data when conducting large-scale experiments.

3.1 The Censorship Inconsistency Problem

Understanding censorship activities often involves sending a probing request that could trigger the censorship from a vantage point inside a country or region and detecting network interference by analyzing abnormal responses. With such a setup, however, the detection of censorship entirely relies on responses observed on the vantage points' side, and thus no path information can be learned and no location of censorship devices can be inferred. Consequently, censorship behaviors are usually characterized by aggregating the results at the country level. However, the censorship policies are usually deployed at the ISP level and different ISPs may have different censorship implementations, making such characterization inaccurate to unveil inconsistent censorship behaviors in a country.

Furthermore, traceroute or application-layer traceroute has been utilized to examine network path and censorship devices [31, 50], but it would suffer from limited coverage of measured networks for identifying inconsistent censorship behaviors. Figure 1 shows how such inconsistency is typically overlooked by existing measurement studies. To trigger censorship devices situated on the network path, the probing packets, including traceroutes, are sent through the vantage points in a country to the test domain. However, as the probing packets are all toward the same destination, the packets issued from distributed vantage points may converge to a certain few upstream transit networks, *e.g.*, AS 1 in Figure 1. On the other hand, AS 2 may implement different censorship policies for the test domain, but the regular measurements cannot identify such behavior due to routing configurations.

3.2 System Design

To systematically examine the phenomenon of censorship inconsistency, we design and deploy a measurement framework, Pathfinder, to detect censorship in various paths. Our design rationale is that if we are able to direct probing packets to traverse different networks inside a country, we can understand the censorship deployments from a different perspective in a finer-grained manner by characterizing the impact of diverse routing paths on the changes of censorship behaviors.

Overview. Figure 2 illustrates the system design of Pathfinder. At a higher level, Pathfinder consists of a set of vantage points distributed across the world and a set of back-end control servers that serve as the destination of probing packets. Also, a client feeds a country-specific test domain list into the vantage points to construct HTTP requests and schedule the measurements for initiating HTTP connections with each different control server. Specifically, each censored test domain is embedded into the HTTP packet header field to trigger censorship. On the other hand, control servers respond to all received HTTP requests with a static payload. Pathfinder then collects the response and connection status for each probing test at both client and control server sides for analysis.

Vantage Points. In order to issue probing requests and conduct experiments, we need to acquire a set of vantage points distributed across the world. In this study, we leverage residential IP proxies (RESIP) [37, 62] to issue probing requests. RESIP services enable customers to proxy generated traffic through a RESIP's entry node called the gateway server. The gateway server then further forwards the traffic to one of exit nodes in the RESIP's infrastructure, which eventually relays traffic to the destination as the traffic source (*i.e.*, the vantage points in the Pathfinder's framework). Moreover, as residential networks have been witnessed to experience more aggressive censorship than the vantage points in commercial infrastructures such as data centers or hosting platforms [40, 60], vantage points from RESIP services can provide a more comprehensive and representative coverage for examining censorship deployments. To this end, we eventually choose and subscribe Proxyrack [45], a popular and stable RESIP service that has been extensively analyzed and used in previous studies [30, 31, 37].

Identified Censored Domain List. As Pathfinder's main goal is to examine the censorship inconsistency, we acquire the existing list of censored domains identified from the previous study, Disguiser [31], to reduce the measurement efforts that purely aim to detect censorship occurrence. Disguiser compiles a test domain list that consists of popular domains and sensitive domains extracted from Alexa's top 1,000 domains [4] and the Citizen Lab [15], and produces a country-specific list of censored domains [52] that has

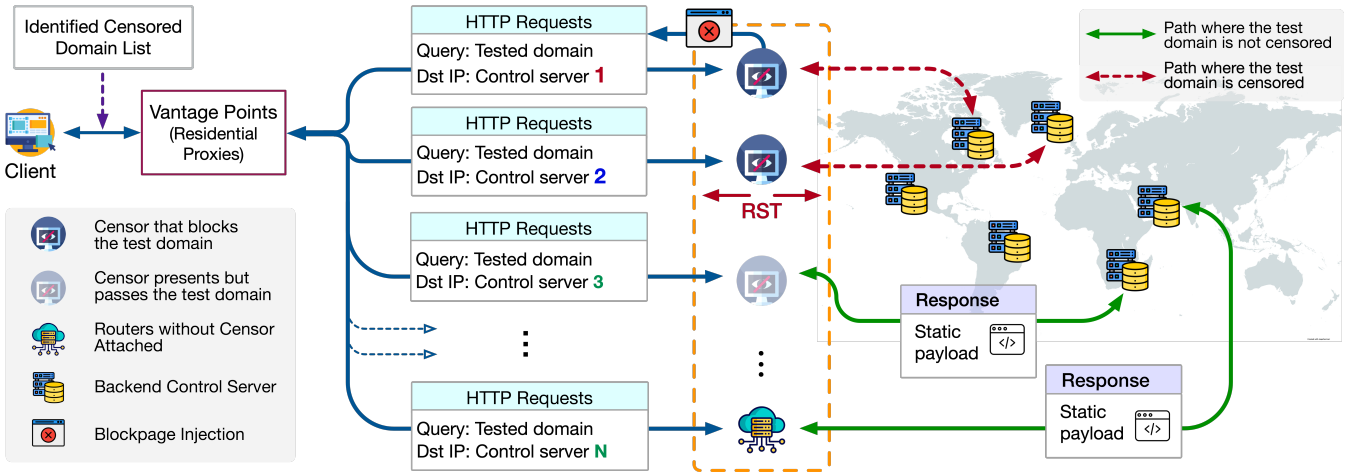

Figure 2: System design of Pathfinder and its exploitation of censorship inconsistency in diverse paths.

been extensively validated with Disguiser's measurements. By applying this censored-domain list to Pathfinder, in this study we only need to issue probing packets with those domains that have been observed to be censored by at least one censorship device in the country where the corresponding vantage point locates.

HTTP Requests. With each vantage point acquired from the RESIP, Pathfinder constructs a set of HTTP requests and sends them to trigger potential censorship on the network paths. In this study, we focus on the censorship for HTTP requests as it is the most prevalent censorship deployed by more countries [31, 40]. Each request is constructed as that the HOST header contains a domain name from the identified censored-domain list and the destination is set to one of the IP addresses of our control servers. Also, we set a 5-second timeout for each issued probing request so that we do not need to wait an extended period of time if a censorship device drops our requests. Accordingly, Pathfinder automatically retries four more times for the timeout requests before declaring it to be blocked by censorship, in case that some requests are dropped due to network congestion.

Control Server. Control servers play a key role in identifying censorship activities while establishing path diversity for various HTTP requests. First, similar to [31], the control servers provide a static payload for each HTTP request received. Such static payload is unique in a way that it should not collide with other legitimate domain pages or block pages, thereby providing a ground truth for efficiently detecting the occurrence of censorship. In addition, Pathfinder's design establishes multiple control servers and each server is deployed at different locations, so that the HTTP requests connecting different control servers could be routed differently due to different destinations, as shown in Figure 2. We detail the control server setup for our experiments in Section 4.

3.3 Special Design Considerations

Eliminating Cache Proxies. To obtain accurate results of censorship occurrence, we also need to consider a special scenario when a cache proxy is present on the network path. In this case, the cache proxy may directly respond to a probing request with the cached result for a test domain from previous connections, and the vantage point issuing the request will receive a response rather than the valid static payload of control servers, resulting in the interference for our measurements as the probing test would be classified as censorship occurrence.

To eliminate such potential impact on our measurements, similar to [31], we perform a real-time cache proxy test for each vantage point before we include it in the experiments, removing those that potentially have cache proxies situated on the path. In doing so, we first set up two reference servers serving the same domain under our control but with distinct destinations and payloads of landing pages. Then, we instruct the vantage points to sequentially probe the two reference servers by fetching the landing pages. If the cache proxy is present, the second probing could receive a response associated with the page of the first reference server which is cached through the first probing test.

By this means, we can successfully exclude most vantage points that are possibly impacted by cache proxies. However, we find that there is still a small portion of vantage points receiving responses from potential intermediate caches. This could be because a cache server selectively caches the content it passes through or it may intercept the connections by independently performing DNS resolution so as to obtain a different IP address other than our control server's. Therefore, we perform an offline check to exclude those suspicious vantage points. We fetch the legitimate landing pages of the test domains and simply compare their title tags with the responses received by vantage points. If a vantage point has obtained a valid title tag from the legitimate page of a test domain, it implies that the page could be retrieved and returned by the intermediate cache proxies. In total, we exclude 164 acquired vantage points from our measurements by offline check.

Eliminating Inbound Censors. As mentioned in Section 2.1, censorship devices can appear as inbound censors or outbound censors. Since we mainly focus on the censorship enforced by the countries in which the vantage points located, the inbound censorship deployed by the countries where control servers are deployed may

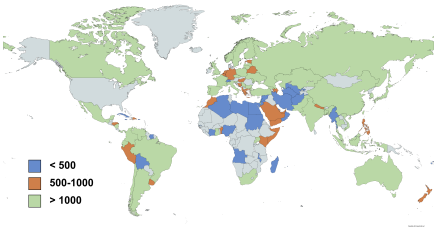

Figure 3: Distribution of vantage points across all countries in §4.1.

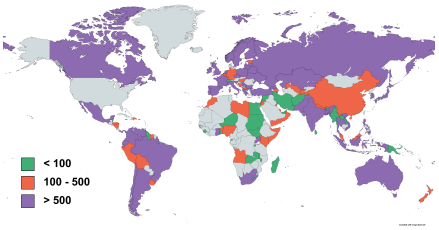

Figure 4: Distribution of vantage points across all countries in § 4.2.

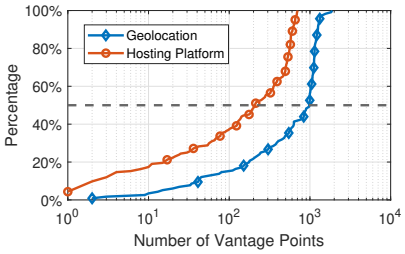

Figure 5: CDF of the number of vantage points in § 4.1 and § 4.2.

also be identified as censorship cases by Pathfinder, which could introduce false cases in our results.

To avoid the impact of inbound censorship, we carefully choose the locations of control servers to deploy them in the countries where no censorship has been widely identified in previous studies. Besides this, we further conduct experiments to verify that there is no inbound censorship on the control servers' side before conducting measurements. We set up 50 VPN servers from hide-my-ip [24] located in different countries and send a series of probing packets to all of our control servers. Each of these packets carries a test domain in the censored-domain list. We identify that, all of the probing packets with non-censored domains in the corresponding country of the VPN node successfully retrieve the pre-defined static payload from our control servers.

4 EXPERIMENTS

To investigate the censorship inconsistency in real-world scenarios, we conduct two experiments that focus on the primary causes of path diversity that lead to censorship inconsistency: *locations of destinations* and *hosting platforms*. In this section, we provide a detailed introduction to the methodologies of both experiments. We also discuss potential ethical issues that may arise from our experiments and our approach to mitigate them.

4.1 Censorship Inconsistency by IP Destinations

In this experiment, we explore the diverse network paths in packet routing due to different geographically distributed destination addresses. Differing from existing research focusing on network paths induced by router load balancing [9], this experiment is carefully designed so that domain requests originating from a vantage point take various network paths to reach their destinations.

Vantage Points. As mentioned in Section 3.2, a vantage point serves as a generic access node in a specific region that we can utilize to measure censorship activities. We use proxies provided by Proxyrack as our vantage points. These proxies are primarily residential, hence traffic from our vantage points simulates real network connections as if it is from regular users.

Moreover, as residential proxies are randomly assigned, one issue we need to address is the uneven distribution of provided proxies across different countries. For instance, we receive a significantly larger number of proxies from several countries including South Korea, Japan, and India, while some countries such as Brazil and South Africa are underrepresented. To rectify this, we establish a cap in Pathfinder, limiting the number of vantage points to a maximum of 80 per country per week. Additional vantage points assigned to us through Proxyrack are discarded, ensuring a more balanced distribution of vantage points for our experiment. This increases the likelihood of observing censorship activities for a larger number of countries while maintaining a substantial amount of data collected from each country.

Control Servers. This experiment aims to investigate censorship activities across various network paths. To achieve this, we have carefully selected our control servers based on the following two criteria. First, our control servers must be geographically spread across the world. This increases the likelihood of requests taking diverse network paths to reach these servers. Second, the control servers must be located in places with no known censorship activities, which eliminates the possibility of inbound censorship interfering with our experiment (Section 3.3). Based on these criteria, we establish six control servers using AWS EC2 instances. These servers are located in Virginia (North America - East), California (North America - West), São Paulo (South America), London (Europe), Bahrain (Middle East), and Cape Town (Africa).

Data Collection. We conduct our measurement experiments for 36 weeks, during which we initiate 47,430,642 requests to our control servers. 30,188,931 (64%) of them retrieve our static payloads successfully, and others encounter censorship in the network paths. Figure 3 shows the distribution of vantage points across all countries. In total, we obtain 88,347 vantage points in 120 countries that have been observed with censorship behaviors in previous studies. Also, these countries represent 6.9 billion (83%) global populations according to the UN population division [16], demonstrating that our experiment setup can comprehensively capture the real-world censorship activities experienced by the majority of Internet users in the world. In addition, the blue line in Figure 5 plots the CDF distribution of the number of vantage points collected in different countries. More than 50% of the countries have vantage points of over 1,000, indicating that we collect a substantial amount of data for each country.

4.2 Censorship Inconsistency by Hosting Platforms

In addition to the locations of IP destinations, we observe that the hosting platform is another important factor that leads to censorship inconsistency. This is because different peering policies would cause network packets to be routed through diverse networks that

enforce different censorship policies. Therefore, we schedule extensive measurements by establishing control servers across different hosting platforms. As such, probing requests from vantage points are routed differently to reach these platforms, which enables us to further evaluate censorship inconsistency caused by various hosting platforms.

Vantage Points. We employ the same method described in Section 4.1 to obtain vantage points from the RESIP, Proxyrack. However, each vantage point assigned by Proxyrack has a limited lifespan, which is not sufficient to gather censorship activities for both experiments. Due to this limitation, we use a different set of vantage points for this experiment.

Control Servers. In order to gain a comprehensive understanding of the impact of hosting platforms on censorship inconsistency, we set up control servers across three large popular cloud platforms: Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure (Azure). To mitigate the impact of geolocation on censorship inconsistency, we specifically select the locations where all three platforms have data centers deployed. As such, the locations of data centers that we choose to host control servers are Virginia (United States), Sydney (Australia), Paris (France), and São Paulo (South America). With that, we establish one control server in each of these locations from each of the platforms, which forms a set of 12 control servers in total.

Data Collection. Our data collection for this set of experiments lasts for 20 weeks, during which we measure censorship activities from 51,609 vantage points across 115 countries. Figure 4 presents a heatmap showing the worldwide distribution of these vantage points. Also, the red line in Figure 5 illustrates the CDF distribution of the number of vantage points collected in this experiment, showing that more than 50% of the countries have over 200 vantage points. Although the number of vantage points per country is not as high as that in our IP destination experiment in Section 5.2 due to a shorter period, we believe that it is still sufficient for a large-scale measurement study.

4.3 Ethical Considerations

Ethical concerns in censorship-related measurement studies have been extensively discussed in previous works [31, 32, 44, 55]. In this study, we leverage a RESIP, Proxyrack, as vantage points to issue probing requests, similar to previous studies [31]. Proxyrack is a benefit-driven platform that recruits residential proxies worldwide, where participants can willingly opt-in to join the network and perform tasks for financial profit [1]. To reduce the potential risk for the participants of the residential proxies, Pathfinder's design ensures that our experiments do not generate traffic to the actual servers associated with the testing domains so as to avoid the access of the actual undesired content. Also, we avoid using the same residential proxy frequently, to further minimize the risk of the proxy owners.

Furthermore, we provide a comprehensive description of our experiment in the static payload of our control servers, along with our contact information. Throughout the entire duration of our experiment, we have not received any concerns regarding our measurements and data collection. Finally, measuring censorship activities using Pathfinder does not involve human participation or the collection of personal information on any individual or entity, so that our study is exempt from the scope of the institutional Internal Review Board (IRB) [32].

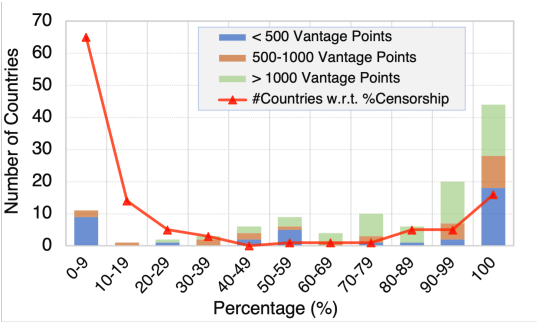

Figure 6: Distribution of the number of countries with respect to their censorship percentage (*line graph*) and inconsistency percentage (*stacked bar graph*).

5 RESULTS & ANALYSIS

In this section, we answer a series of research questions regarding the existence, cause, and exploitation of censorship inconsistency, using the censorship behavior data collected from the experiments described in Section 4:

- In general, how prevalent is the censorship inconsistency across different countries and domains? (§ 5.1)
- How extensively do destination servers' locations impact censorship inconsistency in different countries? (§ 5.2)
- How extensively do different hosting platforms lead to censorship inconsistency? (§ 5.3)
- Can specific paths that experience less censorship be leveraged to achieve detour for censorship circumvention? (§ 5.4)

5.1 Prevalence of Censorship Inconsistency

To answer the first research question, we investigate the prevalence of censorship inconsistency in different countries by analyzing our results from two perspectives: vantage-point-level inconsistency and domain-level inconsistency.

Vantage-Point-Level Inconsistency. Pathfinder sends requests from vantage points provided by Proxyrack to our control servers through various network paths, enabling us to identify inconsistent censorship behaviors across different network paths. In this study, we consider a vantage point being censored if any measurement request originating from the vantage point is censored. The censorship percentage is defined as the number of censored vantage points divided by the total number of vantage points being evaluated. More importantly, we define the inconsistency percentage as the percentage of the censored vantage points that observe inconsistent censorship behaviors across different network paths.

Country	Vantage Points			%	%
	Total	Censored	Incons.	Censored	Incons.
Kazakhstan	1,101	1,101	522	100.00%	47.41%
Pakistan	1,009	1,009	1,006	100.00%	99.70%
South Korea	1,883	1,881	1,279	99.89%	68.00%
China	1,038	1,028	1,020	99.04%	99.22%
Russia	1,322	1,115	668	84.34%	59.91%
Bangladesh	1,092	882	748	80.77%	84.81%
Turkey	1,137	918	406	80.74%	44.23%
Thailand	1,200	865	606	72.08%	70.06%
Vietnam	1,246	768	581	61.64%	75.65%
India	1,284	754	749	58.72%	99.34%

Table 1: List of top 10 countries with more than 1,000 vantage points by the highest censorship percentage and their inconsistency percentage (*Incons.*).

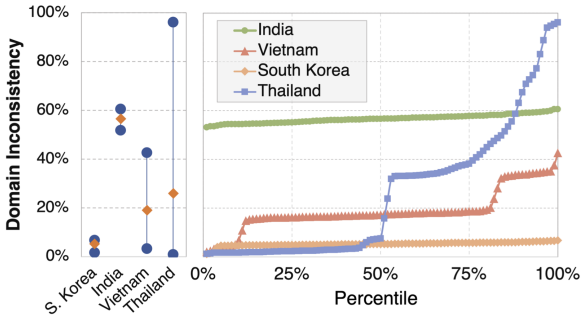

Figure 7: The min-average-max chart (*left bar graph*) and the percentiles of censorship inconsistency (*right line graph*) in South Korea, India, Vietnam, and Thailand.

The censorship percentage implies the extent of censorship deployment and the inconsistency percentage indicates the prevalence of inconsistent censorship behaviors.

Figure 6 displays the distribution of the number of countries according to their censorship percentage (the red line). We observe that the majority of countries exhibit an all-or-nothing censorship behavior, similar to Bhaskar *et al.* [9]. Specifically, 79 (69%) countries experience less than 20% censorship percentage, while 26 (22%) show more than 80%. Only 11 (9%) countries have a censorship percentage between 20% to 80%. These results align with our expectations, as censorship devices are designed to be uniformly deployed across all vantage points.

Figure 6 also presents a bar graph that shows the distribution of the number of countries over censorship inconsistency percentage. Our experiment reveals that only 11 (10%) countries have an inconsistency percentage below 9%, indicating that censorship inconsistency is prevalent worldwide. Moreover, we observe severe censorship inconsistency in a significant number of countries. Specifically, 20 (18%) countries have their inconsistency percentage fall into the range of 90%-99%, and even more countries (44) exhibit 100% censorship inconsistency, where all vantage points located in these countries observe various extents of inconsistent censorship among different network paths.

Additionally, the stacked bars in Figure 6 show the different numbers of vantage points we evaluated in our experiment. We collect a substantial amount of data from countries with more than 1,000 vantage points (green bars), allowing us to comprehensively understand the censorship behaviors in these countries, by which we observe 29 out of 51 (57%) countries experiencing more than 90% censorship inconsistency. On the other hand, 39 countries with less than 500 vantage points (blue bars) have also shown 100% censorship inconsistency. These results show that our experiments have sufficiently demonstrated the prevalence of censorship inconsistency, even for those counties with fewer vantage points.

We perform further analysis on countries with more than 1,000 vantage points. Table 1 lists the top 10 countries with the highest censorship percentage. We can see that all 10 countries listed in the table demonstrate censorship percentages exceeding 50%, with two of them (Kazakhstan and Pakistan) having 100% of vantage points experiencing censorship. Meanwhile, the highlighted countries also exhibit significant censorship inconsistency, for instance, Pakistan and China show both censorship and inconsistency percentages exceeding 99%. This suggests that, despite implementing strict censorship policies, censorship inconsistency is still prevalent in those countries, resulting in the possibility that evades the censorship by routing through a different path.

Domain-Level Inconsistency. In addition to the prevalence of censorship inconsistency encountered by vantage points, we here analyze the inconsistent censorship activities from the perspective of the packets with different blocked domains. We consider a domain as being censored if we observe any measurement requests containing that domain name being blocked by censorship devices in a country. We then define a domain-level censorship percentageas the number of probing requests that are blocked by censorship devices divided by the total number of requests we sent in our experiments for a country.

The results show that domain-level censorship inconsistency varies, and is prevalent across many countries. Figure 7 plots the min-average-max chart and the percentiles of domain-level inconsistency in South Korea, India, Vietnam, and Thailand. Specifically, the height of the bar/curve of each country indicates the inconsistency of censorship activities in each country in terms of different domains. We can see that Thailand and Vietnam demonstrate higher-level of domain-level censorship inconsistency, indicating that the censored domains in both countries experience significantly different censorship activities. Among these four countries in Figure 7, Thailand shows the most prevalent censorship inconsistency at the domain level. For instance, we observe that a domain, `www.livejasmin.com`, encounters only 1% of censorship activities, whereas 96% of the packets containing `www.bongacams.com` are blocked by censorship, suggesting that both centralized and decentralized censorship may be implemented. Vietnam also demonstrates high censorship inconsistency, with a minimum domain-level censorship of 3% and a maximum of 43%. On the other hand, India and South Korea show a lower censorship inconsistency across different blocked domains, suggesting that each blocked domain in India and South Korea has a similar chance of encountering censorship.

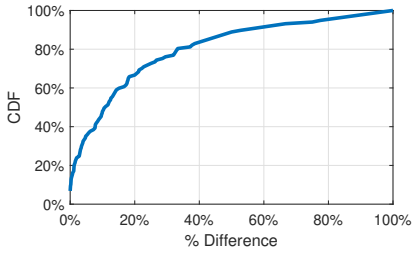

Figure 8: CDF distribution of the percentage difference for censorship inconsistency across different countries.

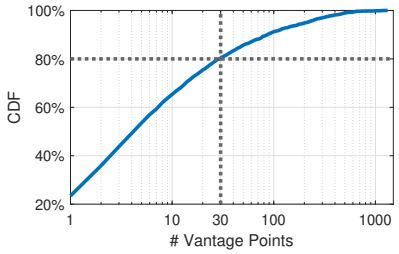

Figure 9: CDF distribution of the number of collected vantage points in different ASes.

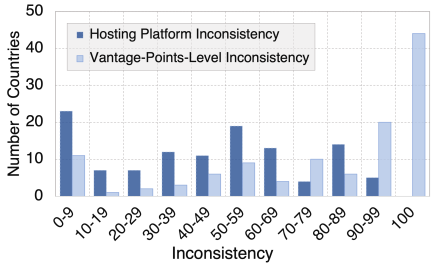

Figure 10: Distribution of censorship inconsistency in terms of different hosting platforms.

5.2 Impact of IP Destinations on Censorship Inconsistency

In this section, we further break down the results presented in Section 5.1 to answer the second research question. As introduced in Section 4.1, we utilize RESIP as vantage points and set up six geographically distributed control servers in various regions of Amazon AWS, enabling our probing requests to traverse different network paths. Then, Pathfinder identifies censorship activities in each network path to examine the impact of the locations of control servers' destinations on censorship inconsistency.

To analyze the extent of censorship inconsistency for different countries, we first calculate the difference between the highest censorship percentage and the lowest censorship percentage of a network path in each country, plotted in Figure 8. The results show that over 50% of countries experience censorship inconsistency greater than 10%, while 20% of countries have the inconsistency of over 40%. This confirms that censorship activities vary widely across network paths toward different control servers.

Figure 11 further illustrates the observed censorship percentages for the network paths toward each control server in three countries, India, Bangladesh, and Thailand. Specifically, we see that India exhibits the most inconsistent censorship activities among various paths. Only 8.4% of the requests toward the control server Bahrain (Middle East) experience censorship, while censorship occurs between 37% to 60% for requests to other control servers. Similarly, Bangladesh's vantage points show the probing requests toward Virginia (North America East) and Bahrain with comparatively less censorship. On the other hand, we observe inconsistent censorship behavior from Thailand where requests sent to Virginia's control server experience significantly higher censorship activities compared to the requests sent to other control servers. These results demonstrate that the location of the destination servers significantly influences the network paths, leading to inconsistent censorship activities.

AS-level Analysis. Next, we extend our analysis from the country level to the Autonomous System (AS) level to gain more fine-grained insights into censorship inconsistency. During our experiments, the vantage points are acquired from a total of 2,959 ASes through the RESIP, with a diverse number of obtained vantage points ranging from 1 vantage point in AS 3238 and AS 63023 to 1,319 in AS 12297. Figure 9 depicts the distribution of the number of vantage points in

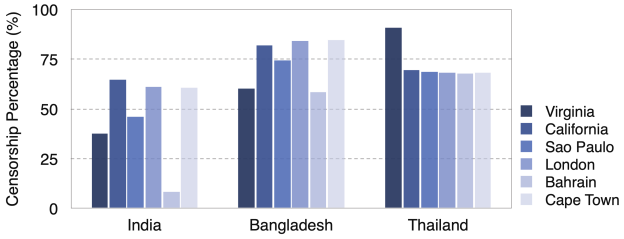

Figure 11: Censorship percentages to different control servers by the vantage points in India, Bangladesh, and Thailand.

different ASes. Here, we filter out the ASes with less than 30 vantage points (represented by the dotted line) in our analysis to focus on those well-represented ASes with sufficient vantage points.

Here, we quantify the AS-level censorship inconsistency by measuring the difference between the maximum and minimum censorship percentage encountered by probing packets from an AS sent to different control servers. Table 2 presents the top 10 ASes with the highest censorship inconsistency between different network paths. All ASes in the list experience at least 88% censorship inconsistency, with AS 24086 showing the highest censorship inconsistency of 97.3%. Additionally, these ASes can be categorized into two groups. While AS 24086, AS 7552, AS 48004, AS 23969, AS 131090, and AS 24550 show a single path with a significantly higher censorship percentage than other paths, AS 132298, AS 134946, AS 59362, and AS 134877 have one or two paths with considerably low censorship activities. Importantly, we identify four network paths with no censorship detected, highlighted in Table 2. The observations here at the AS level further strengthen the conclusion that the destination of the packets can lead to noticeable inconsistent censorship, as shown in the country-level analysis in Figure 11.

5.3 Impact of Hosting Platforms on Censorship Inconsistency

Besides the location of destinations which is one of the primary factors contributing to censorship inconsistency, our experiments also reveal that different hosting platforms (*e.g.*, large cloud service providers) also lead to censorship inconsistency since the network paths, through which the request traverses, can be significantly

ASN (Country)	Censorship Percentage						Inconsist.
	Virginia	California	São Paulo	London	Bahrain	Cape Town	
AS 24086 (VN)	1.5%	1.9%	98.5%	1.5%	1.2%	2.3%	97.3%
AS 7552 (VN)	0.6%	0.6%	97.8%	0.7%	0.8%	1.9%	97.2%
AS 48004 (UA)	5.4%	4.5%	4.8%	98.2%	5.1%	68.3%	93.7%
AS 132298 (BD)	92.6%	93.4%	92.6%	92.6%	0.0%	92.2%	93.4%
AS 134946 (IN)	90.9%	73.6%	80.9%	73.6%	0.0%	72.7%	90.9%
AS 23969 (TH)	3.1%	92.6%	3.9%	3.3%	2.5%	3.5%	90.1%
AS 59362 (BD)	95.5%	94.4%	94.4%	92.1%	5.6%	95.5%	89.9%
AS 131090 (TH)	3.0%	92.1%	2.5%	2.6%	2.6%	3.0%	89.6%
AS 134877 (IN)	84.5%	0.0%	89.1%	84.5%	0.0%	84.5%	89.1%
AS 24550 (NP)	11.4%	7.5%	6.3%	6.3%	94.1%	5.9%	88.2%

Table 2: List of top 10 ASes with the highest censorship inconsistency across various network paths. (*Inconsist.: Inconsistency, measured by the greatest difference of AS-level censorship percentages***)**

Country	# of Requests	% Censorship			Inconsist.
		AWS	GCP	Azure	
South Korea	441,615	81.37%	**37.81%**	98.47%	60.66%
Venezuela	59,299	7.49%	20.85%	65.49%	58.00%
Japan	53,500	29.64%	42.14%	86.29%	56.65%
Indonesia	23,629	9.88%	19.95%	64.31%	54.42%
Mexico	36,585	14.79%	14.52%	66.86%	52.34%
India	186,135	63.38%	**22.19%**	55.85%	41.19%
Thailand	217,625	59.34%	85.42%	48.10%	37.33%
Belarus	34,616	56.63%	55.97%	85.51%	29.54%
Ukraine	56,215	20.95%	21.56%	43.83%	22.88%
Russia	933,287	88.53%	81.36%	94.58%	13.22%

Table 3: Top 10 countries with more than 800 vantage points and their censorship inconsistency measured by the difference of censorship percentages.

influenced due to the different peering relationships between the hosting platforms and ISPs.

In general, the network paths taken by probing requests are determined by the routing policies and peering relationships established between the ASes from the source to the destination. By setting up control servers in different hosting platforms, the requests would be routed to traverse through diverse network paths due to different connections between upstream ISPs and hosting platforms, enabling us to evaluate the censorship inconsistency resulting from different hosting platforms. To investigate such cases, we set up control servers in the same location but at different cloud platforms (including Amazon AWS, Google Cloud, and Microsoft Azure) as the destinations of probing packets, as described in Section 4.2.

We then define the censorship inconsistency of hosting platforms as the largest difference of censorship percentage encountered by probing packets inside a country sent to different hosting platforms. Figure 10 illustrates the distribution of the number of countries with respect to such hosting platform inconsistency (shown as dark blue). The results reveal that 55 (47.8%) countries have censorship inconsistency of more than 50%, indicating that inconsistent censorship behaviors toward various hosting platforms could be a common phenomenon. Compared to the vantage-point-level inconsistency from Section 5.1 (*i.e.*, the bar graph in Figure 6, shown as light blue in Figure 10), the hosting platform inconsistency does not show any apparent correlation. This indicates that our observation here is indeed the impact of hosting platforms, not being coincidentally affected by the destinations of control servers in different clouds.

Furthermore, we conduct additional analysis focusing on the most well-represented countries studied through our experiments. Table 3 shows the top 10 countries with at least 800 acquired vantage points. We can see that the censorship inconsistency with respect to different hosting platforms ranges from 13% to 60%. In particular, South Korea demonstrates significantly lower censorship in GCP with only 37.81% (marked in red in Table 3). In contrast, network paths toward AWS and Azures demonstrate 81.37% and 98.47% censorship percentages, respectively. India shows a similar trend, with paths to GCP showing significantly lower censorship (22.19%) than the other hosting platforms (63.38% in AWS and 55.85% in Azure). Both countries are highlighted in Table 3, and we further conduct detailed case studies for these two countries through application traceroute to investigate the root cause of this censorship inconsistency, which we elaborate on in Section 6.

Table 3 illustrates inconsistent censorship caused by different cloud platforms. To further break down the results into different control servers hosted in each cloud, we examine the censorship percentages associated with each individual control server. Figure 12 illustrates the detailed results of three countries, South Korea, Japan, and India, comparing the censorship percentage observed on the paths toward each control server hosted on GCP, AWS, and Azure. The figure shows that the censorship percentage varies significantly on each hosting platform, confirming the aggregated results at the hosting platform level. Moreover, we can see that censorship percentages for all three countries are relatively stable with GCP and Azure, indicating that (1) the location of our control servers does not have a significant impact on the censorship percentage in these two platforms and (2) GCP and Azure may establish more consistent connections with the networks in these countries. On the other hand, network paths to AWS show a more inconsistent censorship percentage among different locations, implying that its peering

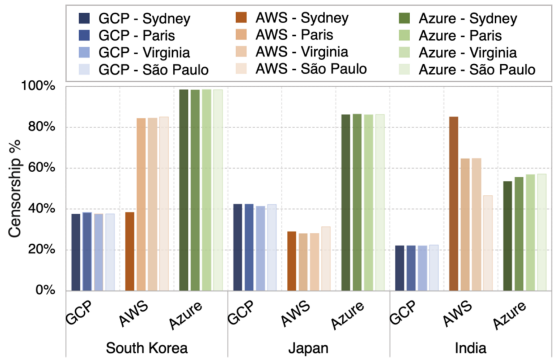

Figure 12: Censorship percentage from South Korea, Japan, and India to the control servers in GCP, AWS, and Azure.

connections may be more diverse. This also aligns with our previous discussion about the impact of IP destinations on censorship inconsistency in Section 5.2.

5.4 Potential of Censorship Circumvention

Censorship inconsistency provides users with a potential strategy to evade censorship and gain access to censored content. Previous sections have discussed that the location of the destination and the hosting platforms can significantly impact the routing paths of requests, implying that censorship circumvention can be potentially achieved by the careful selection of outbound network paths.

To circumvent censorship for certain domains, one straightforward strategy could be to deliberately route the requests to proxy servers located where less/no censorship appears in the network path. For example, as shown in Figure 11, vantage points in India experience significantly fewer censorship activities towards the control server located in Bahrain (Middle East). Hence, selecting proxy servers in Bahrain may unblock the majority of the requests coming from India. On the other hand, Thailand demonstrates that requests toward the control server located in Virginia would experience more aggressive censorship, hence avoiding routing requests to this location would be a desired option in terms of censorship circumvention.

In addition to the location of destinations, censorship could also be evaded with the careful selection of the hosting platforms of proxy servers. As highlighted in Table 3, vantage points from both South Korea and India observe significantly low censorship activities in the network paths toward GCP. With that, it is preferred to deliberately select proxy servers hosted on GCP for evading censorship. Also, service providers could intentionally host their services on less-censored platforms, ensuring that users have a better chance to access their services with less interference.

Moreover, such strategies could be integrated with the existing circumvention framework, to actively and dynamically explore such potential circumventing paths and apply them as an additional vector in the circumvention toolkit. These strategies could also be combined with other methods, *e.g.*, CenFuzz [50] and Geneva [11], to create a more efficient and robust, yet still simple method to achieve censorship circumvention.

6 APPLICATION TRACEROUTE: CASE STUDIES

As shown in Section 5.3, the hosting platform could largely affect the severity of censorship activities experienced by users from certain countries. For instance, we observed that South Korea and India show lower censorship activities in network paths toward Google Cloud Platform (GCP) than Amazon AWS and Microsoft Azure. However, the RESIP vantage points leveraged by Pathfinder cannot perform traceroute to pinpoint the exact location on a network path where censorship occurs, because it is not feasible to set the TTL values of the relayed packets due to the fact that RESIPs usually operate above the transport layer. Therefore, in this section, we leverage application traceroute with commercial VPNs as vantage points to conduct a detailed, end-to-end investigation for censorship inconsistency in the above two countries.

6.1 Methodology

As described in Section 2.2, application traceroute works by sending a series of requests with incremental TTLs and is accomplished when receiving a sign of censorship or our pre-defined static payload. By examining the corresponding response of each request, we are able to reconstruct the network path taken by the probing requests and identify the specific hop where the censorship device is deployed. To collect such path information, we rely on commercial VPNs to perform the application traceroute experiments.

A prior study [58] reveals that many VPN services may lie about their server locations. As the authenticity of the location of vantage points are important to our analysis, we validate whether the VPN servers are located where they advertise. To achieve this, we first attempt to trigger censorship from a VPN server by accessing some censored domains. We then confirm the location of the VPN servers if we receive official blockpages established by the censoring countries (*i.e.*, South Korean or Indian government in our case studies, see Section 6.2 and 6.3). By examining some VPN service providers, we observe that hide-my-ip [24]'s VPN servers advertise relatively reliable location information, and hence, we use it to acquire vantage points.

We reuse the same set of control servers introduced in Section 4.2 as destinations for the application traceroute. These servers are hosted by various cloud providers (Amazon AWS, Google Cloud, and Microsoft Azure) and are geographically located across the globe (Sydney, Paris, Virginia, and São Paulo). For both studied countries (South Korea and India), we randomly select 32 domains from the censored domains list (Section 3.2) and initiate the application traceroute carrying these domains from each of the VPN servers to each of the control servers.

6.2 Case Study: South Korea

Leveraging the application traceroute, we are able to identify the network devices on the paths. Figure 13 shows a concrete example of the paths taken by the probing requests between our vantage points in Seoul, South Korea, and distributed control servers around the world. We observe that censorship consistently occurs on the paths to three control servers hosted by Amazon AWS. Specifically, we receive an official blockpage (through the redirection to

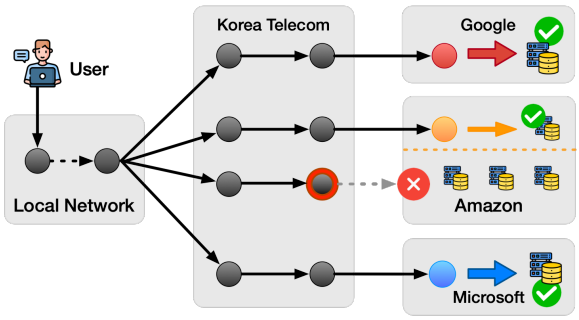

Figure 13: Application traceroute of different paths over different hosting platforms in Seoul, South Korea.

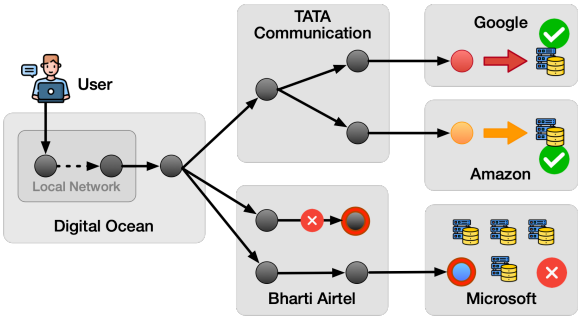

Figure 14: Application traceroute of different paths over different hosting platforms in Bangalore, India.

`http://warning.or.kr`) indicating that the accessed domain is prohibited. The remaining paths, including one path to Amazon AWS in Sydney, did not encounter censorship activities. This aligns with our general observations in Figure 12, where AWS shows more diverse censorship behaviors and the paths to AWS's Sydney site experience less censorship. This could be caused by the censorship devices on this path applying a different domain list that is less aggressive than others.

On the other hand, we observe no censorship occurring on the paths toward Google Cloud and Microsoft Azure. Although we observe that the probing packets, including those to AWS, are all routed with the same ISP (AS 4766, Korea Telecom), the paths are different from certain hops, which leads to inconsistent censorship. More interestingly, we observe the upstream ISP (Korea Telecom) peered with GCP's private networks within South Korea, resulting in the probing requests entering the private networks of GCP before reaching the censorship devices. Table 5 in Appendix B provides more detailed application traceroute results for one domain, including all routers' ASes and ISPs on the paths.

6.3 Case Study: India

Next, we conduct a case study with application traceroute in India and explore censorship inconsistency in depth. Figure 14 depicts one case where the hosting platforms play a key role in inconsistent censorship behaviors. Our experiments reveal that censorship always occurs on the paths to Microsoft Azure's servers by receiving an official block page stating domains containing sensitive information were prohibited. However, we did not encounter any censorship when retrieving static payloads from control servers on Amazon AWS and GCP.

Table 6 in Appendix B displays a detailed illustration of application traceroute performed in Bangalore, India. Through the traceroute results, we observe specific censorship devices deployed in the ASes including AS 9498 (Bharti Airtel) and AS 8057 (Microsoft Azure). However, as censorship is not likely to be enabled by Microsoft's public cloud, we consider that the censorship devices should still be located at AS 9498 (Bharti Airtel). We speculate this experiment results could be interfered with by some censorship devices copying TTL values from the original probing packet, resulting in increased TTL values when tracing the censor's location. Such TTL-copying behavior by censorship devices has been detected and extensively examined in prior study [31].

On the other hand, the packets sent to GCP and AWS are both routed through AS 4755 (TATA Communication) and experience no censorship. Furthermore, similar to the case in South Korea, we observe that no censorship occurred on the paths to the control servers on GCP while no intermediate hops are visible before reaching the control servers, implying that the probing packets are routed to GCP's private network by a direct peer between GCP and AS 4755.

Comparatively, the results of the application traceroute in South Korea show all probing requests to different clouds passed through one upstream provider (AS 4766) before encountering any censorship. In India, the requests issued from one vantage point are routed through two different providers, where one of the provider's networks (AS 9498) enables censorship for the test domain. As shown in Table 3, both South Korea and India experience less censorship on paths toward GCP's control servers. This is due to the peering between local ISPs and GCP's private networks, resulting in no censorship on paths toward GCP's control servers.

7 LIMITATIONS

Censorship Close to Vantage Points. Pathfinder aims to examine censorship deployments by varying probing packets' destinations to explore diverse paths. As shown in our experimental results, it can largely identify different censorship implementations in the various upstream transit networks. However, although censorship devices are more commonly deployed close to a nation's border networks [31], censorship enforced at the hops close to clients also exists [50] but may not be detected by Pathfinder. In such a case, the vantage points would observe consistent censorship behaviors because all probing packets are routed by the same ISP.

Vantage Point Selection. In this study, we utilize the RESIP provided by Proxyrack [45] as vantage points to investigate inconsistent censorship activities in various network paths. We obtain a total of 88,347 vantage points from 120 countries and 51,609 vantage points from 115 countries in our IP destination and hosting platform experiments, respectively. While many countries are extensively studied with a large number of vantage points, other countries, such as Switzerland, Cuba, and Luxembourg, are only collected less

than 50 vantage points, due to limited available residential nodes from Proxyrack. Also, to examine the censorship inconsistency in more detail, we performed case studies using application traceroute through VPNs, instead of RESIPs (Section 6), because RESIPs typically do not support changing the TTLs of issued packets to accomplish application traceroute. However, although VPNs are also widely used in existing studies [40, 48, 50], VPN servers are commonly hosted in commercial data centers, where the traffic may encounter less censorship than in residential networks. These vantage point biases could be rectified by recruiting more vantage points from different platforms, but the skewness of node distribution exists for all these kinds of global measurements.

Test Domain List. Our experiments utilize the existing censored domain list collected from Disguiser [31] to reduce measurement efforts for identifying censored domains. While the domains on the list have been validated with their censorship status, this previous study does not consider the impact of censorship inconsistency, so it may miss some censored domains if the probing requests encounter no censorship on certain testing network paths. However, this should not impact our study much because an incomplete list of censored domains is still sufficient to explore diverse network paths and illustrate the censorship deployments.

8 RELATED WORK

8.1 Global-Scale Censorship Measurements

Nowadays, many countries deploy Internet censorship to prohibit undesired content from being accessed by users. This motivates researchers to develop various mechanisms to measure and understand censorship activities on a global scale. OONI [21] (Open Observatory of Network Interference) has established a community-driven global measurement framework by recruiting participants to run pre-defined measurements to investigate censorship activities. VabderSloot *et al.* [55] proposed Quack, a remote measurement system that efficiently explores application-layer interference by using the Echo protocol. Built upon Quack, Raman *et al.* [49] presented FilterMap, an improved version of Quack to identify the content filtering techniques by analyzing their block pages. Niaki *et al.* [40] developed ICLab, a platform that employs VPNs as the vantage points to launch a variety of longitudinal censorship measurements such as DNS manipulation and other network interference, and proposed a technique to identify unknown blockpages. More recently, Censored Planet [48] integrates multiple existing techniques/frameworks and enables synchronized censorship measurements to enhance data representativeness and coverage. In addition, Pearce *et al.* [44] introduced Iris, a system designed to identify and characterize DNS censorship on a global scale. Then, Bhaskar *et al.* [9] further presented BreadCrumb, a tool that measures DNS censorship variation by manipulating the source parameters of probe packets and router-based load balancing. Lastly, Jin *et al.* [31] presented Disguiser, an end-to-end, ground truth based measurement platform that detects censorship activities and reveals the censor deployment.

Our study complements existing research efforts by designing Pathfinder. Using Pathfinder, we systematically investigate censorship activities in various network paths on a global scale, revealing the wide existence of censorship inconsistency.

8.2 Country-Specific Censorship Studies

While significant research efforts on censorship studies have been undertaken on a global scale, many previous research studies have also been devoted to investigating censorship behaviors in specific countries. As one of the largest censorship systems in the world, the Great Firewall of China (GFW) has been extensively studied [5–7]. Xu *et al.* [60] examined China's border ASes networks and its relationship with foreign countries. Ensafi *et al.* [17] measured the reachability of servers and networks blocked by the GFW using connectivity measurement techniques. Hoang *et al.* [26] developed GFWatch to examine the DNS filtering behaviors of the GFW. Besides GFW, other research studies have focused on Internet censorship deployed by Iran [8, 10], Pakistan [35, 38], Syria [13], India [61], Kazakhstan [47], and Russia [51].

Although similar inconsistent or decentralized censorship activities have been witnessed in several country-specific studies, their observations typically rely on the collaboration from activists on the ground, and the used methods/data cannot be extended to other countries. Instead, Pathfinder aims to systematically examine the censorship inconsistency on a global scale by exploring diverse network paths.

8.3 Censorship Circumvention

With the ever-increasing censorship activities on the Internet, effective censorship circumvention techniques have also been proposed in recent decades. Tschantz *et al.* [53] provided a systematization of knowledge on censorship techniques and circumvention approaches. More specifically, Fifield *et al.* [20] proposed the Domain Fronting technique to avoid censorship detection by concealing the domain names of the communication partners. Furthermore, this technique has also been widely adopted by many other circumvention systems such as Lantern [36] and Psiphon [46]. Burnett *et al.* [12] developed Collage, enabling users to exchange messages in cover traffic. Khattak *et al.* [34] introduced an analysis model that inspects the evasion vulnerabilities discovered by Network Intrusion Detection System (NIDS). Autosonda [29] is a tool designed to study web filters and discover a range of implementation and decision-making techniques. Nisar *et al.* [41] proposed C-Saw, a circumvention system that integrates censorship measurements with circumvention techniques into a single system. Bock *et al.* [11] proposed Geneva, a genetic algorithm that automates the discovery of censorship circumvention strategies against on-path network censors. Wang *et al.* [56] exploited the discrepancies in TCP state machines of deep packet inspection (DPI) implementation as means of bypassing censorship. Raman *et al.* [50] developed Cenfuzz, which employs different HTTP methods to circumvent censorship devices and identify the evasion behavior of vendors by clustering. Other circumvention tools include CDNBrowsing systems [27, 39, 63], Decoy routing [28, 33], Flash proxy [19], Infranet [18], Telex [59], uProxy [54], Alkasir [3], LASTor [2], Astoria [42], *etc.*

Our study uncovers censorship inconsistencies, which can also be leveraged as a complementary component to be integrated with existing censorship circumvention, as discussed in Section 5.4.

9 CONCLUSION

Internet censorship is the control or suppression mechanism on what online content users can access. In this study, we designed and implemented Pathfinder, an end-to-end framework for investigating inconsistent censorship activities in different network paths inside a country. Our findings reveal the prevalent existence of censorship inconsistency in many countries due to the changes of network paths. We further demonstrated that geolocation and hosting platforms of destination servers often result in the requests being routed through various network paths and encountering inconsistent censorship activities. We showed that such censorship inconsistency can be exploited to circumvent censorship in many countries. To further investigate censorship inconsistency, we leveraged the application traceroute to identify the exact node in a network path where such inconsistency occurs.

ACKNOWLEDGMENT

This work was supported by an Internet Freedom Fund from Open Technology Fund (OTF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] Proxyrack Become a Peer. https://www.proxyrack.com/become-a-peer/.
[2] Masoud Akhoondi, Curtis Yu, and Harsha V Madhyastha. LASTor: A low-latency AS-aware Tor Client. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2012.
[3] Walid Al-Saqaf. Internet Censorship Circumvention Tools: Escaping the Control of the Syrian Regime. *Media and Communication*, 2016.
[4] Alexa. https://www.alexa.com/topsites.
[5] Anonymous. The Collateral Damage of Internet Censorship by DNS Injection. *ACM SIGCOMM Computer Communication Review*, 2012.
[6] Anonymous. Towards a Comprehensive Picture of the Great Firewall's DNS Censorship. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.
[7] Anonymous, Arian Akhavan Niaki, Nguyen Phong Hoang, Phillipa Gill, and Amir Houmansadr. Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2020.
[8] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. Internet Censorship in Iran: A First Look. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2013.
[9] Abhishek Bhaskar and Paul Pearce. Many Roads Lead To Rome: How Packet Headers Influence DNS Censorship Measurement. In *Proceedings of the USENIX Security Symposium*, 2022.
[10] Kevin Bock, Yair Fax, Kyle Reese, Jasraj Singh, and Dave Levin. Detecting and Evading Censorship-in-Depth: A Case Study of Iran's Protocol Whitelister. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2020.
[11] Kevin Bock, George Hughey, Xiao Qiang, and Dave Levin. Geneva: Evolving Censorship Evasion Strategies. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.
[12] Sam Burnett, Nick Feamster, and Santosh Vempala. Chipping Away at Censorship Firewalls with User-Generated Content. In *Proceedings of the USENIX Security Symposium*, 2010.
[13] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. Censorship in the Wild: Analyzing Internet Filtering in Syria. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2014.
[14] Shinyoung Cho, Rishab Nithyanand, Abbas Razaghpanah, and Phillipa Gill. A Churn for the Better: Localizing Censorship Using Network-level Path Churn and Network Tomography. In *Proceedings of the International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2017.
[15] Citizen Lab. URL Testing Lists Intended for Discovering Website Censorship. https://github.com/citizenlab/test-lists/, 2019.
[16] The World Bank Population Database. https://data.worldbank.org/indicator/SP.POP.TOTL.
[17] Roya Ensaf, Philipp Winter, Abdullah Mueen, and Jedidiah R. Crandall. Analyzing the Great Firewall of China Over Space and Time. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2015.
[18] Nick Feamster, Magdalena Balazinska, Greg Harfst, Hari Balakrishnan, and David R Karger. Infranet: Circumventing Web Censorship and Surveillance. In *USENIX Security Symposium*, 2002.
[19] David Fifield, Nate Hardison, Jonathan Ellithorpe, Emily Stark, Dan Boneh, Roger Dingledine, and Phil Porras. Evading Censorship with Browser-Based Proxies. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2012.
[20] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. Blocking-resistant Communication through Domain Fronting. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2015.
[21] Arturo Filastò and Jacob Appelbaum. OONI: Open Observatory of Network Interference. In *USENIX Workshop on Free and Open Communications the Internet (FOCI)*, 2012.
[22] Genevieve Gebhart and Tadayoshi Kohno. Internet Censorship in Thailand: User Practices and Potential Threats. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017.
[23] Joseph Lorenzo Hall Hall, D. Aaron Michael, Amelia Andersdotter, Ben Jones, Nick Feamster, and Mallory Knodel. A Survey of Worldwide Censorship Techniques. https://datatracker.ietf.org/doc/draft-irtf-pearg-censorship/.
[24] HideMyIP. https://www.hide-my-ip.com/.
[25] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. How Great is the Great Firewall? Measuring China's DNS Censorship. In *Proceedings of the USENIX Security Symposium*, 2021.
[26] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. How Great is the Great Firewall? Measuring China's DNS Censorship. In *Proceedings of the USENIX Security Symposium*, 2021.
[27] John Holowczak and Amir Houmansadr. CacheBrowser: Bypassing Chinese Censorship Without Proxies Using Cached Content. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015.
[28] Amir Houmansadr, Giang T. K. Nguyen, Matthew Caesar, and Nikita Borisov. Cirripede: Circumvention Infrastructure using Router Redirection with Plausible Deniability. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2011.
[29] Jill Jermyn and Nicholas Weaver. Autosonda: Discovering rules and triggers of censorship devices. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2017.
[30] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. Understanding the Impact of Encrypted DNS on Internet Censorship. In *Proceedings of The Web Conference (WWW)*, 2021.
[31] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. Understanding the Practices of Global Censorship through Accurate, End-to-End Measurements. *ACM SIGMETRICS*, 2022.
[32] Ben Jones, Roya Ensafi, Nick Feamster, Vern Paxson, and Nick Weaver. Ethical Concerns for Censorship Measurement. In *ACM SIGCOMM Workshop on Ethics in Networked Systems Research (NS Ethics)*, 2015.
[33] Josh Karlin, Daniel Ellard, Alden W. Jackson, Christine E. Jones, Greg Lauer, David P. Mankins, and W. Timothy Strayer. Decoy Routing: Toward Unblockable Internet Communication. In *USENIX Workshop on Free and Open Communications the Internet (FOCI)*, 2011.
[34] Sheharbano Khattak, Mobin Javed, Philip D. Anderson, and Vern Paxson. Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion. In *USENIX Workshop on Free and Open Communications the Internet (FOCI)*, 2013.
[35] Sheharbano Khattak, Mobin Javed, Syed Ali Khayam, Zartash Afzal Uzmi, and Vern Paxson. A Look at the Consequences of Internet Censorship Through an ISP Lens. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2014.
[36] Lantern. https://lantern.io/.
[37] Xianghang Mi, Xuan Feng, Xiaojing Liao, Baojun Liu, XiaoFeng Wang, Feng Qian, Zhou Li, Sumayah Alrwais, Limin Sun, and Ying Liu. Resident Evil: Understanding Residential IP Proxy as a Dark Service. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2019.
[38] Zubair Nabi. The Anatomy of Web Censorship in Pakistan. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2013.
[39] Milad Nasr, Hadi Zolfaghari, Amir Houmansadr, and Amirhossein Ghafari. MassBrowser: Unblocking the Censored Web for the Masses, by the Masses. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2020.
[40] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2020.
[41] Aqib Nisar, Aqsa Kashaf, Ihsan Ayyub Qazi, and Zartash Afzal Uzmi. Incentivizing Censorship Measurements via Circumvention. In *Proceedings of the ACM SIGCOMM Conference*, 2018.

[42] Rishab Nithyanand, Oleksii Starov, Adva Zair, Phillipa Gill, and Michael Schapira. Measuring and mitigating AS-level adversaries against Tor. *arXiv preprint arXiv:1505.05173*, 2015.
[43] Paul Pearce, Roya Ensafi, Frank Li, Nick Feamster, and Vern Paxson. Augur: Internet-Wide Detection of Connectivity Disruptions. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2017.
[44] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. Global Measurement of DNS Manipulation. In *Proceedings of the USENIX Security Symposium*, 2017.
[45] Proxyrack. `https://www.proxyrack.com/`.
[46] Psiphon. `https://www.psiphon.ca/`.
[47] Ram Sundara Raman, Leonid Evdokimov, Eric Wurstrow, J. Alex Halderman, and Roya Ensaf. Investigating Large Scale HTTPS Interception in Kazakhstan. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2020.
[48] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensaf. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020.
[49] Ram Sundara Raman, Adrian Stoll, Jakub Dalek, Armin Sarabi, Reethika Ramesh, Will Scott, and Roya Ensafi. Measuring the Deployment of Network Censorship Filters at Global Scale. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2020.
[50] Ram Sundara Raman, Mona Wang, Jakub Dalek, Jonathan Mayer, and Roya Ensafi. Network Measurement Methods for Locating and Examining Censorship Devices. In *Proceedings of the International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2022.
[51] Reethika Ramesh, Ram Sundara Raman, Matthew Bernhard, Victor Ongkowijaya, Leonid Evdokimov, Anne Edmundson, Steven Sprecher, Muhammad Ikram, and Roya Ensafi. Decentralized Control: A Case Study of Russia. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2020.
[52] Disguiser: End to-End Framework for Measuring Censorship with Ground Truth. `https://github.com/e2ecensor/Disguiser_public`.
[53] Michael Carl Tschantz, Sadia Afroz, Anonymous, and Vern Paxson. SoK: Towards Grounding Censorship Circumvention in Empiricism. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2016.
[54] uProxy. `https://www.uproxy.org/`.
[55] Benjamin VanderSloot, Allison McDonald, Will Scott, J. Alex Halderman, and Roya Ensafi. Quack: Scalable Remote Measurement of Application-Layer Censorship. In *Proceedings of the USENIX Security Symposium*, 2018.
[56] Zhongjie Wang, Shitong Zhu, Yue Cao, Zhiyun Qian, Chengyu Song, Srikanth V. Krishnamurthy, Kevin S. Chan, and Tracy D. Braun. SymTCP: Eluding Stateful Deep Packet Inspection with Automated Discrepancy Discovery. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2020.
[57] Zachary Weinberg, Diogo Barradas, and Nicolas Christin. Chinese Wall or Swiss Cheese? Keyword Filtering In The Great Firewall Of China. In *Proceedings of the Web Conference (WWW)*, 2021.
[58] Zachary Weinberg, Shinyoung Cho, Nicolas Christin, Vyas Sekar, and Phillipa Gill. How to Catch When Proxies Lie: Verifying the Physical Locations of Network Proxies with Active Geolocation. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2018.
[59] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J Alex Halderman. Telex: Anticensorship in the Network Infrastructure. In *USENIX Security Symposium*, 2011.
[60] Xueyang Xu, Z. Morley Mao, and J. Alex Halderman. Internet Censorship in China: Where Does the Filtering Occur? In *Proceedings of the Passive and Active Network Measurement (PAM)*, 2011.
[61] Tarun Kumar Yadav, Akshat Sinha, Devashish Gosain, Piyush Kumar Sharma, and Sambuddho Chakravarty. Where The Light Gets In: Analyzing Web Censorship Mechanisms in India. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2018.
[62] Mingshuo Yang, Yunnan Yu, Xianghang Mi, Shujun Tang, Shanqing Guo, Yilin Li, Xiaofeng Zheng, and Haixin Duan. An Extensive Study of Residential Proxies in China. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.
[63] Hadi Zolfaghari and Amir Houmansadr. Practical Censorship Evasion Leveraging Content Delivery Networks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

APPENDIX

A OBSERVED CENSORSHIP INCONSISTENCY IN COUNTRIES

Country	Percentage	# of Countries
Albania, Angola, Argentina, Bolivia, Bosnia Herzegovina, Brazil, Czechia Dominican Republic, El Salvador, Estonia, Georgia, Ghana, Honduras, Kenya Mexico, Nepal, Nicaragua, Norway, Panama, Peru, Philippines, Poland, Puerto Rico Serbia, Slovenia, South Africa, Spain, Sweden, Tajikistan, Uganda, Uruguay	100%	31
Algeria, Australia, Bahrain, Belgium, Bulgaria, Chile, China, Colombia Ecuador, France Germany, Greece, India, Indonesia, Japan Kuwait, Netherlands, Pakistan, Portugal, United Kingdom	90% - 99%	20
Venezuela, Suriname, Moldova, Bangladesh, Singapore, Hong Kong	80% - 89%	6
Israel, Italy, Lithuania, Morocco, New Zealand, Taiwan, Thailand, Vietnam	70% - 79%	9
Belarus, Macao, South Korea, Canada	60% - 69%	4
Egypt, French Polynesia, Latvia, Russia, Ukraine	50% - 59%	5
Hungary, Iran, Kazakhstan, Nigeria, Somalia, Turkey	40% - 49%	6
Finland, Malaysia, United Arab Emirates, Yemen	20% - 39%	4
Azerbaijan, Brunei, Qatar, Saudi Arabia	0% - 19%	4

Table 4: Fraction of VPs that Observe Censorship Inconsistency in Censored Countries/Regions > 200

B APPLICATION TRACEROUTE RESULTS

Table 5 displays a detailed application traceroute result that depicts the paths taken by probing requests from one vantage point located in Seoul, South Korea, to 12 control servers deployed in three clouds for the censored domain `torrentdada.com`.

Table 6 presents the application traceroute from one vantage point located in Bangalore, India, to the control servers for the censored domain `cckerala.com`.

Control servers are located in Sydney, Paris, Virginia, and São Paulo and are hosted on three cloud platforms: Amazon AWS, Google Cloud Platform, and Microsoft Azure. At each hop in the traceroute, we show the corresponding IPs and associated ASes. We have also highlighted the censors in red to indicate the censorship occurrences to demonstrate the inconsistent activities identified on different paths.

Hops	Traceroutes to Control Servers											
	Amazon AWS				Google Cloud				Microsoft Azure			
	Sydney	Paris	Virginia	São Paulo	Sydney	Paris	Virginia	São Paulo	Sydney	Paris	Virginia	São Paulo
ttl = 1	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network	10.238.0.1 local network
ttl = 2	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network	172.32.2.10 local network
ttl = 3	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network	172.30.10.17 local network
ttl = 4	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network	172.30.10.9 local network
ttl = 5	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom	119.196.0.77 AS4766 Korea Telecom
ttl = 6	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom	112.190.32.21 AS4766 Korea Telecom
ttl = 7	*	* 112.174.90.110	* 112.174.90.226	* 112.174.90.154	*	*	*	*	*	*	*	*
ttl = 8	*	AS4766 Korea Telecom	AS4766 Korea Telecom	AS4766 Korea Telecom	*	*	*	*	*	*	*	*
ttl = 9	112.191.117.101 AS4766 Korea Telecom	**Censor**: 112.174.91.218 AS4766 Korea Telecom	**Censor**: 112.174.91.182 AS4766 Korea Telecom	**Censor**: 112.174.86.238 AS4766 Korea Telecom	128.134.10.246 AS4766 Korea Telecom	128.134.10.246 AS4766 Korea Telecom	128.134.10.246 AS4766 Korea Telecom	128.134.10.246 AS4766 Korea Telecom	121.189.3.138 AS4766 Korea Telecom	121.189.3.138 AS4766 Korea Telecom	121.189.3.138 AS4766 Korea Telecom	121.189.3.138 AS4766 Korea Telecom
ttl = 10	112.191.118.177 AS4766 Korea Telecom				34.151.125.165 (Sydney) Google Cloud	34.163.60.19 (Paris) Google Cloud	35.245.157.97 (Virginia) Google Cloud	35.247.224.42 (São Paulo) Google Cloud	104.44.239.244 AS8075 Microsoft Corp.	104.44.239.240 AS8075 Microsoft Corp.	104.44.239.244 AS8075 Microsoft Corp.	104.44.239.244 AS8075 Microsoft Corp.
ttl = 11	211.47.31.78 AS4766 Korea Telecom								104.44.22.41 AS8075 Microsoft Corp.	104.44.22.21 AS8075 Microsoft Corp.	104.44.22.41 AS8075 Microsoft Corp.	104.44.22.49 AS8075 Microsoft Corp.
ttl = 12	150.222.116.153 Amazon Tech. KR Seoul								104.44.19.241 AS8075 Microsoft Corp.	104.44.30.15 AS8075 Microsoft Corp.	104.44.19.241 AS8075 Microsoft Corp.	104.44.28.247 AS8075 Microsoft Corp.
ttl = 13	54.239.123.133 AS16509 Amazon.com KR Seoul								104.44.17.109 AS8075 Microsoft Corp.	104.44.30.16 AS8075 Microsoft Corp.	104.44.17.109 AS8075 Microsoft Corp.	104.44.16.241 AS8075 Microsoft Corp.
ttl = 14	*								104.44.7.231 AS8075 Microsoft Corp.	104.44.7.13 AS8075 Microsoft Corp.	104.44.7.231 AS8075 Microsoft Corp.	104.44.19.112 AS8075 Microsoft Corp.
ttl = 15	15.230.212.61 Amazon Tech. Australia Sydney								104.44.19.112 AS8075 Microsoft Corp.	104.44.17.67 AS8075 Microsoft Corp.	104.44.17.182 AS8075 Microsoft Corp.	104.44.17.182 AS8075 Microsoft Corp.
ttl = 16	15.230.210.56 Amazon Tech. Australia Sydney								104.44.17.203 AS8075 Microsoft Corp.	104.44.29.50 AS8075 Microsoft Corp.	104.44.17.182 AS8075 Microsoft Corp.	104.44.19.85 AS8075 Microsoft Corp.
ttl = 17	15.230.210.91 Amazon Tech. Australia Sydney								*	104.44.11.226 AS8075 Microsoft Corp.	104.44.30.191 AS8075 Microsoft Corp.	104.44.16.5 AS8075 Microsoft Corp.
ttl = 18	15.230.210.152 Amazon Tech. Australia Sydney								104.44.7.123 AS8075 Microsoft Corp.	*	104.44.30.191 AS8075 Microsoft Corp.	104.44.30.107 AS8075 Microsoft Corp.
ttl = 19	15.230.211.34 Amazon Tech. Australia Sydney								*	*	104.44.16.161 AS8075 Microsoft Corp.	104.44.28.254 AS8075 Microsoft Corp.
ttl = 20	*								*	*	104.44.28.254 AS8075 Microsoft Corp.	104.44.28.223 AS8075 Microsoft Corp.
ttl = 21	*								*	*	104.44.28.223 AS8075 Microsoft Corp.	104.44.29.44 AS8075 Microsoft Corp.
ttl = 22	*								*	*	104.44.29.44 AS8075 Microsoft Corp.	104.44.17.216 AS8075 Microsoft Corp.
ttl = 23	*								*	*	104.44.28.120 AS8075 Microsoft Corp.	104.44.19.169 AS8075 Microsoft Corp.
ttl = 24	*								*	*	104.44.16.180 AS8075 Microsoft Corp.	104.44.16.214 AS8075 Microsoft Corp.
ttl = 25	*								*	*	104.44.21.224 AS8075 Microsoft Corp.	104.44.22.60 AS8075 Microsoft Corp.
ttl = 26	3.26.215.12 (Sydney) Amazon AWS								*	*	*	*
ttl = 27									*	*	*	*
ttl = 28									20.5.0.129 (Sydney) Microsoft Azure	*	*	*
ttl = 29										*	*	*
ttl =30										*	*	*
ttl = 31										20.199.11.24 (Paris) Microsoft Azure	*	*
ttl = 32											*	*
ttl = 33											*	*
ttl = 34											*	*
ttl = 35											*	*
ttl = 36											*	191.234.198.54 (São Paulo) Microsoft Azure
ttl = 37											*	
ttl = 38											20.115.40.63 (Virginia) Microsoft Azure	

Table 5: Application traceroute results with `torrentdada.com` in Seoul, South Korea (experiments conducted December 2022).

Hops	Traceroutes to Control Servers											
	Amazon AWS				Google Cloud				Microsoft Azure			
	Sydney	Paris	Virginia	São Paulo	Sydney	Paris	Virginia	São Paulo	Sydney	Paris	Virginia	São Paulo
ttl = 1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1	10.238.0.1
ttl = 2	*	*	*	*	*	*	*	*	*	*	*	*
ttl = 3	10.66.7.17	10.66.7.5	10.66.7.7	10.66.7.7	10.66.6.237	10.66.6.247	10.66.6.227	10.66.6.245	10.66.7.7	10.66.6.247	10.66.7.21	10.66.6.237
ttl = 4	138.197.249.22 DigitalOcean	138.197.249.18 DigitalOcean	138.197.249.14 DigitalOcean	138.197.249.0 DigitalOcean	138.197.249.0 DigitalOcean	138.197.249.18 DigitalOcean	138.197.249.22 DigitalOcean	138.197.249.0 DigitalOcean	138.197.249.0 DigitalOcean	138.197.249.22 DigitalOcean	138.197.249.22 DigitalOcean	138.197.249.22 DigitalOcean
ttl = 5	219.65.110.189 AS4755 TATA Comm.	219.65.110.185 AS4755 TATA Comm.	219.65.110.189 AS4755 TATA Comm.	219.65.110.185 AS4755 TATA Comm.	219.65.110.185 AS4755 TATA Comm.	219.65.110.189 AS4755 TATA Comm.	219.65.110.189 AS4755 TATA Comm.	219.65.110.185 AS4755 TATA Comm.	202.56.198.57 AS9498 Bharti Airtel	202.56.198.29 AS9498 Bharti Airtel	202.56.198.29 AS9498 Bharti Airtel	202.56.198.29 AS9498 Bharti Airtel
ttl = 6	*	*	*	*	*	*	*	*	*	116.119.57.97 AS9498 Bharti Airtel	*	*
ttl = 7	180.87.36.9 AS6453 TATA Comm. (America)	180.87.39.25 AS6453 TATA Comm. (America)	180.87.39.25 AS6453 TATA Comm. (America)	180.87.39.25 AS6453 TATA Comm. (America)	121.240.1.46 AS4755 TATA Comm.	121.240.1.46 AS4755 TATA Comm.	121.240.1.46 AS4755 TATA Comm.	121.240.1.46 AS4755 TATA Comm.	116.119.109.205 AS9498 Bharti Airtel	*	116.119.104.151 AS9498 Bharti Airtel	*
ttl = 8	180.87.36.41 AS6453 TATA Comm. (America)	*	180.87.39.21 AS6453 TATA Comm. (America)	180.87.39.21 AS6453 TATA Comm. (America)	34.151.125.165 (Sydney) Google Cloud	34.163.60.19 (Paris) Google Cloud	35.245.157.97 (Virginia) Google Cloud	35.247.224.42 (São Paulo) Google Cloud	**Censor**: 116.119.94.30 AS9498 Bharti Airtel	*	**Censor**: 116.119.94.32 AS9498 Bharti Airtel	182.79.239.193 AS9498 Bharti Airtel
ttl = 9	180.87.7.18 AS6453 TATA Comm. (America)	80.231.131.1 AS6453 TATA Comm. (America)	80.231.130.106 AS6453 TATA Comm. (America)	*						198.200.130.17 AS8075 Microsoft Corp.		**Censor**: 116.119.57.158 AS9498 Bharti Airtel
ttl = 10	*	80.231.62.57 AS6453 TATA Comm. (America)	66.110.96.62 AS6453 TATA Comm. (America)	66.110.96.62 AS6453 TATA Comm. (America)						**Censor**: 104.44.41.235 AS8075 Microsoft Corp.		
ttl = 11	*	80.231.20.82 AS6453 TATA Comm. (America)	66.110.96.58 AS6453 TATA Comm. (America)	66.110.96.58 AS6453 TATA Comm. (America)								
ttl = 12	*	*	*	*								
ttl = 13	*	*	*	*								
ttl = 14	*	*	*	*								
ttl = 15	*	*	*	*								
ttl = 16	*	*	*	*								
ttl = 17	*	*	*	*								
ttl = 18	*	*	*	*								
ttl = 19	*	*	*	*								
ttl = 20	*	*	*	*								
ttl = 21	*	*	*	*								
ttl = 22	*	35.180.190.69 (Paris) Amazon AWS	*	54.240.244.102 AS16509 Amazon.com								
ttl = 23	*		*	*								
ttl = 24	3.26.215.12 (Sydney) Amazon AWS		*	*								
ttl = 25			54.197.194.180 (Virginia) Amazon AWS	*								
ttl = 26				18.228.203.42 (São Paulo) Amazon AWS								

Table 6: Application traceroute results for `cckerala.com` in Bangalore, India (experiments conducted December 2022).