How India Censors the Web

Kushagra Singh*
Centre for Internet and Society
kushagra14056@iiitd.ac.in

Gurshabad Grover*
Centre for Internet and Society
gurshabad@cis-india.org

Varun Bansal
Centre for Internet and Society
varun13168@iiitd.ac.in

ABSTRACT

One of the primary ways in which India engages in online censorship is by ordering Internet Service Providers (ISPs) operating in its jurisdiction to block access to certain websites for its users. This paper reports the different techniques Indian ISPs are using to censor websites, and investigates whether website blocklists are consistent across ISPs. We propose a suite of tests that prove more robust than previous work in detecting DNS and HTTP based censorship. Our tests also discern the use of SNI inspection for blocking websites, which is previously undocumented in the Indian context. Using information from court orders, user reports and government orders, we compile the largest known list of potentially blocked websites in India. We pass this list to our tests and run them from connections of six different ISPs, which together serve more than 98% of Internet users in India. Our findings not only confirm that ISPs are using different techniques to block websites, but also demonstrate that different ISPs are not blocking the same websites.

KEYWORDS

Internet Censorship Analysis, Internet Service Providers, India

ACM Reference Format:
Kushagra Singh, Gurshabad Grover, and Varun Bansal. 2020. How India Censors the Web. In *12th ACM Conference on Web Science (WebSci '20), July 6–10, 2020, Southampton, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3394231.3397891

1 INTRODUCTION

Nation states around the world engage in web censorship using a variety of legal and technical methods [10, 34, 44, 47]. India is no different in this regard: the Government of India can legally order internet service providers (ISPs) operating in its jurisdiction to block access to certain websites for its users. This makes the situation different from jurisdictions like Iran and China, where internet censorship is largely centralised [10, 46].

Legal provisions in India, namely Section 69A and Section 79 of the Information Technology (IT) Act, allow the Central Government and the various courts in the country to issue website-blocking orders that ISPs are legally bound to comply with [1, 2]. The implementation of these provisions create various uncertainties in how internet users experience web censorship.

First, the regulations do not mandate ISPs to use specific filtering mechanisms. Thus, ISPs are are at liberty to employ various technical methods [47].

Second, website-blocking orders, especially those issued by the Government, are rarely available in the public domain. ISPs are, in fact, mandated by regulations to maintain confidentiality of certain website-blocking orders issued by the Government [3].Various attempts by researchers and advocacy organisations to obtain the complete list of blocked websites have failed [4, 24].

Third, the whimsy of ISPs and the Government aggravates these problems. Despite strict net neutrality regulations in India that prohibit ISPs from arbitrarily restricting access to websites [35], some ISPs may be doing so nonetheless [21]. Reports also suggest that the Government has issued blocking orders and then rescinded them on the same day [40].

These concerns motivated us to study web censorship in India in detail. In particular, we seek to answer two questions pertaining to how internet users in India experience web censorship: (i) what are the technical methods of censorship used by ISPs in India? (ii) are all ISPs blocking the same websites?

We contribute to research in documenting web censorship in India in three distinct ways:

Coverage of censorship mechanisms. Previous work has documented the use of DNS and HTTP based censorship techniques by Indian ISPs [47]. We include tests to determine whether ISPs are blocking websites based on the Server Name Indication, a Transport Layer Security extension. We find that Jio (an ISP serving 49.7% of internet users in India [36]) employs this technique. Additionally, we identify certain ISPs employing multiple censorship schemes, a fact previously undocumented in India.

Inconsistencies in websites being blocked. Although previous work records some inconsistencies in website blocklists across ISPs [47], they use a relatively smaller corpus of potentially blocked websites (1200). We curate the largest-known list of potentially blocked websites in India [1] (4379), which has allowed us to draw extensive conclusions about how experiences of web censorship vary across ISPs. We also report cases of an ISP blocking websites that are not blocked by any other ISPs.

Accuracy of censorship detection techniques. Yadav et al. [47] point out some drawbacks of relying on OONI [19] for measuring internet censorship in India, and propose new methods of detecting DNS and HTTP based censorship. However, we find that their methodology to detect DNS censorship makes unstated assumptions which we highlight and work around in our paper. Additionally, their HTTP censorship detection technique relies heavily on manual inspection, making a study at our scale untenable. We propose a novel HTTP censorship detection algorithm that requires no manual inspection, and is more accurate (in terms of F1 score) than previous automated methods.

*Joint first authors

WebSci '20, July 6–10, 2020, Southampton, United Kingdom

ACM ISBN 978-1-4503-7989-2/20/07…$15.00
https://doi.org/10.1145/3394231.3397891

[1] This list can be accessed at https://bit.ly/blockedwebsitelist

2 RELATED WORK

There have been a fair number of previous studies which explore censorship mechanisms in different countries such as China [13, 31, 37, 46], Pakistan [7, 29, 34], Syria [12], Italy [7], Iran [10], and Korea [7]. Additionally, web censorship monitoring tools such as CensMon [42], and OONI [19] have allowed a similar analysis on a global scale. Such works highlight that countries across the world adopt a melange of techniques to censor the web.

Amongst the most prevalent is DNS based blocking, where the network responds to DNS queries for websites it wishes to block with either (i) DNS errors [30, 34, 38] or (ii) incorrect IP addresses [9, 10, 32, 44, 47]. Another popular technique employed by networks to filter websites is examining HTTP traffic and looking for (i) HTTP headers for blacklisted hostnames, or (ii) the HTTP request and/or response bodies for certain keywords [10, 15, 29, 34, 37, 42, 46]. Upon detecting such requests, censors have been found to either explicitly serve censorship notices [10, 34], close established HTTP connections [37], or both [47]. Some instances of filtering traffic by inspecting TCP/IP packets for destination blacklisted IP addresses have also been reported in Syria [12], Italy [7], and China [13]. There is also recent evidence that China is inspecting and filtering HTTPS web traffic based on the Server Name Indication present in the TLS handshake [13], but previous work has reported that no Indian ISP uses this technique [47].

In the Indian context, there has been an initial attempt to understand the censorship mechanisms employed by Indian ISPs [23, 47]. However, as we find, these studies have not uncovered the full extent of web censorship in India in terms of both technical mechanisms and scale.

3 DATA CURATION

We compile a list of of potentially blocked websites from three types of sources:

Government orders. A website/URL blocking order may come from the Government of India [1, 2]. These orders are usually not in the public domain. For orders issued under section 69A of the IT Act, a confidentiality clause prevents any party from disclosing its contents [3]. We collect published and leaked Government orders that are available publicly, which contribute 890 URLs to our corpus.

Court orders. The various courts in India also have the power to issue website blocking orders[2]. Not all such orders are available in the public domain [27]. However, the Government and BSNL (a public company operating as an ISP) have provided portions of this list when under pressure to respond to Right to Information (RTI) requests. [4, 20]. Court orders contribute 9367 URLs to our list.

User reports. The Internet Freedom Foundation collects and publishes reports from internet users who notice blocked websites[2]. These contribute an additional 62 URLs to our list.

Collecting data from these sources led to a total of 9673 unique URLs. Given that most of these URLs are sourced from recent court orders, there is a high possibility of them being currently blocked. The scope of our analysis is restricted to website-level (rather than webpage-level) blocking, and so we extract unique domain names from this list, resulting in 5798 websites. To limit ourselves to active websites, we exclude all websites for which we could not resolve via Tor circuits. We end up with a total of 4379 websites, which to the best of our knowledge, is the largest known corpus of potentially blocked websites in India.

[2] https://bit.ly/iffuserreport

4 METHODOLOGY

We probe for the presence of different censorship techniques in six major Indian ISPs. These include Reliance Jio Infocomm (Jio), Bharti Airtel (Airtel), Vodafone Idea (Vodafone), Bharat Sanchar Nigam Ltd. (BSNL), Atria Convergence Technologies (ACT), and Mahanagar Telephone Nigam Ltd. (MTNL). The Telecom Regulatory Authority of India reveals that as of October 2019, these six ISPs together serve 657.46 million internet subscribers in India, i.e. 98.82% of the subscriber base in India [36].

4.1 DNS censorship

Domain Name System (DNS) resolution involves translating a hostname to its corresponding IP address(es), and is usually the first step in accessing a website. Traditional DNS resolution is prone to poisoning and injection attacks [11]. There are secure resolution protocols such as DNSSEC [41], DNS over HTTPS (DoH) [25], and DNS over TLS (DoT) [26] that mitigate these attacks; however, they are not widely deployed [14, 16].

DNS Poisoning. By default, DNS queries are sent to a resolver configured by the ISP. Thus, ISPs can return an incorrect IP address or nothing at all in response to clients' DNS queries for websites they wish to block [10, 44, 48].

DNS Injection. ISPs can intercept DNS queries for websites they wish to block and inject incorrect IP addresses in the responses [9, 32, 38].

We term an ISP's use of DNS poisoning or DNS injection attacks to block websites as DNS censorship.

4.1.1 ***Existing techniques.*** Detecting DNS censorship has previously been done by comparing responses from the test resolver with responses from trusted resolvers [19, 22]. However, this can lead to an over-reporting in censorship as these trusted resolvers can respond with a different IP address for legitimate reasons (such as load balancing) [8]. Lowe et al. circumvent this problem by selecting 5 censorship-free control resolvers and only investigating domain names for which all resolvers returned the same IP address; however, this results in a decrease in the size of the test list [31].

Another technique is to rely on the autonomous system (AS) number[3] to which the returned IP address belongs. Kuhrer et al. [30] consider a DNS response legitimate if the IP addresses returned via the trusted and tested resolvers belong to the same AS. This approach fails to take into account that a domain name can resolve to IP addresses belonging to different ASes. Yadav et al. deem a DNS response censored if the returned IP address belongs to the same AS as the client's IP address [47]. This approach rides on two unsubstantiated assumptions: (i) that the incorrect IP address returned by the ISP always belongs to the same AS as the client's, and (ii) the given website is not hosted within the same AS. Our proposed technique works around all these flaws.

4.1.2 ***Proposed technique.*** We begin by creating a set of IP addresses $IP_{d,C}$ for each domain name d in our list by combining the

[3] https://www.apnic.net/get-ip/faqs/asn/

responses obtained by resolving it via 5 censorship-free control networks (collectively termed C): (i) Tor circuits with exit nodes in US, CA and AU; and (ii) DoH servers run by Cloudflare and Google. In a test network, we attempt to resolve each domain name in our list using the ISP-assigned resolver. If the resulting IP address is present in $IP_{d,C}$, we conclude that the ISP is not using DNS censorship to block that website. Otherwise, similar to [30, 47], we flag the domain name censored if (i) the resolver responds with an error (for eg. `NXDOMAIN`), or (ii) the resolver responds with a bogon IP[4].

We further investigate the list of domain names D', for which the ISP-configured resolver returns an IP address not found in $IP_{d,C}$. For these domain names, the test network is returning either (i) a legitimate IP address not captured via the control networks, or (ii) an incorrect IP address. We term the second possibility as DNS tampering. As [10, 47] report, ISPs implementing DNS tampering respond with the same incorrect IP address to DNS queries for websites it wishes to block. We try to identify such behaviour by an ISP by looking at all IP addresses being returned by it for domain names in D'.

For each network n, we construct IP_n, the list of IP addresses received by resolving domain names in D' via that network. Next we calculate MRF_n, the relative frequency of the most frequent IP address in IP_n. By comparing MRF values of the test and control networks, we are able to discern if the test network responds with an abnormally recurring IP address. This would be characteristic of an ISP that is censoring websites by returning an incorrect IP address, i.e. employing DNS tampering[5]. In ISPs where we detect DNS tampering, we mark domain names for which the DNS response was the most frequent IP address as censored.

Algorithm 1: DNS Tampering Detection

Input: Test Network T, Control Networks C, Domain Names D
Result: Determines DNS tampering

$C \leftarrow \{C_1, C_2, \ldots, C_k\}$ `// control networks (Tor, DoH)`
for $d \in D$ **do**
  for $c \in C$ **do**
    $IP_{d,c} \leftarrow$ DNS response for d collected via c
  $IP_{d,C} \leftarrow \{IP_{d,c} | c \in C\}$
  $IP_{d,t} \leftarrow$ DNS response for d collected via test network
$D' \leftarrow \{d | d \in D \wedge IP_{d,t} \notin IP_{d,C}\}$ `// domain names for which test response did not match any of control responses`
for $c \in C$ **do**
  for $d \in D'$ **do**
    $IP_{d,c} \leftarrow$ DNS response for d collected via c
  $IP_c \leftarrow \{IP_{d,c} | d \in D'\}$
  for $ip \in IP_c$ **do**
    $RF_{ip,c} \leftarrow$ Relative frequency of ip in IP_c
  $MRF_c \leftarrow max(\{RF_{ip,c} | ip \in IP_c\})$
$\mu_{MRF_C} \leftarrow$ mean of $MRF_c \; \forall \; c \in C$
$\sigma_{MRF_C} \leftarrow$ standard deviation of $MRF_c \; \forall \; c \in C$
for $d \in D'$ **do**
  $IP_{d,t} \leftarrow$ DNS response for d in test network
$IP_t \leftarrow \{IP_{d,t} | d \in D'\}$
for $ip \in IP_t$ **do**
  $RF_{ip,t} \leftarrow$ Relative frequency of ip in IP_t
$MRF_t \leftarrow max(\{RF_{ip,t} | ip \in IP_t\})$
if $MRF_t - \mu_{MRF_C} > 3 * \sigma_{MRF_C}$ **then**
  return DNS Tampering present
else
  return DNS Tampering not present

4.2 TCP/IP Blocking

ISPs can block access to websites by preventing clients from connecting to the specific IP addresses the website is hosted on [7, 12, 13, 42]. Additionally, the ISP may inspect TCP packet headers for the destination port number if it wishes to block certain types of traffic for that IP address. However, this censorship technique can result in over-blocking due to the popularity of virtual hosting, which allows multiple websites to be hosted on the same IP address [17]. These pitfalls are a plausible explanation for why ISPs in Korea [7], Iran [10] and Pakistan [7, 34] do not use this technique. Yadav et al also conclude the same for Indian ISPs[47]; however, it is unclear how they determine what IP addresses to probe to detect such censorship.

4.2.1 ***Proposed technique.*** We first obtain legitimate IP addresses for websites in our test list as discussed in section 4.1.2. This precludes any DNS censorship by ISPs from interfering with our test.

Building on [42, 47], we perform a two-step test. First, we ping[6] the IP address to verify whether it is reachable through the test network. A response implies that the ISP is not filtering traffic based on the destination IP address. For such IP addresses, we then attempt to establish a TCP connection on ports 80 (used for HTTP traffic) and 443 (used for HTTPS traffic). A successful TCP 3-way handshake with a given IP address and port would imply the absence of TCP-based blocking. A failure in either step, however, can be attributed to network congestion, host unavailability or, of course, censorship by the ISP. To establish that connection failures are indeed due to censorship, we run the same tests via Tor circuits with exit nodes in censorship-free countries (USA, CA and AU). We rule out host unavailability and network congestion by reattempting failed connections five times with a delay of 100 seconds.

4.3 HTTP Filtering

Unencrypted HTTP traffic between a client and host can be intercepted and monitored. Previous studies have discovered different techniques adopted by censors for blocking HTTP access [10, 29, 34, 37, 42]. When a client attempts to access a website that the ISP seeks to block, the ISP sends back forged TCP or HTTP packets that seem to be originating from the host. These can include (i) a TCP packet with the `RST` (reset) bit set, forcing the client to kill the connection instantly [37], (ii) an HTTP 2xx response[7] containing a censorship notice [34], (iii) an HTTP 3xx response[7] redirecting the client to a URL serving a censorship notice [34], or

[4] https://ipinfo.io/bogon
[5] We leverage the fact that most websites in our curated list have a high possibility of being blocked. See section 3
[6] https://linux.die.net/man/8/ping
[7] HTTP response codes (RFC 7231) – https://tools.ietf.org/html/rfc7231#section-6.1

(iv) an HTTP 4xx/5xx response[7] conveying an HTTP error to the client [10].

4.3.1 ***Existing techniques.*** Due to the size of our corpus, we cannot rely on manual inspection as done by [47]. Existing automated techniques for detecting censored HTTP responses usually rely on making comparisons with uncensored responses, collected either via control servers set up in censorship-free countries [7, 42], or via Tor circuits [18, 19, 45, 47]. Due to the dynamic nature of content hosted on websites, comparing verbatim responses can be erroneous [28]. Moreover, the geographical location of a client can also introduce variations (such as the content language) in the received response. To mitigate these issues, prior research utilizes meta information derived from responses for comparison [19, 28, 45].

Jones et al. propose methods for identifying HTML pages which contain a censorship notice [28]. They report that comparing a test response's length and HTML DOM structure with that of an uncensored response can help identify such pages with a high accuracy. Similarly, other authors use response length in conjunction with different HTML similarity metrics for comparisons [7, 18, 19]. However, these approaches perform well only in instances where censors explicitly inject censorship notices, an assumption that generally doesn't hold true: as discussed above, censors have been known to adopt tacit approaches such as responding with HTTP errors or redirections. In the absence of HTML responses, these methods depend solely on response lengths, which as [47] discover, can be inefficacious [47]. The OONI tool [19] does a more elaborate comparison, drawing conclusions by observing differences in status codes, headers, lengths, and HTML titles. However, even their approach culminates in false positives and false negatives [47].

Building upon these approaches, we propose a more robust automated technique for detecting censored responses (outlined in algorithm 2) and use it to probe Indian ISPs for HTTP censorship.

4.3.2 ***Collecting HTTP responses.*** We begin by resolving the IP address of each domain name in our list via trusted resolvers, as discussed in section 4.1.2. Since DNS resolution may return multiple IP addresses for the same domain name, we probe all resulting `(domain name, IP address)` pairs in our experiment.

For each `(domain name, IP address)` pair, we make HTTP GET requests to the IP address, with the `HOST` header set as the domain name. This is done via 5 control servers in censorship-free countries (US, CA, GB, NE and AU), and via the test network. Unlike some studies [18, 19, 45, 47] we avoid using Tor circuits for collecting control responses, since some websites blacklist them and respond differently to HTTP requests originating from them. Additionally, instead of using just one control response [19, 28, 47], we consider multiple responses.

4.3.3 ***Proposed technique.*** After collecting the control and test responses, we follow the detection technique outlined in algorithm 2. First, we compare the HTTP status code of control responses with that of the test response. If these status codes do not match, we classify the test response as censored. However, the opposite need not necessarily imply the absence of censorship. In case the status codes are the same, we investigate further on a case by case basis as explained below.

- **2xx (Success):** We check for response length inconsistency and response body inconsistency.
- **3xx (Redirection):** We compare the domain name present in the redirect URLs.
- **4xx/5xx (Error):** We compare the session header keys.

Algorithm 2: HTTP Censorship Detection

```
Input: DOMAIN NAME dn, IP ADDRESS ip
Result: Determines HTTP censorship
control_res ← HTTP GET response for dn,ip in control networks;
test_res ← HTTP GET response for dn,ip in test network;
if connection reset while getting test_res then
    return censored;
if control_res.status_code ≠ test_res.status_code then
    return censored;
else if test_res.status_code = 2xx then
    if test_res.length inconsistent OR test_res.body inconsistent then
        return censored;
    else
        return uncensored;
else if test_res.status_code = 3xx then
    if mismatch in control_res, test_res redirect HOSTNAMEs then
        return censored;
    else
        return uncensored;
else if mismatch in control_res, test_res header keys then
    return censored;
else
    return uncensored;
```

The response length inconsistency and response body inconsistency used above is defined as follows

Response length inconsistency. Given control response lengths $(L_{c_i}, L_{c_{ii}}, ...L_{c_n})$ and a test response length L_t, we call L_t inconsistent if $|\mu_{L_c} - L_t| > 3 * \sigma_{L_c}$. Here μ_{L_c} is the mean, and σ_{L_c} the standard deviation of the control response lengths.

Response body inconsistency. For each control and test response, we generate term frequency (TF) vectors using HTML tags extracted from the response body. A test response body is called inconsistent if $|\mu_{c,c} - \mu_{t,c}| > 3 * \sigma_{c,c}$, where $\mu_{c,c}$ is the mean cosine similarity between TF vectors of control responses, $\mu_{t,c}$ the mean cosine similarity between TF vectors of test and control responses, and $\sigma_{c,c}$ the standard deviation of cosine similarity between TF vectors of control responses.

To verify the efficacy of our proposed technique, we manually inspect 500 responses from the six ISPs (a total of 3000 responses), and categorise them as censored or uncensored. We implement the previous techniques and compare their predictions on the annotated set to ours. As reported in table 1, our proposed technique detects both censored and uncensored responses with a higher f1-score than previous approaches.

4.4 SNI Based Censorship

The Server Name Indication (SNI) was designed as an extension to TLS to support the hosting of multiple HTTPS websites on the same

Detection Technique	Precision		Recall		F1 score	
	C	U	C	U	C	U
Length difference [28, 47]	0.65	0.73	0.77	0.59	0.70	0.66
HTML similarity [28]	0.45	0.44	0.62	0.28	0.52	0.34
OONI [19]	0.67	1.00	1.00	0.54	0.80	0.70
Proposed	**0.71**	0.98	0.99	**0.63**	**0.83**	**0.77**

Table 1: Performance of various HTTP censorship detection techniques. We report Precission, Recall and F1-score for Censored (C) and Uncensored (U) classes. Our proposed technique has a higher F1-score than the previous techniques.

IP address [6]. The SNI is an attribute included in the `ClientHello` message, where the client specifies the hostname it wishes to connect to. Since the SNI is in clear-text, censors can monitor this field for hostnames and block websites by preventing successful TLS connections [13, 33, 43, 49]. While such censorship has been documented in China [13] and South Korea [5], there has been no prior evidence to suggest that Indian ISPs are using this technique [47].

4.4.1 ***Proposed technique.*** For this test, we take advantage of a server configured to accept TLS connections even if it does not host the website specified in the SNI. For each potentially blocked website, we attempt to establish a TLS version 1.3 connection with that server's IP address using the website's hostname as the SNI. A successful connection would imply the absence of SNI-inspection based censorship in the test network. Using TLS version 1.3 ensures that only the IP address and servername are present in clear-text, since all subsequent communication after the `ClientHello` and `ServerHello` is encrypted [39]. This precludes the interference of any other censorship technique used by the ISP with our test.

5 RESULTS

We first report the different censorship techniques adopted by ISPs, and then compare the consistency of website blocklists across ISPs.

5.1 Censorship techniques

We notice stark differences in censorship mechanisms adopted by Indian ISPs, each using a range of techniques individually or in combination to censor websites (outlined in Table 2).

ISP	DNS	TCP/IP	HTTP	SNI
ACT	✓	✗	✓	✗
Airtel	✓	✗	✓	✗
BSNL	✓	✗	✗	✗
Jio	✗	✗	✓	✓
MTNL	✓	✗	✗	✗
Vodafone	✗	✗	✓	✗

Table 2: Censorship techniques employed by Indian ISPs

Interestingly, we notice different trends as compared to the previous study on Internet censorship in India [47]: (i) They report Airtel to be using only HTTP header inspection based censorship for blocking websites. We notice otherwise, with Airtel using DNS censorship in conjunction with the aforementioned technique. (ii) They report no instances of SNI-inspection based censorship in any ISP, whereas we observe Jio to be using it extensively for blocking (2951 websites). These new observations indicate an evolving nature of censorship mechanisms employed by Indian ISPs. Further, we notice that all ISPs using multiple censorship mechanisms are not blocking the same websites with each mechanism. For instance, ACT uses only DNS censorship for blocking 233 websites, only HTTP censorship for 1873 websites, and both to block 1615 websites. Such irregularities are illustrated in figure 1.

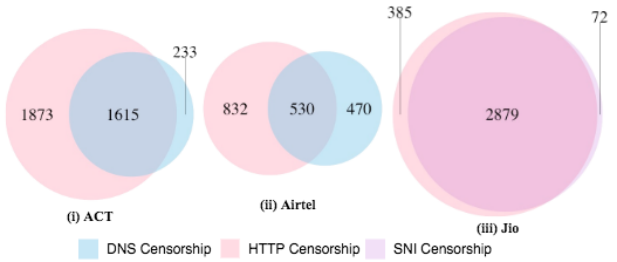

Figure 1: Censorship techniques used by (i) ACT, (ii) Airtel, and (iii) Jio for blocking websites. We notice the same ISP using multiple techniques for blocking different websites.

5.1.1 ***DNS.*** We observed DNS censorship in four ISPs: ACT, Airtel, BSNL, and MTNL. Airtel is unique in responding with `NXDOMAIN` errors to DNS queries for websites it blocks. In the other three ISPs (ACT, BSNL, and MTNL), we found that a distinct IP address appeared unusually frequently in DNS responses when we tried to resolve potentially blocked websites using the ISP-assigned resolver. This was in line with our intuition as detailed in section 4.1.2. By comparing the relative frequency of the most frequently occurring IP address in responses collected from the test to those collected from the control networks, we were able to detect which ISPs were using DNS tampering. For illustration, Figure 2 compares the frequency of IP addresses received for DNS queries for potentially-blocked websites in four networks: an ISP that uses DNS-based censorship (ACT), and three Tor circuits with exit nodes in censorship-free countries. Each of these three ISPs responded with a unique incorrect IP address. Using this fact, we conclude that there was no collateral censorship from DNS-based blocking by other ISPs.

5.1.2 ***TCP/IP.*** Given the immense collateral censorship caused by TCP/IP based methods of censorship, we were not surprised to find that, in line with [47], no ISP we investigated uses this technique.

5.1.3 ***HTTP.*** We observe HTTP-header based censorship in all the six ISPs we investigated, but find only ACT, Airtel, Jio, and Vodafone to be serving distinct censorship notices. Additionally, we notice Airtel to be closing connections and not serving any censorship notice for the majority of the websites it blocks. Using the unique signatures of these responses, we are able to identify collateral censorship in other ISPs stemming from these ISPs. For instance, we find that all instances of HTTP censorship we detect in BSNL and MTNL are attributable to Airtel and ACT. There was also a small

number of instances where we observed Vodafone's censorship notices from tests run through Jio (2), and Airtel's notices in tests run through Vodafone (2).

5.1.4 ***SNI Inspection.*** We observe SNI inspection based censorship only in one ISP, Jio. Since we did not observe SNI-based censorship in any another ISP, we rule out collateral censorship caused by Jio's employment of this technique. Out of 3340 websites we found Jio to be censoring, we notice the use of SNI inspection for 2951 websites.

5.2 Website blocklists

If we observe a particular ISP blocking a website by any method, we mark it as censored by that ISP. We term this list of censored websites by a particular ISP as its website blocklist. Note that for an ISP's blocklist, we only consider websites are censored by the ISP using its own mechanisms, i.e. we ignore collateral censorship which we highlighted in section 5.1).

From our list of 4379 potentially blocked websites that we tested, we find that 4033 appear in at least one ISP's blocklist. We use this list of 4033 websites for further analysis to see whether website blocklists are consistent across ISPs.

Interestingly, we notice large inconsistencies in ISPs' blocklists. For instance, we find that in terms of absolute numbers, ACT blocks the maximum number of websites (3721). Compared to ACT, Airtel blocks roughly half the number of websites (1892). Table 2 notes the size of each ISP's blocklist. Perhaps most surprisingly, we find that only 1115 websites out of the 4033 (just 27.64%) are blocked by all six ISPs. Figure 3 illustrates the variation in different ISPs' blocklists.

We also found that several websites (215) are being blocked by only a single ISP out of the six. For instance, ACT blocks 62 websites that are not blocked by another ISP. This calls into question whether blocking of these websites has any standing legal basis, and is potential evidence of the fact that ISPs are blocking websites arbitrarily.

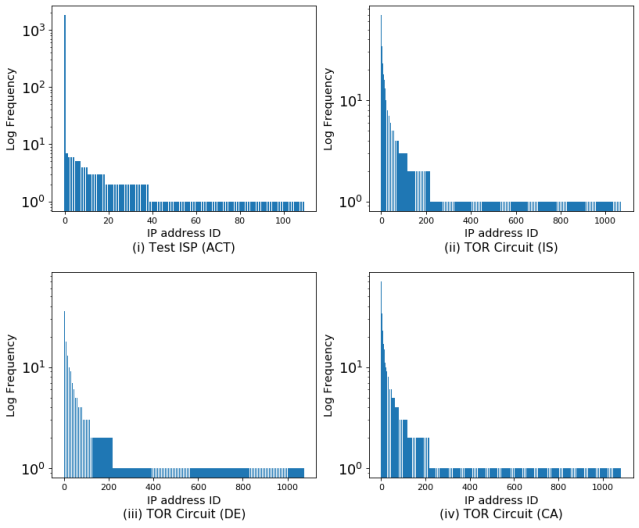

Figure 2: Log frequency plot of IP addresses received by resolving websites in our test list. We notice an abnormally large spike in subplot (i) (corresponding to ACT), compared to subplots (ii), (iii), and (iv) (corresponding to Tor circuits)

ACT	Airtel	BSNL	Jio	MTNL	Vodafone
3721	1892	3033	3340	3182	2273

Table 3: Number of websites (out of 4033) blocked by ISPs

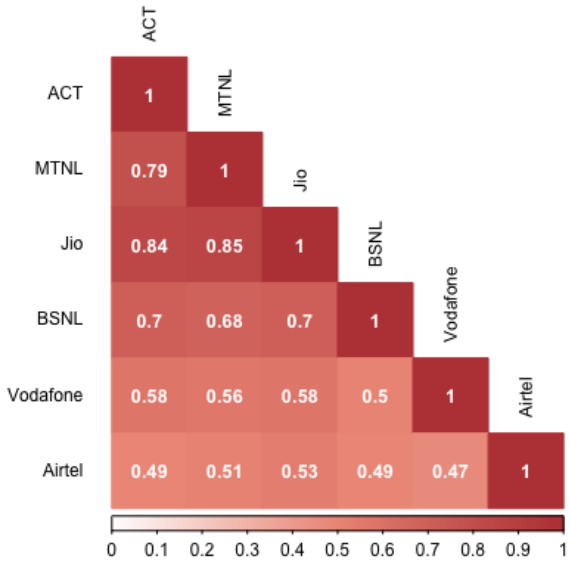

Figure 3: Heatmap illustrating the overlap of blocklists of different ISPs. For each pair of ISP blocklists L_a and L_b, we calculate the Jaccard similarity coefficient, i.e. $\frac{|L_a \cap L_b|}{|L_a \cup L_b|}$.

6 CONCLUSION

Our work presents the largest study of web censorship in India, both in terms of number of censorship mechanisms that we test for, and the corpus size of the potentially-blocked websites. In terms of censorship methods, our results confirm that ISPs in India are at liberty to use any technical filtering mechanism they wish: there was, in fact, no single mechanism common across ISPs. We also found a deep packet inspection technique, namely SNI inspection, already being employed by the largest ISP in India (Jio) to censor websites. Our work also records large inconsistencies in website blocklists across ISPs in India.

Simply stated, we find conclusive proof that Internet users in India can have wildly different experiences of web censorship.

Analysing inconsistencies in blocklists also makes it clear that ISPs in India are (i) not properly complying with website blocking (or subsequent unblocking orders), and/or (ii) arbitrarily blocking websites without the backing of a legal order. This has important legal ramifications: India's net neutrality regulations, codified in the license agreements that ISPs enter with the Government of

India[35], explicitly prohibit such behaviour. Thus, our work provides empirical evidence of the fact that Indian ISPs may be violating net neutrality regulations.

Our work also points to how the choice of technical methods used by ISPs to censor websites can decrease transparency about state-ordered censorship in India. While some ISPs were serving censorship notices, other ISPs made no such effort. For instance, Airtel responded to DNS queries for websites it wishes to block with NXDOMAIN. Jio used SNI-inspection to block websites, a choice which makes it technically impossible for them to serve censorship notices. Thus, the selection of certain technical methods by ISPs exacerbate the concerns created by the opaque legal process that allows the Government to censor websites.

Web censorship is a restriction on the right to freedom of expression and the right to access information, which are guaranteed to all citizens by the Constitution of India. There is an urgent need to reevaluate the legal and technical mechanisms of web censorship in India to make sure the curtailment is transparent, and the actors accountable.

7 FUTURE WORK

Recent user reports have suggested that website blocklists may vary within the same ISP based on geographical location. Additionally, this variance in blocklists may also occur within mobile and broadband networks belonging to the same ISP. Future work may involve running our tests from different vantage points in the country to determine the extent of such vagaries.

Contrasting results from previous studies also seem to suggest an evolving nature of the internet censorship mechanism in India. Such an evolving mechanism adopted by the ISPs demands that censorship-evasion techniques also adapt in tandem. Future research can focus on developing tools that help Indian netizens evade website blocking.

ACKNOWLEDGMENTS

We thank Devashish Gosain, Tushar Kataria, Divyank Katira and three anonymous reviewers for their constructive feedback and suggestions. We are also grateful to Suhan S and Harikarthik Ramesh for helping curate the list of potentially blocked websites used in the tests. We also thank the Internet Freedom Foundation for collecting and maintaining user reports on blocked websites, and IPinfo.io for giving us access to their IP address data.

REFERENCES

[1] 2000. Section 69A, The Information Technology Act.
[2] 2000. Section 79, The Information Technology Act.
[3] 2009. Rule 16, The Information Technology (Procedure and Safeguards for Blocking for Access of Information by Public) Rules.
[4] 2018. RTI: MeitY provides details of Blocked Websites/URLs. *SLC* (2018). https://sflc.in/rti-meity-provides-details-blocked-websitesurls
[5] 2019. Press Release. *The Korea Communications Commission* (2019). https://kcc.go.kr/user.do?mode=view&page=A05030000&dc=K05030000&boardId=1113&cp=1&boardSeq=46820
[6] Donald E. Eastlake 3rd. 2011. Transport Layer Security (TLS) Extensions: Extension Definitions. RFC 6066. https://doi.org/10.17487/RFC6066
[7] Giuseppe Aceto, Alessio Botta, Antonio Pescapè, Nick Feamster, M. Faheem Awan, Tahir Ahmad, and Saad Qaisar. 2015. Monitoring Internet Censorship with UBICA. In *Traffic Monitoring and Analysis*. Springer. http://wpage.unina.it/giuseppe.aceto/pub/aceto2015monitoring_TMA.pdf
[8] Bernhard Ager, Wolfgang Mühlbauer, Georgios Smaragdakis, and Steve Uhlig. 2010. Comparing DNS Resolvers in the Wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*. ACM, New York, NY, USA, 15–21. https://doi.org/10.1145/1879141.1879144
[9] Anon. 2012. The Collateral Damage of Internet Censorship by DNS Injection.
[10] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. 2013. Internet Censorship in Iran: A First Look. In *Presented as part of the 3rd USENIX Workshop on Free and Open Communications on the Internet*. USENIX, Washington, D.C. https://www.usenix.org/conference/foci13/internet-censorship-iran-first-look
[11] Derek Atkins and Rob Austein. 2004. Threat Analysis of the Domain Name System (DNS). RFC 3833. https://doi.org/10.17487/RFC3833
[12] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. 2014. Censorship in the Wild: Analyzing Internet Filtering in Syria. In *Proceedings of the 2014 Conference on Internet Measurement Conference (IMC '14)*. ACM, New York, NY, USA, 285–298. https://doi.org/10.1145/2663716.2663720
[13] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. 2019. On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention. In *Free and Open Communications on the Internet*. USENIX. https://www.usenix.org/system/files/foci19-paper_chai_update.pdf
[14] Zhou Li Shuang Hao Haixin Duan Mingming Zhang Chunying Leng Ying Liu Zaifeng Zhang Chaoyi Lu, Baojun Liu and Jianping Wu. 2019. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?. In *Proceedings of the 2019 Internet Measurement Conference (IMC '19)*. ACM.
[15] Jakub Dalek, Bennett Haselton, Helmi Noman, Adam Senft, Masashi Crete-Nishihata, Phillipa Gill, and Ronald J. Deibert. 2013. A Method for Identifying and Confirming the Use of URL Filtering Products for Censorship. In *Internet Measurement Conference*. ACM. http://conferences.sigcomm.org/imc/2013/papers/imc112s-dalekA.pdf
[16] I. DC Communications. 2019. DNSSEC deployment report 2019. (2019). http://rick.eng.br/dnssecstat/
[17] Benjamin Edelman. 2003. Web Sites Sharing IP Addresses: Prevalence and Significance. (2003). https://cyber.harvard.edu/archived_content/people/edelman/ip-sharing/
[18] Welch I. Esnaashari, S. and B. Chawner. 2013. WCMT: Web censorship monitoring tool. In *Australasian Telecommunication Networks and Applications Conference (ATNAC)*.
[19] Arturo Filastò and Jacob Appelbaum. 2012. OONI: Open Observatory of Network Interference. In *Free and Open Communications on the Internet*. USENIX. https://www.usenix.org/system/files/conference/foci12/foci12-final12.pdf
[20] Internet Freedom Foundation. 2019. A Tale of Rogue Pirates, Ashok Kumars and ISPs caught in the middle: RTI data on court ordered website blocking reveals worrying trends. (2019). https://bit.ly/iffwebsiterti
[21] Internet Freedom Foundation. 2019. What the block! Our net neutrality rules require a monitoring and enforcement structure SaveTheInternet. (2019). https://bit.ly/iffnetneutrality
[22] Antonio Montieri Giuseppe Aceto and Antonio Pescapè. 2017. Internet censorship in Italy: An analysis of 3G/4G networks. In *IEEE International Conference on Communications*. USENIX.
[23] Devashish Gosain, Anshika Agarwal, Sahil Shekhawat, H. B. Acharya, and Sambuddho Chakravarty. 2017. Mending Wall: On the Implementation of Censorship in India. In *SecureComm*. Springer. https://censorbib.nymity.ch/pdf/Gosain2017a.pdf
[24] Gurshabad Grover. 2019. RTI Application to BSNL for the list of websites blocked in India. *The Center for Internet and Society* (2019). https://bit.ly/grovercis
[25] Paul E. Hoffman and Patrick McManus. 2018. DNS Queries over HTTPS (DoH). RFC 8484. https://doi.org/10.17487/RFC8484
[26] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. 2016. Specification for DNS over Transport Layer Security (TLS). RFC 7858. https://doi.org/10.17487/RFC7858
[27] IFF. 2019. Why is porn being blocked in India? (2019). https://internetfreedom.in/why-is-porn-being-blocked-in-india-whattheblock/
[28] Ben Jones, Tzu-Wen Lee, Nick Feamster, and Phillipa Gill. 2014. Automated Detection and Fingerprinting of Censorship Block Pages. In *Internet Measurement Conference*. ACM. http://conferences2.sigcomm.org/imc/2014/papers/p299.pdf
[29] Sheharbano Khattak, Mobin Javed, Syed Ali Khayam, Zartash Afzal Uzmi, and Vern Paxson. 2014. A Look at the Consequences of Internet Censorship Through an ISP Lens. In *Proceedings of the 2014 Conference on Internet Measurement Conference (IMC '14)*. ACM, New York, NY, USA, 271–284. https://doi.org/10.1145/2663716.2663750
[30] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. 2015. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proceedings of the 2015 Internet Measurement Conference (IMC '15)*. ACM, New York, NY, USA, 355–368. https://doi.org/10.1145/2815675.2815683
[31] Graham Lowe, Patrick Winters, and Michael L. Marcus. 2007. *The Great DNS Wall of China*. Technical Report. New York University. https://censorbib.nymity.ch/pdf/Lowe2007a.pdf
[32] Lorenz Schwittmann Matthäus Wander, Christopher Boelmann and Torben Weis-Torben Weis. 2014. Measurement of Globally Visible DNS Injection. (2014).

[33] Kathleen Moriarty and Al Morton. 2018. Effects of Pervasive Encryption on Operators. RFC 8404. https://doi.org/10.17487/RFC8404

[34] Zubair Nabi. 2013. The Anatomy of Web Censorship in Pakistan. In *Free and Open Communications on the Internet*. USENIX. https://censorbib.nymity.ch/pdf/Nabi2013a.pdf

[35] Ministry of Communications. 2018. Regulatory Framework on Netu Neutrality. (2018). https://bit.ly/netneutralityframework

[36] Telecom Regulatory Authority of India. 2019. The Indian Telecom Services Performance Indicator Report April - June 2019. (2019). https://main.trai.gov.in/sites/default/files/PIR_01102019.pdf

[37] Jong Chun Park and Jedidiah R. Crandall. 2010. Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China. In *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems (ICDCS '10)*. IEEE Computer Society, Washington, DC, USA, 315–326. https://doi.org/10.1109/ICDCS.2010.46

[38] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. 2017. Global Measurement of DNS Manipulation. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 307–323. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/pearce

[39] Eric Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. https://doi.org/10.17487/RFC8446

[40] Reuters. 2019. Reddit, Telegram among websites blocked in India: internet groups. (2019). https://reut.rs/2UmZ3WV

[41] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends. 2005. DNS Security Introduction and Requirements. RFC 4033. https://doi.org/10.17487/RFC4033

[42] Andreas Sfakianakis, Elias Athanasopoulos, and Sotiris Ioannidis. 2011. S.: Censmon: A web censorship monitor. In *In: USENIX FOCI 2011*.

[43] Wazen M Shbair, Thibault Cholez, Antoine Goichot, and Isabelle Chrisment. 2015. Efficiently bypassing SNI-based HTTPS filtering. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 990–995.

[44] John-Paul Verkamp and Minaxi Gupta. 2012. Inferring Mechanics of Web Censorship Around the World. In *Presented as part of the 2nd USENIX Workshop on Free and Open Communications on the Internet*. USENIX, Bellevue, WA. https://www.usenix.org/conference/foci12/workshop-program/presentation/Verkamp

[45] Vasilis Ververis, George Kargiotakis, Arturo Filastò, Benjamin Fabian, and Afentoulis Alexandros. 2015. Understanding Internet Censorship Policy: The Case of Greece. In *5th USENIX Workshop on Free and Open Communications on the Internet (FOCI 15)*. USENIX Association, Washington, D.C. https://www.usenix.org/conference/foci15/workshop-program/presentation/ververis

[46] Xueyang Xu, Z. Morley Mao, and J. Alex Halderman. 2011. Internet Censorship in China: Where Does the Filtering Occur?. In *Proceedings of the 12th International Conference on Passive and Active Measurement (PAM'11)*. Springer-Verlag, Berlin, Heidelberg, 133–142. http://dl.acm.org/citation.cfm?id=1987510.1987524

[47] Tarun Kumar Yadav, Akshat Sinha, Devashish Gosain, Piyush Kumar Sharma, and Sambuddho Chakravarty. 2018. Where The Light Gets In: Analyzing Web Censorship Mechanisms in India. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18)*. ACM, New York, NY, USA, 252–264. https://doi.org/10.1145/3278532.3278555

[48] LI Bin YAN Boru, FANG Binxing and WANG Yao. 2006. Detection and Defence of DNS Spoofing Attack. *Computer Engineering* (2006). http://www.ecice06.com/EN/abstract/abstract17823.shtml

[49] Hadi Zolfaghari and Amir Houmansadr. 2016. Practical Censorship Evasion Leveraging Content Delivery Networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, New York, NY, USA, 1715–1726. https://doi.org/10.1145/2976749.2978365