Pathfinder: Exploring Path Diversity for Assessing Internet Censorship Inconsistency

Xiaoqin Liang*, Guannan Liu†, Lin Jin‡, Shuai Hao*, Haining Wang§
*Old Dominion University, Norfolk, VA, USA
†Colorado School of Mines, Golden, CO, USA
‡University of Delaware, Newark, DE, USA
§Virginia Tech, Arlington, VA, USA
{xlian001, shao}@odu.edu, guannan.liu@mines.edu, linjin@udel.edu, hnw@vt.edu

***Abstract*—Internet censorship is commonly enabled by authorities to enforce information control. So far, existing censorship studies have largely focused on country-level characterization, primarily because (1) censorship enforcement is often mandated through nationwide policies and (2) it is difficult to control the routing of probing packets to trigger censorship across different networks within a country. However, censorship mechanisms can vary significantly at the ISP level, revealing a more diverse landscape than previously assumed. In this paper, we investigate Internet censorship from a new perspective by scrutinizing diverse censorship deployments within a country. We design and deploy a measurement framework that utilizes multiple geo-distributed backend servers to probe various network paths from a single vantage point. By generating traffic targeting the same domain but different backend server IPs, we induce path diversity that exposes the traffic to distinct transit networks, and potentially, different censorship devices, thereby enabling a more granular analysis of censorship practices. Through our large-scale experiments and in-depth analysis, we reveal that diverse censorship resulting from varying routing paths within a country is widespread, implying that (1) the implementations of centralized censorship are commonly incomplete or flawed and (2) decentralized censorship is also prevalent. Moreover, we find that different hosting platforms also contribute to inconsistent censorship behavior due to their varying peering relationships with ISPs within a country. Finally, we present detailed case studies to illustrate the configurations that lead to such inconsistencies and to explore their underlying causes.**

1. Introduction

Internet censorship has become increasingly pervasive, with a growing number of governments relying on it to restrict users' access to undesired content. Conventional insights assume that censorship is typically enforced at the national level, resulting in relatively consistent policies across a country. Consequently, prior studies have predominantly focused on country-level characterizations, often aggregating observations without accounting for intra-national variation [32], [33], [44], [49], [55], [57], [62].

However, in practice, the severity and implementations of censorship can be highly diverse at the ISP level, making the country-level results too coarse to accurately capture the deployment landscape. For instance, a recent study [68] reveals that the Great Firewall of China (GFW), which has long been regarded as a censorship system with centralized policies and unified implementation, has emerged with additional regional censorship at the provincial level. Due to methodological limitations, such diversity has been largely underestimated in prior work, as researchers typically lack control over the transit networks or gateways that experimental traffic traverses and thus cannot attribute inconsistent censorship behaviors with a country to diverse censorship deployment.

Although some prior studies have reported decentralized information control in specific countries [24], [71], [58], there remains a lack of systematic methodology that can investigate such inconsistent censorship enforcement within a country on a global scale. For instance, Cho *et al.* [16] leveraged BGP churn to identify the Autonomous Systems (ASes) involved in censorship enforcement. However, it entirely relied on measurement data from ICLab [44] that uses VPNs only, which has limited coverage and may omit critical observations. Also, as the diversity of network-level paths induced by BGP churn is inherently limited and random, it cannot be experimentally controlled. Other recent studies [9], [10] investigated the impact of Equal-Cost Multi-Path (ECMP) routing, showing that censorship variation can stem from router load balancing (*e.g.*, based on packet attributes such as source port or lower bits of source IP). Yet such inconsistencies may occur only at a small scale, since ECMP policies are typically administered by the same AS and thus tend to exhibit similar behaviors.

To this end, we bridge this gap by conducting an in-depth investigation to scrutinize the inconsistency of censorship enforcement within a country. We design and deploy a measurement framework, named *Pathfinder*, to uncover inconsistent censorship behavior across different network paths. Pathfinder leverages multiple geo-distributed back-end control servers as probing destinations to induce routing variability. Specifically, by issuing probing packets from a single vantage point to multiple geo-distributed destinations, we can force the traffic to traverse different transit networks, enabling the detection of inconsistent censorship enforcement across various networks. Moreover, following the approach developed in Disguiser [33], we configure Pathfinder's con-

trol servers with a static payload that serves as the ground truth of server responses, enabling accurate identification of censorship activities without manual inspection.

Using Pathfinder, we conduct a large-scale measurement study to understand censorship inconsistency arising from diverse routing paths. Over a 60-week period, we collect data from more than 144K acquired vantage points across 120 countries with observed censorship activities. We reveal that such a phenomenon is highly prevalent, where 91.7% of countries (110 in total) exhibit varying levels of censorship inconsistency, though many of them are typically considered as having centralized censorship control. Notably, we observe that certain paths encounter significantly fewer censorship events than others, indicating promising potential for censorship circumvention. For instance, with the vantage points from India, probing packets routed toward control servers deployed in the Middle East experience much less censorship (8%) than those directed along other paths (40–60%).

Moreover, we uncover that different cloud platforms also contribute to censorship inconsistency, largely due to their varying peering connections or preferences, which results in probing packets to traverse distinct transit networks with different censorship policies. Also, direct peering between cloud platforms and eyeball networks within a country could allow certain censorship circumvention, as the packets would enter cloud providers' private infrastructure before encountering any upstream censorship devices. To illustrate an in-depth investigation, we conduct extensive case studies for two representative countries (South Korea and India). In addition, as our primary goal in this study is to analyze and understand the underlying causes of censorship inconsistency, we focus our experiments on HTTP-based censorship as a lens to examine the problem but leave other censorship forms for future work. The main contributions of this study can be summarized as follows:

- We develop Pathfinder, a framework to simultaneously explore multiple potential routing paths for identifying censorship activities when probing packets from one node are routed to different ISPs or transit networks.
- We conduct extensive measurements and uncover 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.
- We discover that large cloud platforms can affect the occurrence of censorship due to various peering connections. We leverage application-layer traceroute to perform case studies in two countries (South Korea and India), collecting detailed network paths and examining how different peering configurations lead to inconsistent censorship.

2. Background

2.1. Censorship Techniques

Internet censorship can be implemented using various techniques [25], such as IP-layer or application-layer censorship. IP-based censorship filters traffic based on the destination IP address; however, it can be easily evaded by changes of service IPs. Moreover, the widespread use of cloud services and CDNs has led to highly dynamic and shared IP resources, making IP-based blocking prone to collateral damage affecting legitimate services [22], [72]. In contrast, application-layer censorship offers more precise control over undesired content and has garnered increasing attention from both censorship regimes and the research community, which is the focus of this work.

Application-layer Censorship. Application-layer censorship involves the information inspection inside the application-layer protocols, *e.g.*, domain-name-based blocking in DNS and HTTP(S), and keyword-based filtering in packet streams.

Domain names explicitly indicate the online resources a user intends to access, enabling censorship devices to block content prohibited by authorities. When a user device resolves a domain name to the corresponding 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 reveals the domain a client is attempting to access. As the HTTP protocol is unencrypted, censors can know exactly the requested domain. While HTTPS encrypts HTTP packets and thereby conceals the Host header after the TLS handshake, the Server Name Indication (SNI) field in TLS 1.2 remains unencrypted and can still expose the target domain to censors.[1]

To exert finer control over undesired content, keyword-based censorship inspects unencrypted packet streams for predefined sensitive keywords and disrupts traffic upon detection. This involves intercepting and parsing HTTP traffic, while searching for these keywords in certain locations of HTTP requests or responses, such as request line, headers, or payload body [65].

Interference Techniques. The mechanisms for blocking undesired Internet content also vary. To perform DNS manipulation for denying a domain's access, censorship devices usually inject forged DNS responses that redirect user requests to a censor-controlled address (*e.g.*, that shows a *blockpage* indicating the domain is prohibited), a non-routable private IP address, or a public IP address that is itself subject to IP-layer blocking [27]. In cases involving TCP-based DNS, censors typically inject RST/FIN packets to tear down the TCP connection [33].

In HTTP(S) blocking, when a prohibited domain is detected (*i.e.*, from Host header in unencrypted HTTP messages or the plaintext Client Hello message in HTTPS handshake), the censor can simply drop the request, resulting in a client-side timeout. Alternatively, the censors could forcibly tear down the connection by injecting RST/FIN packets or return a dedicated blockage to inform the user of the domain restriction.

1. In TLS 1.3, encrypted SNI has been proposed to conceal the accessed domain [18]. Nevertheless, as censors still must enable these fundamental forms of censorship to enforce their policies, our study on censorship deployments remains relevant and unaffected by this development.

Inbound and Outbound Censorship. Censorship devices can operate on different traffic directions, *i.e.*, inbound and outbound censorship [62]. Inbound censorship inspects traffic entering the network from external sources, whereas outbound censorship targets traffic originating within the censoring regions and heading to external destinations. Outbound censorship is more commonly deployed, as censorship primarily aims to restrict how users within the censoring regions access sensitive or prohibited content.

2.2. Application Traceroute

To better understand censorship activities, application(-layer) traceroute has been explored to locate censorship devices and examine their behavior [33], [57]. This technique sends probing packets with incremented TTL values, embedding application-layer payloads designed to trigger censorship. Before reaching the censor, the probing packets expire at intermediate hops, prompting ICMP Time Exceeded messages. As the TTL increases and the packet reaches the censor, the signs of interference (*e.g.*, RST/FIN packets) indicate the exact hop at which the censorship occurs.

3. Pathfinder

In this section, we present our methodology and framework design to systematically investigate censorship inconsistency by measuring censorship activities across diverse network paths concurrently. In particular, we outline the challenges faced by existing measurement approaches and highlight key design considerations to minimize noise and ensure reliable results during large-scale experiments.

3.1. The Censorship Inconsistency Problem

Understanding censorship activities typically involves sending probing requests that could trigger censorship from vantage points within a country or region and detecting network interference by analyzing abnormal responses. However, this approach relies solely on observations from the client side, offering no visibility into the underlying network paths or the locations of censorship devices. As a result, censorship behaviors are often characterized at the country level. However, censorship policies are usually enforced at the ISP level, and different ISPs may implement these policies in diverse ways, making country-level characterization inadequate to unveil inconsistent censorship behaviors within a country.

Figure 1 illustrates how censorship inconsistency is largely overlooked in existing measurement approaches. To trigger censorship devices along the network path, probing packets, including traceroutes, are sent from the vantage points within a country. However, since probing packets all target the same destination, those issued from distributed vantage points often converge to a limited set of upstream transit networks, *e.g.*, AS 1 in Figure 1. Meanwhile, other networks such as AS 2, which may enforce different censorship policies for the same domain, remain unobserved due to routing configurations. As a result, such inconsistencies are missed by conventional measurements.

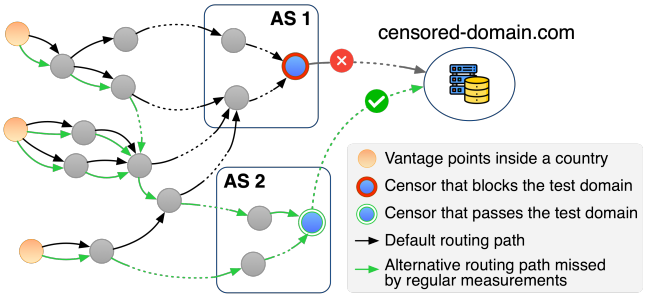

Figure 1: Inconsistent censorship in different paths due to different censorship implementations.

3.2. Threat Model

In this work, we consider two primary entities in the censorship ecosystem: censors and evaders. Censors are operated at either border routers of ISPs or international gateways for monitoring and filtering network traffic to enforce access restrictions, *e.g.*, through TCP connection reset or selective packet dropping. On the other hand, evaders attempt to bypass the censorship enforcement to access restricted content, using proxies such as VPN or obfuscation tools that disguise traffic to avoid detection.

3.3. System Design

To systematically examine censorship inconsistency, we design and deploy a measurement framework, *Pathfinder*, to detect censorship across diverse paths. The key idea is that by directing probing packets through different networks within a country, we can identify how varying routing paths influence censorship behaviors, gaining a more fine-grained understanding of censorship deployments.

Overview. Figure 2 depicts the system design of Pathfinder. At a higher level, Pathfinder comprises a set of globally distributed vantage points and multiple back-end control servers that serve as the destinations for probing packets. A client feeds a list of country-specific test domains, which is used by the vantage points to construct HTTP requests and schedule measurements. A test domain is embedded into the HTTP header to potentially trigger censorship, and the constructed requests are sent to different control servers to explore diverse routing paths. The control servers uniformly respond to all received requests with a static, non-sensitive payload. Pathfinder then collects response data and connection status for each probing test from both the client and control server sides for analysis.

Vantage Points. To issue probing requests and conduct our experiments, we need to acquire a set of vantage points (VPs) distributed globally. Similar to the prior study [33], we utilize SOCKS IP proxies to send probing requests. SOCKS IP services maintain large pools of hosts that voluntarily participate, enabling traffic to be routed through their infrastructure. There are different types of SOCKS IP proxies, such as residential proxies, ISP proxies, and datacenter proxies. We conducted extensive evaluations across these different services and platforms to validate

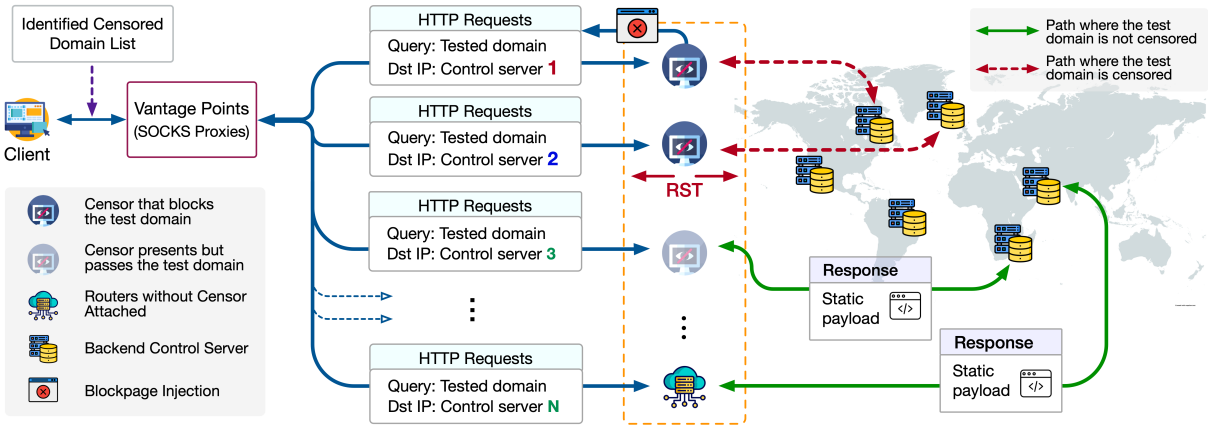

Figure 2: System design of Pathfinder and its exploitation of censorship inconsistency. A set of geographically distributed back-end control servers is deployed to induce path diversity by enforcing probing packets to traverse different network paths within a country.

their usability. We confirmed that residential networks often face more aggressive censorship compared to vantage points in commercial infrastructures like datacenters [44], [70]. Therefore, vantage points from residential IP proxies (RESIPs) can offer more comprehensive and representative coverage for examining censorship deployments. To this end, we subscribed to Proxyrack [51], a popular and stable RESIP service that has been well-studied and adopted in previous research [32], [33], [40]. Additionally, we discuss the ethical considerations of using RESIP and our mitigation strategies in detail in §4.3.

Censored Domain List. Since the primary goal of Pathfinder is to examine censorship inconsistency, we leverage the existing list of identified censored domains to streamline our measurements, focusing on issuing probing packets only for domains that have been confirmed to be censored in a given country. As such, we acquire the list from Disguiser [33], which compiles a set of popular and sensitive domains, sourced from Alexa's top 1K domains [3] and the Citizen Lab [17], and produces a country-specific list of censored domains [59] validated with Disguiser's measurements.

Probing Requests. Using each vantage point acquired through RESIP, Pathfinder constructs a series of HTTP requests and sends them to trigger potential censorship along the network paths. In this study, we focus specifically on HTTP-based censorship as it remains one of the most prevalent forms deployed across countries [44]. Each request is constructed such that the Host header contains a domain from the identified censored-domain list, while the destination is set to one of our back-end control servers. We set a 5-second timeout for each probing request to prevent excessive delays if a censorship device drops the requests. Our experiments show that if no response is received within 5 seconds, it is highly unlikely to receive the response with a longer wait, which is also aligned with prior studies [47], [48]. Also, Pathfinder automatically retries four more times for the timeout requests before declaring it to be blocked by censorship, considering that some requests may be dropped (*e.g.*, due to network congestion).

Control Server. Control servers are central to identifying censorship activities and enabling varying routing paths. Following the approach in [33], each control server provides a static payload for incoming HTTP requests. This payload is designed to be distinct from any legitimate content or blockpages, thereby serving as a ground truth for efficient censorship detection. Furthermore, Pathfinder deploys multiple control servers across geographically diverse locations, and thus the HTTP requests to different control servers will be routed through distinct network paths. We detail the control server setup in §4.

3.4. Special Design Considerations

Eliminating Cache Proxies. To accurately detect censorship, it is essential to account for scenarios where a cache proxy exists along the network path. In this case, a cache proxy may respond to a probing request with a cached result for the test domain, and the vantage point could receive a response rather than the static payload of control servers. To eliminate this issue, we perform a real-time cache proxy test for each vantage point, excluding those that are potentially affected by cache proxies along the path. Specifically, we configure two reference servers that host the same domain under our control but differ in destination IPs and landing page content. We then instruct the vantage points to sequentially probe both servers by fetching their landing pages. If a cache proxy is present, the second request may receive a cached response from the first server, revealing the proxy's interference.

While our real-time cache proxy test filters out most vantage points affected by caching, we find that a small subset of vantage points still receive responses likely from intermediate caches. This could occur because a cache selectively stores content, or it may intercept the connections by independently performing DNS resolution [11], resulting in obtaining an IP address different from that of our control servers. Therefore, we conduct an additional offline check. We fetch the legitimate landing pages for the test domains and simply compare their Title tags with the responses received by the vantage points. If a vantage point obtains a Title tag

from the legitimate page of a test domain, it implies that the page could be returned by an intermediate cache rather than our control server. In total, we exclude 164 acquired vantage points by offline check. To this end, by verifying the static payload and using the IP addresses of our control servers, we ensure that other factors, such as CDNs' caching or third-party DNS resolution, do not interfere with our experiments or affect the results.

Eliminating Inbound Censors. As discussed in §2.1, censorship can be enforced either by inbound or outbound censors. Since our focus is on censorship enabled by the countries where vantage points are located, inbound censorship at the control server side could also be mistakenly identified as censorship, leading to false cases. To mitigate this issue, we carefully select control server locations in countries with no widely identified censorship. Additionally, we further conduct a dedicated experiment to validate the absence of inbound censorship on the control server side. Specifically, we set up 50 VPN servers, each located in a different country, and send a series of probing packets to all our control servers. Each packet carries a test domain in the censored-domain list. We identify that *all* of the probing packets consistently retrieve the pre-defined static payload from control servers. This confirms that our control servers are indeed deployed in countries where no inbound censorship is enabled.

4. Experiments

To investigate censorship inconsistency in real-world scenarios, we conduct two experiments exploring the primary sources of path diversity that lead to such inconsistency: the *locations of destinations* and the *hosting platforms*. This section provides a detailed overview of our experiment design and data collection. We also discuss ethical considerations and outline our strategies to minimize the impacts.

4.1. Impact of IP Destinations

We aim to explore how varying destinations lead to diverse network paths, resulting in inconsistent censorship. Unlike prior research relying on paths induced by router load balancing [9], our experiment is designed to explicitly route probing requests from a single vantage point through multiple distinct network paths by targeting geographically distributed destination servers.

Vantage Points. As described in §3.3, we use RESIPs (provided by Proxyrack [51]) as our vantage points, emulating normal traffic of regular users. On the other hand, since the proxies are randomly assigned, we encounter an uneven distribution of proxies across countries. To rectify this, we implement a cap that limits the number of vantage points to a maximum of 80 per country per week, ensuring a more balanced distribution of vantage points. This increases the chances of observing censorship activities across a wider range of countries while maintaining a substantial amount of data collected from each country.

Control Servers. This experiment aims to investigate censorship activities across diverse paths. To achieve this, we carefully select control servers based on two key criteria. First, the servers must be geographically distributed to maximize the likelihood that probing requests traverse various network paths to reach these servers. Second, as the probing requests aim to trigger the potential censorship activities within the censoring regions where the vantage points are located, the servers must be located in places with no *inbound* censorship activities to eliminate potential interference (§3.4). Following these criteria, we establish six control servers[2] using AWS EC2 instances, located in Virginia (North America - East Coast), California (North America - West Coast), São Paulo (South America), London (Europe), Bahrain (Middle East), and Cape Town (Africa).

Data Collection. We conduct our measurement experiments over a 60-week period. In total, we obtain 144,418 vantage points in 120 countries where censorship behaviors have been observed in previous studies. The geographic distribution of these vantage points is illustrated in Figure 11 in Appendix A.

4.2. Impact of Hosting Platforms

In addition to IP destinations, we observe that the hosting platform is another key factor that leads to censorship inconsistency. This is because different peering policies can result in packets being routed through diverse networks with different censorship policies. Thus, we conduct extensive measurements by deploying control servers across different hosting platforms. As such, probing requests are routed differently to reach these platforms, which enables us to further assess how hosting platforms influence censorship inconsistency.

Vantage Points. We follow the same method described in §4.1 to obtain vantage points from the RESIP. Meanwhile, due to the limited lifespan of each assigned vantage point, we utilize a different set of vantage points for this experiment.

Control Servers. To thoroughly examine the impact of hosting platforms on censorship inconsistency, we deploy control servers across three major cloud platforms: Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure (Azure). To minimize the influence of geographic variation, we specifically select the locations where all three platforms have data centers deployed: Virginia (United States), Sydney (Australia), Paris (France), and São Paulo (Brazil). We establish one control server in each of these locations from each of the platforms, resulting in a total of 12 control servers.

Data Collection. Our data collection lasts for 36 weeks, during which we measure censorship activities from 66,123 vantage points across 120 countries. In Appendix A, Figure 12 shows the worldwide distribution of vantage points. Additionally, Figure 13 illustrates the distribution of the total number of vantage points collected in both §4.1 and §4.2.

2. Due to the limited geographic distribution of cloud data centers, we deploy our control servers in Paris and Bahrain to ensure broader coverage. We notice that while the vantage points in France and Bahrain may experience censorship, our experiments are unaffected because no inbound censorship is observed in their networks (see §3.4).

4.3. Ethical Considerations

Ethical concerns in censorship-related measurement studies have been extensively discussed [9], [10], [34], [50], [62], under the guidelines of the Belmont [43] and the Menlo Report [8]. When measuring censorship, it is critical to consider the experiment's ethical implications carefully. In our study, we utilize RESIPs from Proxyrack as vantage points to issue probing requests, similar to prior work [33]. Proxyrack is a commercial platform that recruits residential proxies worldwide, where participants voluntarily opt-in to join the network and perform tasks for financial gain [52].

Although our experiments align with their Terms of Service and the participants are informed about such usage, they may not fully understand the associated risks. To minimize potential risk to the participants of residential proxies, Pathfinder is designed to ensure that our experiments do not generate traffic to the actual servers associated with the testing domains, avoiding access to any actual undesired content. Also, we limit the usage frequency for each proxy to further reduce potential risk to proxy owners, *i.e.*, each proxy IP is used only once per week for our measurements. Furthermore, we provide a comprehensive description of our experiment, along with our contact information, in the static payload of our control servers. During our entire experiments, we did not receive any concerns regarding our measurements. Additionally, our measurements do not involve any human subjects or the collection of any personal data, and so are typically outside the scope of the institutional Internal Review Board (IRB) [34]. Nevertheless, we follow the standard practice of censorship research to obtain an official IRB exemption from our institute.

Furthermore, we highlight our adherence to the principle of *beneficence*, as emphasized in prior studies [9], [49], [50]. This principle aims to balance benefits and risk, seeking to minimize potential harm to participants to the greatest extent possible when complete elimination of risk is not feasible. Our study offers meaningful benefits by advancing the understanding of censorship practices. We reveal widespread censorship inconsistencies in many countries driven by changes in network paths, which could facilitate the development of more adaptive circumvention technologies.

5. Results & Analysis

In this section, we analyze the data collected from the experiments described in §4 to address a series of research questions related to the existence, causes, and exploitation of censorship inconsistencies. Specifically, the first two questions are based on the findings of IP-destination experiments in §4.1, while the latter two are based on the results of hosting platform experiments in §4.2.

- **RQ1**: In general, how prevalent is the censorship inconsistency across different countries and domains? (§5.1)
- **RQ2**: To what extent do the geographic locations of destination servers influence censorship inconsistency across different countries? (§5.2)

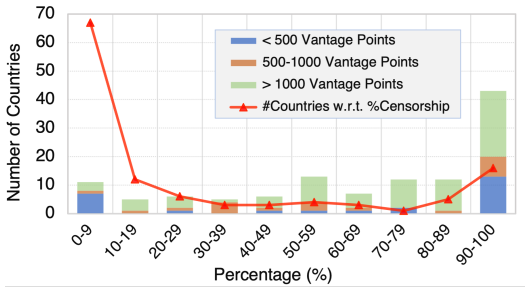

Figure 3: Distribution of the number of countries with respect to their censorship percentage (*line graph*) and inconsistency percentage (*stacked bar graph*).

- **RQ3**: To what extent do different hosting platforms contribute to censorship inconsistency? (§5.3)
- **RQ4**: Can specific paths with less censorship be leveraged to facilitate circumvention via routing detours? (§5.5)

5.1. Prevalence of Censorship Inconsistency

To answer RQ1, we examine the prevalence of censorship inconsistency in different countries by analyzing our experiment results from two perspectives: vantage-point-level inconsistency and domain-level inconsistency.

Vantage-Point-Level Inconsistency. Pathfinder sends requests from RESIP vantage points to our control servers via diverse network paths, enabling us to identify inconsistent censorship behaviors across these paths. In this analysis, we consider a vantage point as censored if any of its associated measurement requests are subject to censorship. Accordingly, the vantage-point-level censorship percentage (*VP-Censored*) is defined as the ratio of the number of censored vantage points to the total number of vantage points evaluated. More importantly, we define the vantage-point-level inconsistency percentage (*VP-Inconst.*) as the proportion of censored vantage points that exhibit inconsistent censorship behaviors across different network paths within a country. The censorship percentage implies the extent of censorship enforcement, while the inconsistency percentage indicates the prevalence of inconsistent censorship behaviors. Table 4 in Appendix A lists the inconsistency percentages for countries with more than 200 vantage points.

Figure 3 shows the distribution of countries based on their vantage-point-level censorship percentage (the red line). We observe that the majority of countries exhibit an "all-or-nothing" censorship pattern, consistent with prior results from Bhaskar *et al.* [9]. Specifically, 79 countries (66%) have a censorship percentage below 20%, while 21 countries (17.6%) exhibit levels above 80%. Only 20 countries (16.8%) have a vantage-points-level 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 3 also presents a bar graph illustrating the distribution of countries by their censorship inconsistency percentage. Our results show that only 11 countries (10%) have an inconsistency percentage below 9%, indicating

Country	Vantage Points			%VP-Censored	%VP-Inconst.
	Total	#Censored	#Inconst.		
Kazakhstan	1,825	1,825	894	100.00%	48.99%
Kuwait	1,338	1,336	1254	99.85%	93.86%
China	1,256	1,229	1,220	97.85%	99.27%
Pakistan	1,719	1,582	1,574	92.03%	99.49%
Russia	2,003	1,721	905	85.92%	52.59%
Bangladesh	1,786	1,432	1,156	80.18%	80.73%
Thailand	1,925	1,182	680	61.40%	57.53%
Vietnam	1,971	1,106	830	56.11%	75.05%
India	2,015	1,051	992	52.16%	94.39%
South Korea	2,596	1,300	424	50.08%	32.62%

TABLE 1: Top 10 countries (with >1K VPs) with the highest censorship percentage (*VP-Censored*) and their inconsistency percentage (*VP-Inconst.*).

that censorship inconsistency is prevalent. Furthermore, we observe severe censorship inconsistency in a substantial number of countries. Specifically, 20 countries (18%) have their inconsistency percentage fall into the range of 90%–99%, while 23 countries even exhibit 100% censorship inconsistency, where all vantage points within these countries observe various extents of inconsistent censorship among different network paths.

Additionally, the stacked bars in Figure 3 show the number of vantage points evaluated in our experiment. We collected a substantial volume of data from countries with more than 1,000 vantage points (green bars), allowing us to comprehensively explore censorship behaviors in these countries. Among them, 34 out of 74 (46%) countries exhibit censorship inconsistency exceeding 80%. On the other hand, 12 countries with fewer than 500 vantage points (blue bars) also show 100% inconsistency. These results highlight that our experiments effectively demonstrate the prevalence of censorship inconsistency, irrespective of the number of vantage points available per country.

We conduct further analysis on countries that are well represented (*i.e.*, over 1,000 vantage points). Table 1 lists the top 10 countries with the highest censorship percentage, all of which exceed 50%. Kazakhstan, Kuwait, China, and Pakistan have over 90% vantage points experiencing censorship. Meanwhile, the highlighted countries also exhibit significant censorship inconsistency, *e.g.*, China, Pakistan, and Kuwait also show inconsistency percentages above 92%. This indicates that, *even in countries with strict and pervasive censorship policies, inconsistency remains prevalent, suggesting the potential of bypassing the censorship by exploiting alternative routing paths.*

Domain-Level Inconsistency. In addition to the inconsistency encountered by vantage points, we here analyze it from the perspective of censored domains. A domain is considered censored if any request containing that domain is blocked by censors within a country. Then, we define the domain-level censorship percentage as the ratio of requests toward a given domain that are blocked to the total number of requests sent to that domain within a country during our experiments.

The results reveal that domain-level inconsistency varies, and is also prevalent across many countries. Figure 4 presents a min-average-max chart along with the percentiles of

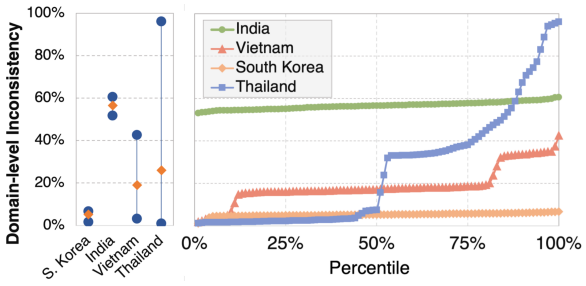

Figure 4: The min-average-max chart (*left bar-graph*) and the percentiles of censorship inconsistency (*right line-graph*) in South Korea, India, Vietnam, and Thailand.

domain-level inconsistency in South Korea, India, Vietnam, and Thailand. The height of each bar/curve indicates the level of censorship inconsistency in each country toward different domains. We can see that Thailand and Vietnam demonstrate higher levels of domain-level inconsistency, indicating that censored domains in these countries experience significantly different censorship behaviors. In particular, Thailand shows the most prevalent inconsistency at the domain level. For example, 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, suggesting the coexistence of both centralized and decentralized censorship. Similarly, Vietnam also exhibits high inconsistency, with a wide range of domain-level censorship percentages ranging from 3% to 43%. In contrast, India and South Korea exhibit relatively low domain-level inconsistency, suggesting that censorship is more uniformly applied across different blocked domains.

5.2. IP Destination Inconsistency

Here, we further break down the results presented in §5.1 to address RQ2. As described in §4.1, we utilize RESIP as vantage points and set up six geographically distributed control servers in distinct regions of Amazon AWS, enforcing probing requests to traverse diverse network paths depending on the server's location. This enables us to identify censorship activities along different paths to examine how the geographical location of IP destinations influences censorship inconsistency.

Accordingly, we define the censorship inconsistency caused by IP destinations for each country (referred to as *destination inconsistency*) as the percentage difference in censorship across different network paths. Figure 6 demonstrates the CDF of destination inconsistency in each country. The results show that over 50% of countries experience destination inconsistency greater than 75%. This confirms that censorship activities vary widely across network paths toward various control servers.

Figure 5 further illustrates the censorship percentages observed along paths toward each control server in three countries: India, Bangladesh, and Thailand. Specifically, we see that India exhibits the highest destination inconsistency among different paths. Only 8.4% of requests to the control

ASN (Country)		Censorship Percentage						Inconsist.
		Virginia	California	São Paulo	London	Bahrain	Cape Town	
AS 204457	(Turkey)	0.5%	81.5%	0.1%	28.0%	0.9%	0.9%	81.4%
AS 137526	(Bangladesh)	39.3%	0.0%	0.0%	78.1%	39.3%	39.3%	78.1%
AS 135987	(Vietnam)	1.2%	2.4%	76.8%	1.2%	2.4%	0.0%	76.8%
AS 57011	(Russia)	76.1%	0.0%	67.1%	67.1%	67.1%	67.1%	76.1%
AS 1312934	(Thailand)	69.4%	9.4%	1.7%	71.1%	71.1%	70.6%	69.4%
AS 132298	(Bangladesh)	67.5%	67.2%	66.7%	68.2%	0.0%	65.3%	68.2%
AS 124946	(India)	52.6%	52.8%	57.8%	64.9%	0.0%	51.9%	64.9%
AS 55492	(Bangladesh)	68.6%	15.5%	25.9%	77.3%	77.7%	77.7%	62.1%
AS 133227	(India)	52.4%	54.4%	54.5%	61.2%	0.0%	52.7%	61.2%
AS 199634	(Russia)	0.9%	59.8%	55.6%	36.8%	0.9%	0.4%	59.4%

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

server in Bahrain (Middle East) experience censorship, while censorship occurs between 37% to 60% for requests to other servers. Similarly, Bangladesh's vantage points show the probing requests toward Virginia and Bahrain with comparatively less censorship. On the other hand, in Thailand, requests sent to the server in Virginia experience significantly higher censorship than those sent to other destinations. These findings demonstrate that the location of destination servers significantly influences the network paths, leading to substantial censorship inconsistency.

AS-level Analysis. We extend our analysis from the country level to the AS level to gain more fine-grained insights into destination inconsistency. During our experiments, vantage points are acquired from a total of 3,913 ASes via the RESIP, with a diverse number of vantage points per AS ranging from one (*e.g.*, AS 3238 and AS 63023) to 1,186 (AS 4766). Figure 14 in Appendix A shows the distribution of the number of vantage points across different ASes. We filter out ASes with less than 80 vantage points in our analysis to focus on those well-represented ASes with sufficient vantage points.

To quantify destination inconsistency at the AS level, we measure the difference between the maximum and minimum censorship percentages encountered by probing packets sent from an AS to different control servers. Table 2 presents the top 10 ASes with the highest destination inconsistency between different network paths. Each of these ASes experiences at least 59.4% inconsistency, with AS 204457 showing the highest case at 81%. These ASes can be categorized into two groups. While AS 204457, AS 137526, and AS 135987 show few paths with a significantly higher censorship percentage than other paths, other ASes have one or two paths with notably low censorship activities. Importantly, we identify four network paths with no detected censorship, highlighted in Table 2. These AS-level observations reinforce our findings at the country level (Figure 5), confirming that the destination of the packets can significantly contribute to inconsistent censorship enforcement.

5.3. Hosting Platform Inconsistency

In addition to the destination location, our experiments reveal that different hosting platforms (*e.g.*, large cloud providers) also play a significant role in censorship inconsistency. This is because the network paths taken by requests can greatly vary due to different peering relationships between hosting platforms and ISPs.

In general, the network paths taken by probing requests are determined by the routing policies and peering relationships established between ASes from the source to the destination. By deploying control servers on different hosting platforms, the requests can 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 examine such cases, we set up control servers in the same geographical location but at different cloud platforms (AWS, GCP, and Azure) as the destinations of probing packets (§4.2).

Similar to §5.2, we define censorship inconsistency caused by different hosting platforms (*i.e.*, *hosting inconsistency*) as the percentage difference in observed censorship when probing packets are sent to various hosting platforms. Figure 7 illustrates the distribution of the number of countries based on their hosting inconsistency (shown as dark blue). The results reveal that 16 countries (13.3%) exhibit hosting inconsistency greater than 10%, indicating such inconsistencies are present but largely exist in a smaller subset of countries. Moreover, compared to the vantage-point-level inconsistency from §5.1 (*i.e.*, the bar graph in Figure 3, shown as light blue in Figure 7), there is no apparent correlation identified. This indicates that the inconsistencies identified here indeed stem from the impact of hosting platforms, rather than being coincidentally affected by the geographical location of control servers in different clouds.

Furthermore, we conduct additional analysis focusing on the most well-represented countries in our experiments. Table 3 shows the top 10 countries with at least 800 collected vantage points. We can see that the hosting inconsistency with respect to different platforms ranges from 9.09% to 66.72%. In particular, South Korea exhibits significantly lower censorship on GCP, with a censorship percentage of only 32.21% (highlighted in red in Table 3), compared to 98.93% and 76.64% for AWS and Azure, respectively. As highlighted in Table 3, India shows a similar pattern, with paths to GCP showing significantly lower censorship (18.57%) than the other platforms (50.89% for AWS and 37.95% for Azure). We further investigate the root causes be-

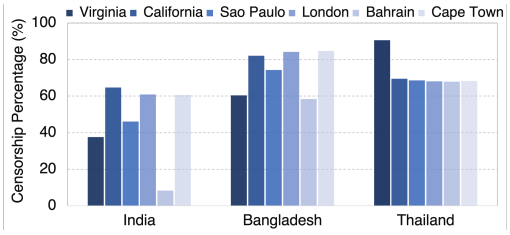

Figure 5: Censorship percentages to different control servers by vantage points in India, Bangladesh, and Thailand.

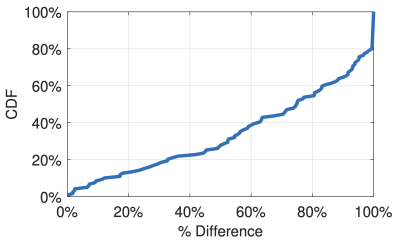

Figure 6: Distribution of censorship percentages for destination inconsistency across different countries.

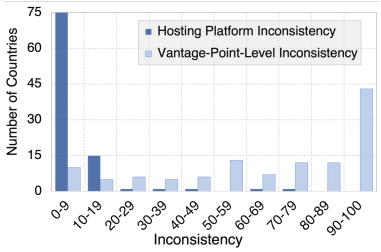

Figure 7: Distribution of censorship inconsistency in terms of different hosting platforms.

hind such inconsistencies using application-layer traceroute, shown as case studies detailed in §6.

Table 3 illustrates censorship inconsistency across different cloud platforms. To further break down these results, we examine the censorship percentages associated with each individual control server hosted on the respective platforms. Figure 8 presents the results of three representative countries, South Korea, Thailand, and India, comparing the censorship percentage observed on network paths toward each control server hosted on GCP, AWS, and Azure. The figure shows that censorship percentage varies significantly across hosting platforms, confirming the aggregated results at the hosting platform level in Table 3. Moreover, we can see that censorship percentages for all three countries are relatively stable with GCP and Azure, indicating that (1) the geographic location of control servers does not significantly influence censorship percentage in these two platforms and (2) GCP and Azure may likely maintain more consistent peering relationships with local networks in these countries. In contrast, in the case of South Korea, network paths to AWS show inconsistent censorship percentages among different locations, implying that AWS's routing paths and peering connections are diverse in South Korea. This also aligns with our earlier discussion on the impact of IP destinations on censorship inconsistency in §5.2.

5.4. Other Influencing Factors

Censorship devices actively examine all network traffic to detect and block access to undesired domains. When a domain changes its registration or hosting status, or it no longer hosts undesired content, censors may lift restrictions, allowing users to access it. However, if only a subset of censors update their filtering rules accordingly, requests examined by censors on different paths may experience inconsistent censorship.

During our experiments and data collection, we notice such behavior for a domain `onekorea.org` due to changes in its domain status from a regular webpage to a parked page. As part of the experiments described in §4.1, we evaluated censorship activities from 1,883 vantage points to our control server with `onekorea.org` as the domain name, resulting in a total of 22,596 domain requests. Out of these, 19,556 (87%) are blocked by censors, with 16,864 of them receiving a government blockpage while 2,692 experiencing

Country	# of Requests	% Censorship			Inconsist.
		AWS	GCP	Azure	
South Korea	763,875	98.93%	**32.21%**	76.64%	66.72%
Thailand	90,962	92.14%	85.39%	44.11%	48.04%
India	405,281	50.89%	**18.57%**	37.95%	32.32%
Russia	1,265,372	88.41%	77.62%	89.01%	11.39%
Venezuela	104,367	19.71%	22.15%	30.83%	11.12%
Jamaica	84,275	2.03%	1.62%	12.19%	10.57%
Poland	103,476	0.82%	0.82%	10.75%	9.93%
South Africa	91,285	3.19%	2.91%	12.76%	9.85%
Canada	128,472	7.04%	7.54%	16.48%	9.43%
Brazil	130,232	1.02%	0.91%	10.00%	9.09%

TABLE 3: Top 10 countries with more than 800 vantage points and their hosting platform inconsistency.

connection teardown or timeout. The rest 3,040 requests (13%) do not encounter any censorship and are able to successfully retrieve the expected static payload from our control servers. Furthermore, censorship inconsistency was observed by 1,686 vantage points when sending requests for `onekorea.org` to different control servers.

We suspect that the observed changes in censorship activities are associated with the change in the domain's hosting status. The fact that this domain continuously receives a government blockpage indicates that it is officially blocked by South Korean authorities, suggesting a country-wide censorship policy. In the meantime, our observation that censorship is lifted on only certain network paths implies that the censors are not uniformly deployed. This further reinforces our broader finding regarding the prevalence of censorship inconsistency across different network paths, even within the scope of a centralized policy.

5.5. Potential of Censorship Circumvention

Censorship inconsistency offers users a potential avenue to bypass censorship and gain access to censored content. As shown previously, both the location of destinations and the hosting platforms can significantly influence the routing paths of requests, implying that censorship could be potentially circumvented by strategically selecting outbound paths.

To circumvent censorship for specific domains, a straightforward but practical strategy involves deliberately routing requests to proxy servers in regions where less/no censorship occurs on network paths. For instance, as shown in Figure 5, vantage points in India experience fewer censorship activities

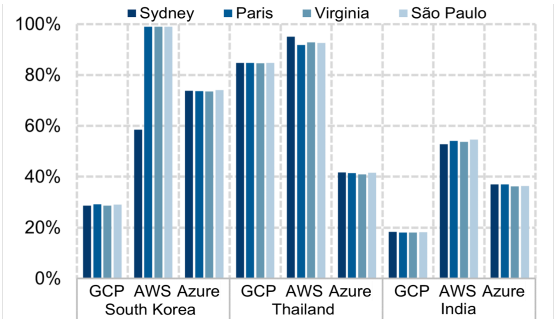

Figure 8: Censorship percentages from South Korea, Thailand, and India to control servers in GCP, AWS, and Azure.

when targeting the control server located in Bahrain (Middle East). Consequently, utilizing proxy servers in Bahrain may allow users in India to access content that would otherwise be blocked. Conversely, in the case of Thailand, requests toward the control server in Virginia experience more aggressive censorship. Thus, avoiding routing requests to such regions becomes a favorable choice for circumvention purpose.

In addition to destination location, censorship could also be circumvented by carefully selecting hosting platforms for proxy servers. As highlighted in Table 3, vantage points from both South Korea and India experience significantly lower censorship activities on the paths toward GCP. With that, it is preferred to deliberately select proxy servers hosted on GCP to evade censorship. Also, content providers could proactively host their services on platforms with less censorship interference, ensuring that users from a specific region have a greater likelihood to access their services without disruption.

Moreover, these strategies can be integrated into existing circumvention frameworks, to dynamically explore alternative paths as an additional vector for evasion. They could also be combined with other techniques, *e.g.*, CenFuzz [57], Geneva [13], and NetShuffle [38], to create a more efficient and resilient, yet still simple circumvention mechanisms.

6. Application Traceroute: Case Studies

As shown in §5.3, hosting platforms can significantly influence the severity of censorship experienced by users in certain countries. However, RESIPs cannot perform traceroute to pinpoint the exact location along the path where censorship occurs, because RESIPs typically operate above the transport layer, preventing control over the TTLs of the relayed packets. Therefore, in this section, we utilize commercial VPNs to conduct application traceroute, enabling a detailed investigation of censorship inconsistency for two representative cases in South Korea and India.

6.1. Methodology

As described in §2.2, application traceroute functions by sending a series of requests with incrementally increased TTLs and is accomplished when either a sign of censorship or our pre-defined static payload is received. By examining the corresponding response of each request, we can reconstruct the network path taken by probing requests and identify the specific hop where censorship policies are enforced. To collect such path information, we rely on commercial VPNs for conducting these application traceroute experiments.

The prior study [66] reveals that many VPNs may lie about their server locations. As the accuracy of vantage point locations is critical to our analysis, we validate whether the VPN servers are located where they are advertised. To do so, we attempt to access known censored domains from a VPN server to trigger censorship. Specifically, once we obtain a VPN node that claims to be located in the studied censoring countries (*i.e.*, South Korea or India), we issue probing requests to a small set of manually selected domains that are known to trigger censorship in our experiments in §5. We then confirm the location of the VPN server if we receive official blockpages issued by the respective authorities of censoring countries. After evaluating several different VPN services, we selected hide-my-ip [26]'s VPN as our platform, as its servers advertise relatively reliable location information in these two countries.

We use the same set of control servers described in §4.2 as destinations for application traceroute. These servers are hosted on various cloud providers (AWS, GCP, and Azure) and are geographically located in Sydney, Paris, Virginia, and São Paulo. For both studied countries (South Korea and India), we randomly select 32 domains from the censored domains list (§3.3) and initiate application traceroute carrying these domains from each VPN server to each control server. Additionally, we use IPinfo [30] to perform IP-to-AS mapping for the routers along the paths.

6.2. Case Study: South Korea

Leveraging application traceroute, we are able to identify network devices on the paths. Figure 9 shows a concrete example of the paths taken by probing requests sent from a vantage point in Seoul, South Korea. We observe that censorship consistently occurs on the paths to three control servers on AWS. Specifically, we receive an official blockpage via redirection (`http://warning.or.kr`), indicating that the accessed domain is prohibited. The remaining paths, including one to AWS in Sydney, do not encounter any censorship. This aligns with our general observations in Figure 8, where AWS shows more diverse censorship behaviors and the paths to AWS's Sydney 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 on the paths toward GCP and Azure. Although all probing packets, including those to AWS, are routed through the same ISP (AS 4766, Korea Telecom), the paths differ from certain hops, leading to inconsistent censorship. Interestingly, we observe that the upstream ISP (Korea Telecom) is directly peered with GCP's private networks within South Korea, resulting in the probing requests entering GCP's infrastructure before reaching any censorship devices. Table 6 in Appendix A provides detailed application traceroute results for a representative domain.

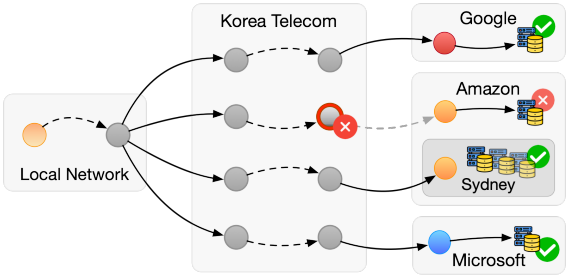

Figure 9: Application traceroute of different paths over different hosting platforms in Seoul, South Korea.

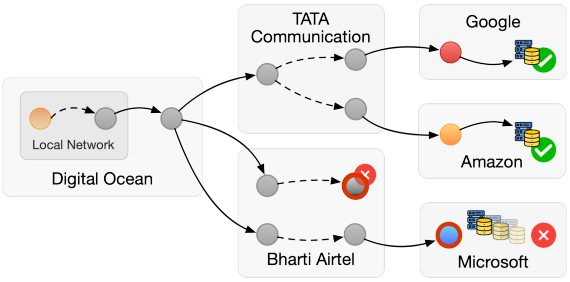

Figure 10: Application traceroute of different paths over different hosting platforms in Bangalore, India.

6.3. Case Study: India

Next, we present a case study of application traceroute in India to explore censorship inconsistency in depth, showing how hosting platforms influence censorship behavior. As shown in Figure 10, censorship consistently occurs on paths to control servers on Azure by receiving official blockpages, while no censorship is observed when retrieving static payloads from control servers hosted on AWS and GCP. This observation aligns with our earlier discussion in §5.3, highlighting that hosting platforms can significantly influence censorship behavior.

Table 5 in Appendix A illustrates a representative case of detailed application traceroute conducted in Bangalore, India. The results reveal that the packets sent to GCP and AWS are routed through AS 6465 and AS 4755, which are both associated with TATA Communications, and experience no censorship. Meanwhile, similar to the case in South Korea, we observe that the paths to the GCP-hosted servers show no intermediate hops before reaching the destinations, implying that probing packets are routed to GCP's private network by a direct peer between GCP and AS 4755.

For the censored paths to Azure, we identify specific censorship devices deployed in AS 9498 (Bharti Airtel) and AS 8057 (Microsoft Azure). While we consider that public cloud providers such as Microsoft Azure are unlikely to implement censorship, the traceroute results alone cannot confirm this. On the other hand, we speculate that it is more likely that censorship devices should still be located at AS 9498 (Bharti Airtel). These 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 has been detected and extensively examined in prior studies [33], [57].

In comparison, the application traceroute results in South Korea show all probing requests to different clouds traversing one upstream provider (AS 4766), while the requests issued from India are routed through different providers, with censorship observed only on the paths in one of the provider's networks (AS 9498). As shown in Table 3, both South Korea and India experience less censorship on paths toward GCP's control servers. This is due to direct peering between local ISPs and GCP's private networks, resulting in no censorship on paths toward GCP's control servers.

7. Limitations & Discussions

Censorship Close to Vantage Points. Pathfinder is designed to explore diverse paths by varying probing packets' destinations, effectively identifying inconsistent censorship implementations across various upstream networks. However, in such cases, vantage points would observe consistent censorship behavior because all probing packets are routed by the same local paths with censors deployed.

Vantage Point Impact. In this study, we utilize RESIPs as vantage points to conduct our experiments. While many countries are extensively studied with a large number of vantage points, other countries, such as Switzerland, Sudan, and Zambia, have limited available nodes (*e.g.*, less than 200). This may introduce certain bias, while it is inherent to all measurement studies. Despite the limited number of vantage points in some countries, our large-scale, longitudinal measurements still capture and demonstrate the prevalence of this issue. Also, to examine the censorship inconsistency in more detail, we conduct case studies through VPNs rather than RESIPs, because RESIPs typically do not support changing packets' TTLs for application traceroute. However, although VPNs are also widely used in prior studies [44], [57], [55], VPN servers are commonly hosted in commercial data centers, where the traffic may encounter less censorship than that in residential networks.

Test Domain List. Our experiments leverage an existing list of censored domains collected by Disguiser [33] to reduce measurement efforts. While the domains on this list have been validated with their censorship status, the prior study does not consider the impact of inconsistent censorship, so some censored domains may be excluded if probing requests encountered no censorship on specific network paths. Nevertheless, this does not significantly impact our study, as even an incomplete list is still sufficient to explore diverse network paths and illustrate varied censorship deployments.

System Adaptiveness & Extensiveness. In this work, we primarily focus on examining HTTP-based censorship as a *lens* to explore censorship inconsistency. We anticipate that the inconsistency phenomenon observed through other protocols, *e.g.*, DNS, HTTPS, and QUIC, would be similar and our conclusion should remain intact. Even with the adoption of evolving and sophisticated techniques, censors must still enable censorship enforcement of the most fundamental protocols, which ensures that our study on

censorship deployments remains relevant and applicable. On the other hand, as the design of Pathfinder aims to explore diverse network paths, it can also be integrated with novel detection schemes while identifying inconsistencies caused by underlying paths.

8. Related Work

Global Censorship Measurements. OONI (Open Observatory of Network Interference) [23] has established a community-driven global measurement framework by recruiting participants to run pre-defined tests to investigate censorship activities. VanderSloot *et al.* [62] proposed Quack, a remote measurement system that explores application-layer interference by using the Echo protocol. FilterMap [56] then enhances Quack to identify the content filtering techniques by analyzing their blockpages. Niaki *et al.* [44] developed ICLab, a platform that employs VPNs to launch a variety of longitudinal censorship measurements and proposed techniques to identify unknown blockpages. Censored Planet [55] integrates multiple existing techniques/frameworks, enabling synchronized censorship measurements to enhance data representativeness and coverage. In addition, Pearce *et al.* [50] introduced Iris, a system designed to identify and characterize DNS censorship on a global scale. Then, BreadCrumb *et al.* [9] is designed to further explore DNS censorship variation by manipulating the source parameters of probe packets and router-based load balancing. Lastly, Jin *et al.* [33] presented Disguiser, a ground truth-based framework that detects censorship activities with minimized manual inspection and reveals the censor deployment.

Our study complements these existing efforts by leveraging Pathfinder to systematically investigate censorship activities across various network paths on a global scale, revealing the wide presence of censorship inconsistency.

Country-Specific Censorship Studies. While significant research has examined Internet censorship on a global scale, many studies have also focused on censorship behaviors in specific countries. As one of the most extensive censorship systems in the world, the Great Firewall of China (GFW) has been extensively studied [4], [5], [6], [19], examining its policies of border ASes [70], DNS filtering behaviors [27], or capacities for censoring encrypted traffic [67]. Beyond GFW, other studies have examined censorship systems deployed in Iran [7], [12], Pakistan [37], [41], Syria [15], India [71], Kazakhstan [54], Russia [58], Turkmenistan [48], *etc.*

Although decentralized censorship has been observed in country-specific studies, such findings typically rely on collaboration with activists on the ground, and the used data/methods cannot be extended to other countries. Instead, Pathfinder is designed to systematically examine censorship inconsistency at scale by exploring diverse network paths.

Censorship Circumvention. With the ever-increasing censorship activities on the Internet, a wide range of circumvention techniques have been developed and examined [60]. Fifield *et al.* [22] proposed Domain Fronting, which conceals the true target domains by using different domain names in different layers of an HTTPS connection. This technique has been further explored by several other circumvention systems such as Lantern [39] and Psiphon [53]. Domain Shadowing [64] exploits the fact that CDNs allow their customers to bind arbitrary back-end domains, which can be configured as blocked domains to evade censorship. Burnett *et al.* [14] developed Collage that enables covert messages embedded within user-generated content. Khattak *et al.* [36] presented an analysis framework for identifying evasion vulnerabilities in Network Intrusion Detection System (NIDS). Nisar *et al.* [45] proposed C-Saw, a circumvention system that integrates censorship measurements with circumvention techniques into a single system. Bock *et al.* [13] proposed Geneva, a genetic algorithm that automates the discovery of censorship circumvention strategies against on-path network censors. Wang *et al.* [63] exploited the discrepancies in TCP state machines of deep packet inspection (DPI) implementation to bypass censorship. Raman *et al.* [57] 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 [28], [42], [72], Decoy routing [29], [35], Flash proxy [21], Infranet [20], Telex [69], uProxy [61], Alkasir [2], LASTor [1], Astoria [46], Autosonda [31], *etc.*

In this study, we reveal censorship inconsistencies, which could also be leveraged as a complementary component to be integrated with existing censorship circumvention (§5.5).

9. Conclusion

Internet censorship refers to the control or suppression of online content that users can access. In this study, we design and implement Pathfinder, a framework to systematically examine inconsistent censorship activities across different network paths within a country. Our findings reveal that censorship inconsistency is prevalent, where more than 90% of examined countries with censorship enforcement exhibit varying levels of censorship inconsistency when altering network paths. We further demonstrate that factors such as the geolocation and hosting platforms of destination servers often influence the routing of requests, resulting in inconsistent censorship. Notably, we identify that certain paths could exhibit significantly lower or even zero censorship, indicating that such inconsistencies could be exploited for censorship circumvention. To gain deeper insights, we conduct detailed case studies in two representative countries, South Korea and India, by leveraging application-layer traceroute to identify the exact nodes along network paths where such inconsistency occurs. The code and data used in this study are publicly available at https://github.com/e2ecensor/Pathfinder.

Acknowledgments

We would like to thank our shepherd and the anonymous reviewers for their detailed and insightful feedback, which helps to improve the quality of this paper. This work was supported in part by an Internet Freedom Fund from Open Technology Fund (OTF) and the National Science Foundation (NSF) under Grant Number CNS-2317829.

References

[1] M. Akhoondi, C. Yu, and H. V. Madhyastha, "LASTor: A low-latency AS-aware Tor Client," in *IEEE S&P*, 2012.

[2] W. Al-Saqaf, "Internet Censorship Circumvention Tools: Escaping the Control of the Syrian Regime," *Media and Communication*, 2016.

[3] Alexa, https://www.alexa.com/topsites.

[4] Anonymous, "The Collateral Damage of Internet Censorship by DNS Injection," *ACM SIGCOMM Computer Communication Review*, 2012.

[5] ——, "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship," in *USENIX FOCI*, 2014.

[6] Anonymous, A. A. Niaki, N. P. Hoang, P. Gill, and A. Houmansadr, "Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior," in *USENIX FOCI*, 2020.

[7] S. Aryan, H. Aryan, and J. A. Halderman, "Internet Censorship in Iran: A First Look," in *USENIX FOCI*, 2013.

[8] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan, "The Menlo Report," *IEEE Security & Privacy*, 2012.

[9] A. Bhaskar and P. Pearce, "Many Roads Lead To Rome: How Packet Headers Influence DNS Censorship Measurement," in *USENIX Security*, 2022.

[10] ——, "Understanding Routing-Induced Censorship Changes Globally," in *ACM CCS*, 2024.

[11] R. Bian, L. Jin, S. Hao, H. Wang, and C. Cotton, "Silent Observers Make a Difference: A Large-scale Analysis of Transparent Proxies on the Internet," in *IEEE INFOCOM*, 2024.

[12] K. Bock, Y. Fax, K. Reese, J. Singh, and D. Levin, "Detecting and Evading Censorship-in-Depth: A Case Study of Iran's Protocol Whitelister," in *USENIX FOCI*, 2020.

[13] K. Bock, G. Hughey, X. Qiang, and D. Levin, "Geneva: Evolving Censorship Evasion Strategies," in *ACM CCS*, 2019.

[14] S. Burnett, N. Feamster, and S. Vempala, "Chipping Away at Censorship Firewalls with User-Generated Content," in *USENIX Security*, 2010.

[15] A. Chaabane, T. Chen, M. Cunche, E. De Cristofaro, A. Friedman, and M. A. Kaafar, "Censorship in the Wild: Analyzing Internet Filtering in Syria," in *ACM IMC*, 2014.

[16] S. Cho, R. Nithyanand, A. Razaghpanah, and P. Gill, "A Churn for the Better: Localizing Censorship Using Network-level Path Churn and Network Tomography," in *ACM CoNEXT*, 2017.

[17] Citizen Lab, "URL Testing Lists Intended for Discovering Website Censorship," https://github.com/citizenlab/test-lists/, 2019.

[18] Cloudflare, "Encrypt it or lose it: how encrypted SNI works," https://blog.cloudflare.com/encrypted-sni/, 2018.

[19] R. Ensaf, P. Winter, A. Mueen, and J. R. Crandall, "Analyzing the Great Firewall of China Over Space and Time," in *PETS*, 2015.

[20] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. R. Karger, "Infranet: Circumventing Web Censorship and Surveillance," in *USENIX Security*, 2002.

[21] D. Fifield, N. Hardison, J. Ellithorpe, E. Stark, D. Boneh, R. Dingledine, and P. Porras, "Evading Censorship with Browser-Based Proxies," in *PETS*, 2012.

[22] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant Communication through Domain Fronting," in *PETS*, 2015.

[23] A. Filastò and J. Appelbaum, "OONI: Open Observatory of Network Interference," in *USENIX FOCI*, 2012.

[24] G. Gebhart and T. Kohno, "Internet Censorship in Thailand: User Practices and Potential Threats," in *IEEE EuroS&P*, 2017.

[25] J. L. H. Hall, D. A. Michael, A. Andersdotter, B. Jones, N. Feamster, and M. Knodel, "A Survey of Worldwide Censorship Techniques," https://datatracker.ietf.org/doc/draft-irtf-pearg-censorship/.

[26] HideMyIP, https://www.hide-my-ip.com/.

[27] N. P. Hoang, A. A. Niaki, J. Dalek, J. Knockel, P. Lin, B. Marczak, M. Crete-Nishihata, P. Gill, and M. Polychronakis, "How Great is the Great Firewall? Measuring China's DNS Censorship," in *USENIX Security*, 2021.

[28] J. Holowczak and A. Houmansadr, "CacheBrowser: Bypassing Chinese Censorship Without Proxies Using Cached Content," in *ACM CCS*, 2015.

[29] A. Houmansadr, G. T. K. Nguyen, M. Caesar, and N. Borisov, "Cirripede: Circumvention Infrastructure using Router Redirection with Plausible Deniability," in *ACM CCS*, 2011.

[30] IPinfo.io, http://ipinfo.io/.

[31] J. Jermyn and N. Weaver, "Autosonda: Discovering rules and triggers of censorship devices," in *USENIX FOCI*, 2017.

[32] L. Jin, S. Hao, H. Wang, and C. Cotton, "Understanding the Impact of Encrypted DNS on Internet Censorship," in *WWW*, 2021.

[33] ——, "Understanding the Practices of Global Censorship through Accurate, End-to-End Measurements," *ACM SIGMETRICS*, 2022.

[34] B. Jones, R. Ensafi, N. Feamster, V. Paxson, and N. Weaver, "Ethical Concerns for Censorship Measurement," in *ACM SIGCOMM Workshop on Ethics in Networked Systems Research (NS Ethics)*, 2015.

[35] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer, "Decoy Routing: Toward Unblockable Internet Communication," in *USENIX FOCI*, 2011.

[36] S. Khattak, M. Javed, P. D. Anderson, and V. Paxson, "Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion," in *USENIX FOCI*, 2013.

[37] S. Khattak, M. Javed, S. A. Khayam, Z. A. Uzmi, and V. Paxson, "A Look at the Consequences of Internet Censorship Through an ISP Lens," in *ACM IMC*, 2014.

[38] P. T. J. Kon, A. Gattani, D. Saharia, T. Cao, D. Barradas, A. Chen, M. Sherr, and B. E. Ujcich, "NetShuffle: Circumventing Censorship with Shuffle Proxies at the Edge," in *IEEE S&P*, 2024.

[39] Lantern, https://lantern.io/.

[40] X. Mi, X. Feng, X. Liao, B. Liu, X. Wang, F. Qian, Z. Li, S. Alrwais, L. Sun, and Y. Liu, "Resident Evil: Understanding Residential IP Proxy as a Dark Service," in *IEEE S&P*, 2019.

[41] Z. Nabi, "The Anatomy of Web Censorship in Pakistan," in *USENIX FOCI*, 2013.

[42] M. Nasr, H. Zolfaghari, A. Houmansadr, and A. Ghafari, "MassBrowser: Unblocking the Censored Web for the Masses, by the Masses," in *NDSS*, 2020.

[43] National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research, "The Belmont Report: Ethical Principles and Guidelines for the Protection of Human Subjects of Research," 1979.

[44] A. A. Niaki, S. Cho, Z. Weinberg, N. P. Hoang, A. Razaghpanah, N. Christin, and P. Gill, "ICLab: A Global, Longitudinal Internet Censorship Measurement Platform," in *IEEE S&P*, 2020.

[45] A. Nisar, A. Kashaf, I. A. Qazi, and Z. A. Uzmi, "Incentivizing Censorship Measurements via Circumvention," in *ACM SIGCOMM*, 2018.

[46] R. Nithyanand, O. Starov, A. Zair, P. Gill, and M. Schapira, "Measuring and Mitigating AS-level Adversaries Against Tor," *arXiv, arXiv:1505.05173*, 2015.

[47] S. Nourin, E. Rye, K. Bock, N. P. Hoang, and D. Levin, "Is Nobody There? Good! Globally Measuring Connection Tampering without Responsive Endhosts," in *IEEE S&P*, 2025.

[48] S. Nourin, V. Tran, X. Jiang, K. Bock, N. Feamster, N. P. Hoang, and D. Levin, "Measuring and Evading Turkmenistan's Internet Censorship: A Case Study in Large-scale Measurements of a Low-penetration Country," in *WWW*, 2023.

[49] P. Pearce, R. Ensafi, F. Li, N. Feamster, and V. Paxson, "Augur: Internet-Wide Detection of Connectivity Disruptions," in *IEEE S&P*, 2017.

[50] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson, "Global Measurement of DNS Manipulation," in *USENIX Security*, 2017.

[51] Proxyrack, https://www.proxyrack.com/.

[52] ——, "Become a Peer," https://www.proxyrack.com/become-a-peer/.

[53] Psiphon, https://www.psiphon.ca/.

[54] R. S. Raman, L. Evdokimov, E. Wurstrow, J. A. Halderman, and R. Ensaf, "Investigating Large Scale HTTPS Interception in Kazakhstan," in *ACM IMC*, 2020.

[55] R. S. Raman, P. Shenoy, K. Kohls, and R. Ensaf, "Censored Planet: An Internet-wide, Longitudinal Censorship Observatory," in *ACM CCS*, 2020.

[56] R. S. Raman, A. Stoll, J. Dalek, A. Sarabi, R. Ramesh, W. Scott, and R. Ensafi, "Measuring the Deployment of Network Censorship Filters at Global Scale," in *NDSS*, 2020.

[57] R. S. Raman, M. Wang, J. Dalek, J. Mayer, and R. Ensafi, "Network Measurement Methods for Locating and Examining Censorship Devices," in *ACM CoNEXT*, 2022.

[58] R. Ramesh, R. S. Raman, M. Bernhard, V. Ongkowijaya, L. Evdokimov, A. Edmundson, S. Sprecher, M. Ikram, and R. Ensafi, "Decentralized Control: A Case Study of Russia," in *NDSS*, 2020.

[59] D. E. to-End Framework for Measuring Censorship with Ground Truth, github.com/e2ecensor/Disguiser_public.

[60] M. C. Tschantz, S. Afroz, Anonymous, and V. Paxson, "SoK: Towards Grounding Censorship Circumvention in Empiricism," in *IEEE S&P*, 2016.

[61] uProxy, https://www.uproxy.org/.

[62] B. VanderSloot, A. McDonald, W. Scott, J. A. Halderman, and R. Ensafi, "Quack: Scalable Remote Measurement of Application-Layer Censorship," in *USENIX Security*, 2018.

[63] Z. Wang, S. Zhu, Y. Cao, Z. Qian, C. Song, S. V. Krishnamurthy, K. S. Chan, and T. D. Braun, "SymTCP: Eluding Stateful Deep Packet Inspection with Automated Discrepancy Discovery," in *NDSS*, 2020.

[64] M. Wei, "Domain Shadowing: Leveraging Content Delivery Networks for Robust Blocking-Resistant Communications," in *USENIX Security*, 2021.

[65] Z. Weinberg, D. Barradas, and N. Christin, "Chinese Wall or Swiss Cheese? Keyword Filtering In The Great Firewall Of China," in *WWW*, 2021.

[66] Z. Weinberg, S. Cho, N. Christin, V. Sekar, and P. Gill, "How to Catch When Proxies Lie: Verifying the Physical Locations of Network Proxies with Active Geolocation," in *ACM IMC*, 2018.

[67] M. Wu, J. Sippe, D. Sivakumar, J. Burg, P. Anderson, X. Wang, K. Bock, A. Houmansadr, D. Levin, and E. Wustrow, "How the Great Firewall of China Detects and Blocks Fully Encrypted Traffic," in *USENIX Security*, 2023.

[68] M. Wu, A. Zohaib, Z. Durumeric, A. Houmansadr, and E. Wustrow, "A Wall Behind A Wall: Emerging Regional Censorship in China," in *IEEE S&P*, 2025.

[69] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, "Telex: Anticensorship in the Network Infrastructure," in *USENIX Security*, 2011.

[70] X. Xu, Z. M. Mao, and J. A. Halderman, "Internet Censorship in China: Where Does the Filtering Occur?" in *PAM*, 2011.

[71] T. K. Yadav, A. Sinha, D. Gosain, P. K. Sharma, and S. Chakravarty, "Where The Light Gets In: Analyzing Web Censorship Mechanisms in India," in *ACM IMC*, 2018.

[72] H. Zolfaghari and A. Houmansadr, "Practical Censorship Evasion Leveraging Content Delivery Networks," in *ACM CCS*, 2016.

Country/Region	Percentage	Count
Albania, Angola, Bolivia, Bosnia Herzegovina, Brazil, Gambia, Honduras, Jamaica, Madagascar, Malawi, Mauritius, Myanmar, New Caledonia, Oman, Nepal, Puerto Rico, Somalia, Tajikistan, Uruguay	100%	19
Algeria, Bahrain, China, Estonia, Germany, Greece, India, Indonesia, Japan, Kenya, Kuwait, Libya, Mexico, Netherlands, Pakistan, Portugal, Serbia, Sweden	90% – 99%	18
Argentina, Australia, Bangladesh, Belgium, Czechia, Georgia, Italy, Norway, Spain, Suriname, United Kingdom, Venezuela	80% – 89%	12
Belarus, Chile, France, Hong Kong, Laos, Lebanon, New Zealand, Philippines, Singapore, South Africa, Uganda, Vietnam	70% – 79%	12
Canada, Hungary, Israel, Lithuania, Macao, Palestine (including Palestinian Territory)	60% – 69%	6
Benin, Colombia, Ecuador, El Salvador, French Polynesia, Iran, Latvia, Nigeria, Panama, Taiwan, Thailand, Ukraine	50% – 59%	12
Afghanistan, Egypt, Kazakhstan, Morocco, Nicaragua, Turkey	40% – 49%	6
Curacao, Dominican Republic, Finland, Jordan, Malaysia, Moldova, Qatar, South Korea, Sri Lanka, United Arab Emirates, Yemen	20% – 39%	11
Azerbaijan, Slovenia, Bulgaria, Uzbekistan, Saudi Arabia, Poland, Cuba, Peru, Fiji, Brunei, Equatorial Guinea, Ivory Coast	0% – 19%	12

TABLE 4: Countries/Regions ($>$ 200 VPs) with observed censorship inconsistency. The *Percentage* indicates the fraction of VPs experiencing inconsistent censorship behaviors.

Appendix

A. Distribution of Vantage Points

Figure 11 shows the distribution of the vantage points in the experiment on locations of destinations (§4.1). Figure 12 presents the distribution of the vantage points in the experiment on hosting platforms (§4.2). Figure 14 shows the distribution of the number of vantage points across different ASes. Additionally, Figure 13 plots the CDF distribution of the number of vantage points collected in §4.1 and §4.2.

B. Censorship Inconsistency Percentage

Table 4 lists the countries/regions where the vantage points observe inconsistent censorship (with the numbers of vantage points greater than 200).

C. Detailed Application Traceroutes

Table 5 presents the application traceroute from one vantage point located in Bangalore, India, to the control servers for the censored domain `cckerala.com`. Table 6 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`. 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.

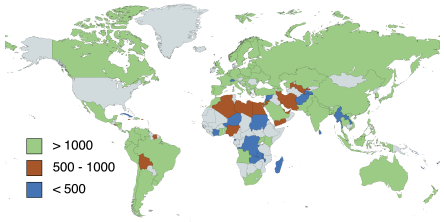

Figure 11: Distribution of vantage points across all countries in the experiments of IP destinations (§4.1).

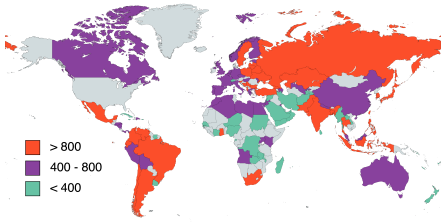

Figure 12: Distribution of vantage points across all countries in the experiments of hosting platforms (§4.2).

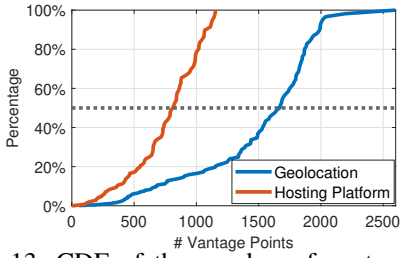

Figure 13: CDF of the number of vantage points in §4.1 and §4.2.

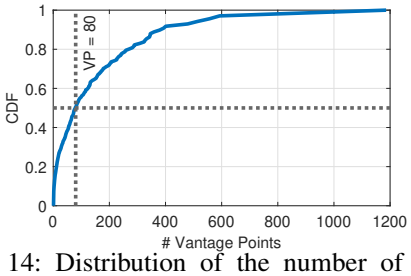

Figure 14: Distribution of the number of collected vantage points in different ASes.

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 219.65.110.189	138.197.249.18 DigitalOcean 219.65.110.185	138.197.249.14 DigitalOcean 219.65.110.189	138.197.249.0 DigitalOcean 219.65.110.185	138.197.249.0 DigitalOcean 219.65.110.185	138.197.249.18 DigitalOcean 219.65.110.189	138.197.249.22 DigitalOcean 219.65.110.189	138.197.249.0 DigitalOcean 219.65.110.185	138.197.249.0 DigitalOcean 202.56.198.57	138.197.249.22 DigitalOcean 202.56.198.29	138.197.249.22 DigitalOcean 202.56.198.29	138.197.249.22 DigitalOcean 202.56.198.29
ttl = 5	AS4755 TATA Comm.	AS4755 TATA Comm.	AS4755 TATA Comm.	AS4755 TATA Comm.	AS4755 TATA Comm.	AS4755 TATA Comm.	AS4755 TATA Comm.	AS4755 TATA Comm.	AS9498 Bharti Airtel	AS9498 Bharti Airtel	AS9498 Bharti Airtel	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 5: Application traceroute results for `cckerala.com` in Bangalore, India.

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	*	*	*	*	*	*	*	*	*	*	*	*
ttl = 8	*	112.174.90.110 AS4766 Korea Telecom	112.174.90.226 AS4766 Korea Telecom	112.174.90.154 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 6: Application traceroute results with `torrentdada.com` in Seoul, South Korea.