A Case for Marrying Censorship Measurements with Circumvention

Aqib Nisar, Aqsa Kashaf, Zartash Afzal Uzmi, Ihsan Ayyub Qazi
SBA School of Science and Engineering
LUMS, Lahore, Pakistan
{16100099,16100223,zartash,ihsan.qazi}@lums.edu.pk

ABSTRACT

Existing research on Internet censorship primarily focuses on either measurements or circumvention. Considering these two in isolation often leads to designs with limited capabilities: Circumvention is not driven by measurement data and end users find little incentive to help gather such data. We present the preliminary design and implementation of C-Saw, a platform that offers both. The circumvention capability of C-Saw incentivizes consumers to opt-in. As more and more consumers crowdsource, the monitoring data gets richer. This, in turn, offers greater insights into the censorship mechanisms over a wider region, offering even better circumvention capabilities. C-Saw is a browser-based platform set up as a lightweight client-side proxy. C-Saw adapts its circumvention approach to the particular censorship mechanism deployed by a user's ISP, achieving a better balance of circumvention effectiveness and performance. In addition, it can also leverage the heterogeneity in filtering mechanisms across ISPs to achieve better circumvention performance. We demonstrate this using page load times across various ISPs and locations in a censored region. Unlike previous measurement approaches, C-Saw does not require the knowledge of a target URL to be tested. In fact, as URLs get blocked, their information can be monitored in real time.

Categories and Subject Descriptors

C.2.0 [**Computer-communication Networks**]: General—*Security and protection*; C.2.3 [**Computer-communication Networks**]: Network Operations—*Network Monitoring*

General Terms

Measurement, Security, Design

Keywords

Censorship, Internet Measurements, Circumvention

HotNets'15, November 16–17, 2015, Philadelphia, PA, USA
Copyright 2015 ACM. 978-1-4503-4047-2/15/11 ...$15.00
DOI: http://dx.doi.org/10.1145/2834050.2834110.

1. INTRODUCTION

Internet censorship has become increasingly pervasive with nearly 70 countries restricting Internet communication in one way or another [14]. The resulting impact on the user base is widespread and has drawn a lot of interest from networking researchers towards studying Internet censorship (see [4] and the references therein). Most such studies focus on either measuring censorship (what is blocked and where?) or devising circumvention techniques (how to bypass blocking?).

Considering measurement and circumvention systems independently, as is the current practice, often results in their individual designs having limited capabilities. For example, without a measurement system, circumvention techniques may not be well adapted to the deployed censorship, making them either ineffective or an overkill. Due to lack of measurements, existing circumvention techniques are crafted either as a 'one-size-fits-all' solution [1, 2] or based on blocking mechanisms learned through manual testing and anecdotal channels [4]. Similarly, without an offer to provide anti-censorship tools, end users have little incentive to help gather continuous censorship measurements.

We present C-Saw, a system that simultaneously provides censorship *measurements* and *circumvention*. Through end user crowd-sourcing, C-Saw gathers reliable and continuous information about censored URLs and the precise blocking mechanisms being used; this information can be valuable to researchers and activists who desire to understand censorship policies employed by ISPs and the governments [16]. At the same time, C-Saw benefits end users by offering them low overhead circumvention methods based on fine-grained measurement data [5].

The data-driven circumvention capability of C-Saw creates incentives for the end users to opt-in. As more and more users crowd-source, the monitoring data gets richer. Using this fine-grained data, C-Saw builds a database of blocking mechanisms employed by various ISPs in the region. This allows C-Saw to offer even better circumvention capabilities for better user experience.

C-Saw is set up as a browser extension working in conjunction with a client-side proxy. With C-Saw, users can choose to assist other users in accessing censored content through a peer-to-peer channel. The design of C-Saw explicitly accounts for the following challenges faced by existing censorship measurement tools and circumvention systems:

- Short of capturing packets at access routers, widespread end user deployments are generally needed for accurately and continuously measuring Internet censorship (e.g., see OONI [13] and Centinel [7]). This is particularly hard to achieve without offering incentives to the end users. Encore [5] gathers censorship data without requiring widespread user deployments by installing scripts at popular websites and then harnessing cross-origin requests. However, it requires creating incentives for popular website operators and does not measure the precise filtering mechanism being used.
- Existing censorship measurement systems require a list of target URLs to be tested for censorship [5, 13]. In many cases, such lists may not be available or fully known.
- Design of circumvention systems often relies on having a prior knowledge of the blocking mechanisms—gathered through manual measurements or anecdotal evidence. The result is to use a de facto set of circumvention methods against all blocking scenarios, which can greatly differ from provider to provider, region to region, and even time to time (§3).

C-Saw addresses all of the above challenges. We summarize the features of C-Saw below.

- For widespread user deployment, C-Saw creates incentives by offering high-performance (in terms of page load times) circumvention built into the browser.
- C-Saw measures censorship for only those URLs that are accessed by the users, without requiring a pre-populated database. In fact, a dynamic database of blocked URLs is built up as the user browses those sites and, thus, a URL not known to the user is not tested. We use existing techniques to build a highly accurate database of censored URLs (see [17] and censorship indicators in [18]).
- Fine-grained censorship measurements enable C-Saw browsers to automatically and dynamically pick a circumvention technique—from a list of standard techniques—that incurs the least overhead for a given blocking mechanism. Thus, users opt in not because they do not have access to alternate circumvention tools but because they may experience better performance in accessing blocked websites.
- Circumvention capability of C-Saw is also designed to exploit heterogeneity in ISP blocking mechanisms by using cross-ISP paths when users consent to help other users.
- The design of C-Saw allows collecting explicit feedback from users, which opens up the possibility of crafting new circumvention techniques by using machine learning algorithms based on fine-grained censorship measurements.

Through extensive experiments across several ISPs in various cities of Pakistan, we demonstrate that opportunities for improved user experience exist. Our goal in this work is two-fold: (a) to highlight the opportunities for obtaining reliable and continuous censorship related measurements by providing high performance circumvention and (b) to come up with a feasible road-map towards developing a framework that realizes these opportunities. Our preliminary results give us a reason to be optimistic and we are currently in the process of rolling out a beta version of the C-Saw platform.

2. BACKGROUND

This section provides background on filtering techniques commonly used by censors as well as some common circumvention techniques used to bypass them. The censor has a variety of choices to block content by intercepting a user request at various levels such as DNS, TCP/IP, or HTTP/HTTPS. Similarly, the user, in the censored region, can use a number of popular tools (e.g., Tor [8]) to circumvent censorship.

2.1 Censorship Techniques

We now discuss some of the most common censorship techniques employed by censors.

DNS-level blocking. This involves using DNS hijacking or DNS injection in order to manipulate DNS queries for censored content [4]. In both cases, the user can face the following consequences: (a) no response from the DNS server, (b) NXDOMAIN response, indicating that the queried domain name does not exist, or (c) a fake response containing either the IP address of a server hosting a page which tells the user that the content is blocked (referred to as *block page*) or the IP address that does not host any page.

TCP/IP blocking. This type of censorship is done by comparing the IP addresses in an IP packet against a blacklist. On a match, the censor either drops the SYN packet originated from a client or sets the RST flag in the response from the server to be blocked.

HTTP-level blocking. In this type of blocking, the censor intercepts the HTTP GET request and matches the resource path and ‘Host’ field in the header against a blacklist consisting of URLs and keywords. In case of a match, the censor can: (a) drop the GET request and send no response to the client, (b) fake a RST response to the client from the server, (c) send Error Response Codes, (d) 3XX redirection to a block page, or (e) forge the response by embedding an iframe corresponding to a block page.

TLS blocking. Requests to popular services such as Facebook and YouTube are generally through secure Transport Layer Security (TLS) protocol connections. While such connections are encrypted, censors may still monitor certain fields that are sent in plaintext. For example, censors often detect and block on the Server Name Indication (SNI) field in TLS handshake header which was introduced as a TLS extension to facilitate the existence of virtual servers [9].

In addition to these, censors use a variety of other blocking techniques, many of which are surveyed in [5].

Website/Categories	ISP-A	ISP-B
YouTube	HTTP Blocking → Redirected to a block page	1) DNS Blocking → Resolved to a local host 2) HTTP/HTTPS Blocking → Request dropped
Rest (Religious, Social, Porn,..)	HTTP Blocking → Redirected to a block page	HTTP Blocking → Block page via iframe

Table 1: Comparison of blocking mechanisms used by ISP-A and ISP-B, both of which are located in Pakistan.

2.2 Circumvention Tools/Mechanisms

Following are some of the popular techniques and tools used to circumvent censorship.

Using a global/public DNS. In case of DNS hijacking, clients can use global/public DNS servers to get the hostname resolution. This will not work in case of DNS injection.

Domain Fronting (DF). It is a technique used to hide the endpoint of a connection using HTTPS while communicating with censored hosts [12]. In a normal client-server interaction, the destination server name appears in the DNS query (plaintext), the TLS SNI extension (plaintext), and in the HTTP Host header (encrypted). With domain-fronting, the DNS query and SNI carry the name of a front-end server (which is not blocked by the censor), while the HTTP Host header (which is encrypted and thus hidden from the censor), carries the name of the intended backend server (the blocked destination). For example, `google.com` acts as a front-end server for the `youtube.com` backend destination.

VPNs. A large number of clients in censored regions use VPNs in order to connect to proxy servers outside the censored region to access content. However, these proxy servers can be easily blocked by the censor.

Tor. Tor [8] was initially designed as an anonymity tool but in recent years, it has become popular as a circumvention tool as well. It circumvents almost all kinds of blocking but fails in regions that block addresses of Tor bridges [20].

Lantern and uProxy. Using secure servers and trusted peers, Lantern provides access to blocked websites [1]. It employs a network of shared HTTPS proxy servers, and client software that allows censored users to find and use those proxy servers with their web browsers. The Lantern client also allows uncensored users to host proxy servers. Unlike Tor, it does not employ onion routing and focuses more on performance and availability than on anonymity. uProxy [2] also leverages trust relationships but runs as a browser extension.

3. COMBINING MEASUREMENTS WITH CIRCUMVENTION

In this section, we address three key questions whose answers motivate the potential benefits of combining fine-grained measurements and circumvention in a single system. (a) do different ISPs within a country employ different censorship techniques? (b) can we use fine-grained measurements about filtering mechanisms to improve end-user performance? and (c) can one leverage the diversity in blocking mechanisms to use cross-ISP paths for better performance? We answer these questions by analyzing a traffic dataset we collected from two of the largest ISPs in Pakistan.

3.1 Methodology and Dataset

Our dataset was collected from a University campus as well as residential networks in two cities within the censored region. The University connects to the Internet via two of the largest ISPs in Pakistan (referred to as ISP-A and ISP-B from now on). The dataset was collected by sending HTTP/HTTPS requests through these ISPs for different blocked websites. We use the term *test* to refer to a set of results collected by a user at a given point in time for a single URL. The University site we used for performing these tests does not itself censor the type of content being tested. While we focus primarily on YouTube, we also consider anti-religious and pornographic blocked content inside Pakistan.

3.2 Analysis

We now discuss the key insights we draw from our measurement study.

Insight-1: (a) Individual ISPs can employ different blocking techniques for the same URL, and (b) one ISP can censor separate URLs using different filtering mechanisms. We found that different ISPs used different filtering mechanisms for enforcing censorship. For example, ISP-A was carrying out HTTP-level blocking for YouTube (as well as other websites) whereas ISP-B blocked both HTTP and HTTPS traffic (see Table 1). In addition, ISP-B was also observed to be carrying out DNS-level blocking (essentially multi-stage blocking). These differences in blocking mechanisms generally exist due to (a) cost considerations (e.g., for buying filtering devices/tools as well as human resource costs) and (b) performance considerations (e.g., multi-stage censorship is usually carried out to balance traffic load across filters). Such heterogeneity in blocking mechanisms has also been observed in other countries including Yemen, Thailand, Kyrgyzstan, and China [14]. Interestingly, we further observed that a given ISP may use different filtering techniques for different URLs. For example, we observed that unlike YouTube, some websites in ISP-B were accessible with HTTPS. Fine-grained measurements can reveal such differences in blocking mechanisms, which in turn, can be used to select the most appropriate circumvention technique.

Insight-2: Circumvention tools/techniques can lead to widely different overheads. The circumvention techniques or tools employed by users can lead to different overheads and thus page load times. We carried out measurements over several weeks to study the page load times under *direct* circumven-

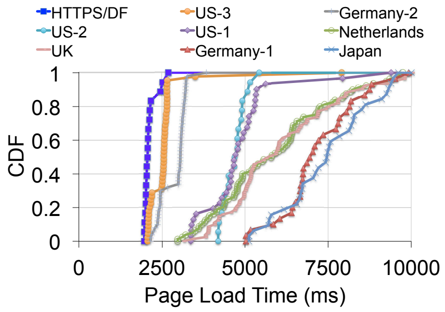


Figure 1: Performance comparison of static proxies located around the world with HTTPS/Domain-Fronting.

Static Proxy Location	Ping Latency (ms)
US-1	329
US-2	429
US-3	160
Germany-1	309
Germany-2	174
UK	228
Netherlands	172
Japan	387

Table 2: Comparison of average ping latencies to different static proxies from our measurement location. The ping latency to YouTube from the same location was 186ms.

tion mechanisms (e.g., using HTTPS in ISP-A which blocks only HTTP traffic, or using domain fronting in ISP-B to unblock HTTPS traffic) as well as with *indirect* approaches that use relays (e.g., Tor, Lantern, and static proxies). For these experiments, we focused on the page load times of the YouTube homepage (size ≈ 360KB). The results showed similar relative trends across weeks. Hence, we report only a subset of the results.

Comparison with static proxies: Users often access blocked URLs using static proxies that are spread throughout the world. We observed that page load times under a direct method (i.e., using HTTPS/DF at ISP-B without using a proxy server) were significantly better[1] than in case of static proxies located in US, Europe, and Asia as shown in Figure 1. The average ping latencies[2] are shown in Table 2. Observe that some proxies (Germany-1, UK, Japan, and Netherlands) resulted in page load times that varied widely during measurements suggesting either real-time on-path congestion or high load at the proxy.

In general, the direct method provided better throughput. For example, the average throughput under HTTPS/DF was ≈1.5Mbps whereas for most static proxies, it was less than 0.9Mbps. A simple model of TCP's slow start algorithm [6] shows that the average throughput would have been ≈2Mbps had the flow finished in slow start.

Comparison with Tor: Tor is widely used as an anonymization/circumvention tool, with hundreds of thousands of daily users [11]. Tor builds circuits for anonymization and changes them over time (usually every 10mins unless the circuit fails). Thus, we collected and isolated measurement results for every unique circuit. We recorded the location of the exit relay used by Tor across the measurement runs, which indicated its approximate latency. We observed that in most cases, using HTTPS for YouTube resulted in lower page load times as shown in Figure 2a. This is because Tor's circuits do not necessarily optimize for performance (often resulting in longer paths) even though they do account for available bandwidth when selecting relays.

Comparison with Lantern: We now compare the performance of Lantern with directly using the IP address as hostname in the URL of a blocked porn page of size ≈50KB to bypass keyword filtering. Unlike Tor, Lantern does not provide anonymity and focuses more on availability [12]. Observe that Lantern results in ≈1.5x longer page load times compared to the "IP as hostname" approach as shown in Figure 2b[3]. Also observe the long tail latency with Lantern. This happens because Lantern leverages trust relationships when choosing relays. As a result, traffic can go through longer paths compared to the direct approach. We also ran experiments with Tor and observed that its performance varied widely as circuits changed several times during the 200 back-to-back runs (where a run corresponds to a single fetch). This can be seen by the long tail of page load times under Tor.

These results suggest that different circumvention techniques can lead to widely different page load times. Therefore, fine-grained measurements can help in quantifying the performance overhead introduced by each blocking technique. This, in turn, can be used to improve end-user performance by choosing the least overhead circumvention mechanism.

Insight-3: Opportunity exists for leveraging cross-ISP paths even within a country. We now assess the potential benefit of accessing content via relays spread across ISPs (within a country) that use different filtering mechanisms. In particular, we study the efficacy of using a relay inside ISP-A for carrying HTTPS traffic from a user in ISP-B and perform comparison with HTTPS/DF and Tor. We found that in most cases, leveraging cross-ISP paths leads to the smallest page load times as shown in Figure 2c. This is because the path via ISP-A had lower latency. Of course, if page sizes are large and such paths are bottlenecked, the page load times may be higher. One can collect periodic measurements in such cases so that for a new request the most suitable circumvention technique can be used.

Note that there are countries that employ centralized censorship (e.g., Iran) [14]. In such regions, relays outside of the censoring region would be needed to leverage heterogeneity in blocking mechanisms across ISPs.

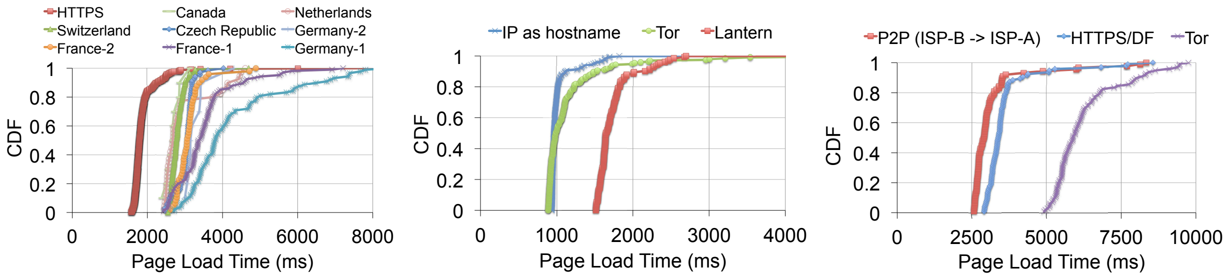
3.3 Censorship Measurement Benefits

High performance circumvention is likely to incentivize users to use C-Saw, which in turn is likely to create an evolv-

[1] Prior studies show that people react to sub-second differences in the delay of operations (see [15] and references therein).

[2] Note that the ping latencies for the static proxies did not include the latency from the proxies to YouTube.

[3] We obtained similar results in case of YouTube.



(a) HTTPS vs. Tor

(b) Lantern vs. "IP as hostname"

(c) HTTPS/DF vs. cross-ISP paths

Figure 2: Comparison of (a) HTTPS vs. Tor with different exit relay locations for fetching the YouTube homepage (≈360KB in size), (b) Lantern with "IP as hostname" (i.e., using the IP address of the blocked website in the URL as opposed to the hostname) for a porn website with ≈50KB page size, and (c) HTTPS/DF with cross-ISP paths for the YouTube homepage from a residential network (in this case, a host in ISP-B is using a host in ISP-A as a relay). Note that experiments corresponding to (a) and (b) were performed on a University campus. Each figure shows results for 200 back-to-back runs.

ing set of vantage points that are needed for reliable and continuous measurements especially from the censored regions. The key benefits are as follows: (a) users can obtain precise information about the *type* of blocking being used. This is unlike approaches that only return binary information [5] and (b) we do not mandate knowledge of a list of target URLs to be tested for censorship. In fact, as new URLs get blocked by organizations or governments, this can be communicated in real-time to the relevant sources.

4. C-SAW OVERVIEW

Motivated by the insights from our measurements, we set forth the following design goals for C-Saw:

- It should provide fine-grained measurements about censorship as experienced by a user at a particular time.
- The system should not require knowledge of a particular target URL to be tested. New URLs, as they get blocked, should automatically be measured.
- The system should allow high performance circumvention by determining the least overhead circumvention mechanism for censored URLs.

4.1 Threat Model

We assume an adversary that can block, modify, or reject a web connection at any time in order to filter access but is unwilling to filter *all* web traffic. The adversary may attempt to block client's access to C-Saw relays, thereby preventing C-Saw to collect remote measurements for better circumvention. Moreover, an adversary may also distort C-Saw measurements by providing false information. We consider these aspects in our design.

4.2 Preliminary Design

C-Saw has been structured as a proxy which runs on the client machine. It has two key modules:

- **Measurement module**: This module performs two functions. First, it carries out measurements on a URL that a *local* user is trying to access to determine if it is being censored. Second, it collects information about the type of blocking being experienced by *remote* users for the same URL. It logs this information in a local database and shares this information with remote users (when requested) as well as uploads it to designated public repositories for broader access.
- **Circumvention module**: This module maintains a list of circumvention options (e.g., HTTPS, domain fronting, relay) that can be used for each URL. For each circumvention option, it maintains a short-term history of performance metrics (e.g., page load times) and path properties (e.g., available bandwidth, latency) that it uses to determine the likely best performing circumvention strategy for a given URL.

When a user tries to access a URL (which has not been accessed recently), the measurement module identifies the blocking mechanism being used, informs the circumvention module, which returns the least overhead circumvention mechanism known so far for a given URL. In addition, the circumvention module tries alternate circumvention techniques during idle times to assess their efficacy, thus building a performance history, which it uses for future requests. C-Saw collects measurements for only those URLs that a user tries to access or allows explicitly. This is important to ensure that a user is aware of the URLs being measured in order to avoid exposing the user to unknown risks.

The collected measurements can be shared with other users by using a centralized platform or a distributed infrastructure (e.g., DHTs). Users will have the option to enable/disable the use of cross-ISP paths. However, they will only be able to use a cross-ISP path if they allow themselves to be used as potential relays.

Selecting relays. The choice of a relay depends on (a) the type of filtering being used (if at all) at the relay's ISP and (b) performance a client can get via the relay that is based on path properties (e.g., end-to-end latency and available bandwidth) and any end-system bottlenecks.

Malicious users. A malicious user can affect C-Saw by either providing false information about itself (so that others can start using it as a relay) and thereafter blackholing user traffic or by snooping on other users' traffic. The first challenge can be addressed by using a voting system where users vote on the truthfulness of the information provided by remote users. Such a system can be gamed but a careful design is part of future work. Another alternative is to leverage existing trust relationships (e.g., online social circle) to guide the choice of a relay as done in [1, 2]. The second challenge can be met by sending encrypted traffic whenever possible.

Impact on ISP traffic. The use of remote clients as relays can increase traffic from one ISP to the other. For example, in our case, some users on ISP-B may decide to use ISP-A to forward traffic using HTTPS. This may lead to increase in cross traffic. This, in turn, can incentivize ISPs to adopt more robust filtering mechanisms. In such scenarios, the cross-ISP traffic can be reduced by load balancing traffic across relays located in different ISPs if such relays are available.

4.3 Initial Prototype

Our prototype implementation is divided into two main components. The first component is a background running service that works as a proxy server. This service contains the measurement and circumvention modules and is implemented using Node.js APIs. The other component acts as a message passing method between the browser and the background service and uses the Browser's extension APIs.

The user generates a request for a particular website. The extension routes that request to the background service which processes and forwards it accordingly. The response is then fed back to the browser through the service. We are currently implementing information distribution layer between browsers based on DHTs to allow browsers to be used as potential relays for other uses. We are focusing on WebRTC [3] for its implementation which is becoming a popular framework for real time communication between browsers. Our current prototype is a Chrome-Extension since Chrome is a popular browser with a wide ranging support for extensions.

5. DISCUSSION

Ethical and privacy considerations. C-Saw measures censorship for only those URLs that a user tries to access. Thus, C-Saw incorporates explicit user consent when conducting measurements. In addition, users can choose to enable the use of relays for circumvention. This is unlike systems that perform measurements of potentially censored URLs without informed consent [5]. Despite user consent, C-Saw, like Lantern and VPNs, may expose users to some level of risk, which requires more exploration.

Why not use tools like Tor and Lantern for measurements? One could potentially extend systems like Tor and Lantern to obtain fine-grained measurements, however, the differences in their goals, raises some challenges. For example, Tor is geared towards providing anonymity, thus it does not make a distinction between blocked and unblocked websites. Both Tor and Lantern use relays for circumvention in *all* cases. Our results show, that in many scenarios, it may be better to use direct paths without the use of relays if we know the precise filtering mechanism being used.

6. RELATED WORK

Censorship measurement tools. Existing tools for measuring Internet censorship, such as OONI [13], Centinel [7], and CensMon [19], try to identify users who are willing to either host a device that collects measurements or install a customer measurement software. CensMon used PlanetLab nodes hosted in academic networks but was deployed only for a short time. While both Centinel and OONI identify the type of blocking, they have seen limited deployment [5].

Circumvention tools. Flash proxy [11] aims at creating many short-lived proxies to outpace the censor's ability to block them. Infranet [10] is designed to conceal traffic that would otherwise be blocked within seemingly normal HTTP traffic. Telex [21] allows tagging normal TLS streams cryptographically so that an ISP-level router may redirect it to a blocked destination. Unlike infranet, unblocked web sites do not need to participate in or know about circumvention. A detailed survey of existing circumvention tools and techniques can be found in [4]. Lantern [1] and uProxy [2] leverage trust relationships for choosing proxy servers for circumvention. Tor focuses on anonymity and therefore, routes all traffic (blocked or unblocked) from its circuits.

We are not aware of any tool that carries out both censorship measurements as well as circumvention, which is the gap C-Saw fills. Due to the built-in incentive mechanisms, it is expected to provide continuous measurements from diverse vantage points.

7. CONCLUSION

Collecting continuous and reliable censorship measurements has been a challenging problem due to lack of access to diverse vantage points and lack of incentives for user participation. In this paper, we show that this problem can potentially be addressed if the measurement platform also provides circumvention. Our initial measurement study shows that fine-grained knowledge about the filtering mechanisms used by ISPs can lead to significant improvement in end user performance for accessing censored URLs. This provides an added incentive to use such a platform. We hope that this study will help spawn a broader discussion in the community by aligning user incentives with collecting censorship measurements.

Acknowledgments. We would like to thank the anonymous reviewers, Ethan Katz-Bassett, Phillipa Gill, Fahad R. Dogar, Mobin Javed, and Sheharbano Khattak for their feedback on the paper. We also thank Saad Hussain for his help with the initial experiments.

8. REFERENCES

[1] Lantern. https://getlantern.org/.

[2] uProxy. https://www.uproxy.org/.

[3] WebRTC. http://www.webrtc.org.

[4] G. Aceto and A. PescapÃĺ. Internet censorship detection: A survey. *Computer Networks*, 83(0):381 – 421, 2015.

[5] S. Burnett and N. Feamster. Encore: Lightweight measurement of web censorship with cross-origin requests. In *ACM SIGCOMM*, 2015.

[6] N. Cardwell, S. Savage, and T. Anderson. Modeling tcp latency. In *IEEE INFOCOM*, 2000.

[7] Centinel. https://github.com/iclab//.

[8] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, 2004.

[9] D. Eastlake. RFC 6066: Transport layer security (tls) extensions: Extension definitions, Jan. 2011.

[10] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing web censorship and surveillance. In *USENIX Security Symposium*, 2002.

[11] D. Fifield, N. Hardison, J. Ellithorpe, E. Stark, R. Dingledine, P. Porras, and D. Boneh. Evading censorship with browser-based proxies. In *Privacy Enhancing Technologies Symposium*, 2012.

[12] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson. Blocking-resistant communication through domain fronting. *Privacy Enhancing Technologies*, 1(2), 2015.

[13] A. Filasto and J. Appelbaum. Ooni: Open observatory of network interference. In *Free and Open Communications on the Internet*, 2012.

[14] P. Gill, M. Crete-Nishihata, J. Dalek, S. Goldberg, A. Senft, and G. Wiseman. Characterizing web censorship worldwide: Another look at the opennet initiative data. *ACM Trans. Web*, 9(1):4:1–4:29, Jan. 2015.

[15] W. D. Gray and D. A. Boehm-davis. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *J. Exp. Psychol.: Applied*, pages 322–335, 2000.

[16] B. Jones, R. Ensafi, N. Feamster, V. Paxson, and N. Weaver. Ethical concerns for censorship measurement. In *Ethics in Networked Systems Research*, 2015.

[17] B. Jones, T.-W. Lee, N. Feamster, and P. Gill. Automated detection and fingerprinting of censorship block pages. In *ACM IMC*, 2014.

[18] S. Khattak, M. Javed, S. A. Khayam, Z. A. Uzmi, and V. Paxson. A look at the consequences of internet censorship through isp lens. In *ACM IMC*, 2014.

[19] A. Sfakianakis, E. Athanasopoulos, and S. Ioannidis. A web censorship monitor. In *Free and Open Communication on the Internet*, 2011.

[20] P. Winter and S. Lindskog. How the great firewall of china is blocking tor. In *Free and Open Communications on the Internet*, 2012.

[21] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the network infrastructure. In *USENIX Security Symposium*, 2011.