Roya Ensafi*, Philipp Winter, Abdullah Mueen, and Jedidiah R. Crandall

Analyzing the Great Firewall of China Over Space and Time

Abstract: A nation-scale firewall, colloquially referred to as the "Great Firewall of China," implements many different types of censorship and content filtering to control China's Internet traffic. Past work has shown that the firewall *occasionally fails.* In other words, sometimes clients in China are able to reach blacklisted servers outside of China. This phenomenon has not yet been characterized because it is infeasible to find a *large* and *geographically diverse* set of clients in China from which to test connectivity.

In this paper, we overcome this challenge by using a *hybrid idle scan* technique that is able to measure connectivity between a remote client and an arbitrary server, neither of which are under the control of the researcher performing measurements. In addition to hybrid idle scans, we present and employ a *novel side channel* in the Linux kernel's SYN backlog. We show that both techniques are practical by measuring the reachability of the Tor network which is known to be blocked in China. Our measurements reveal that failures in the firewall occur throughout the entire country without any conspicuous geographical patterns. We give some evidence that routing plays a role, but other factors (such as how the GFW maintains its list of IP/port pairs to block) may also be important.

Keywords: Tor, GFW, censorship analysis, network measurement, idle scan

DOI 10.1515/popets-2015-0005
Received 11/22/2014; revised 2/12/2015; accepted 2/12/2015.

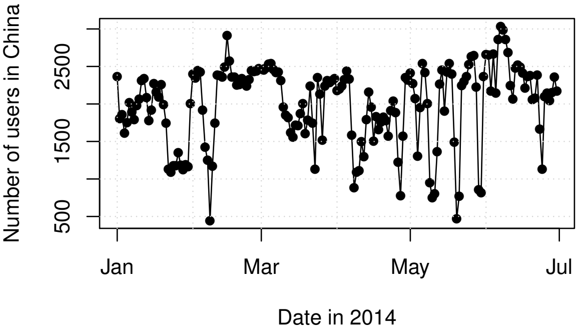


Fig. 1. The approximate amount of directly connecting Tor users (as opposed to connecting over bridges) for the first months of 2014 [40]. While the number of users varies, it rarely exceeds 3,000; only a fraction of the 30,000 users the network once counted.

1 Introduction

More than 600 million Internet users are located behind the world's most sophisticated and pervasive censorship system: the Great Firewall of China (GFW). Brought to life in 2003, the GFW has a tight grip on several layers of the TCP/IP model and is known to block or filter IP addresses, TCP ports, DNS requests, HTTP requests, circumvention tools, and even social networking sites.

This pervasive censorship gives rise to numerous circumvention tools seeking to evade the GFW by exploiting a number of opportunities. Of particular interest is the Tor anonymity network [15] whose arms race with the operators of the GFW now counts several iterations. Once having had 30,000 users solely from China, the Tor network now is largely inaccessible from within China's borders as illustrated in Figure 1.

The amount of users trying to connect to the Tor network indicates that there is a strong need for practical and scalable circumvention tools. Censorship circumvention, however, builds on *censorship analysis.* A solid understanding of censorship systems is necessary in order to design sound and sustainable circumvention systems. However, it is difficult to analyze Internet censorship without controlling either the censored source machine or its—typically uncensored—communication destination. This problem is usually tackled by obtaining access to censored source machines, finding open

***Corresponding Author: Roya Ensafi:** University of New Mexico, E-mail: royaen@cs.unm.edu
Philipp Winter: Karlstad University, E-mail: philwint@kau.se
Abdullah Mueen: University of New Mexico, E-mail: mueen@cs.unm.edu
Jedidiah R. Crandall: University of New Mexico, E-mail: crandall@cs.unm.edu

proxies, renting virtual systems, or by cooperating with volunteers inside the censoring country. In the absence of these possibilities, censorship analysis has to resort to observing traffic on the server's side and inferring what the client is seeing, which is not always feasible either.

Our work fills this gap by presenting and evaluating network measurement techniques which can be used to expose censorship while controlling *neither the source nor the destination machine*. This puts our study in stark contrast to previous work which had to rely on proxies or volunteers, both of which provide limited coverage of the censor's networks.

Our techniques are currently limited to testing *basic IP connectivity*. Thus, we can only detect censorship on lower layers of the network stack, *i.e.*, before a TCP connection is even established. This kind of low-level censorship is very important to the censors, however. For example, while social media controls on domestic sites in China, such as Weibo, can be very sophisticated, users would simply use alternatives such as Facebook if the low-level IP address blocking were not in place to prevent this. Also, deep packet inspection (DPI) does not scale as well in terms of raw traffic as does lower-level filtering. Nevertheless, we acknowledge that our techniques are not applicable if censors only make use of DPI to block Tor as it was or is done by Ethiopia, Kazakhstan, and Syria [1].

1.1 Limitations of Previous Work

Previous related work on China's filtering of packets based on IP addresses and TCP ports left open two key questions: are there geographic patterns in the cases where the GFW lets through packets that it would otherwise block?; and, are the GFW's failures on a given route persistent or intermittent?

Winter and Lindskog [46] used a virtual private server (VPS) in Beijing as a vantage point to reach the set of all Tor relays. For experiments where seeing packets on both the client and server side was necessary they performed the experiments between the VPS in Beijing and a Tor relay in Sweden, that was under their control. They observed that, for some Tor relays, their VPS in China was able to connect to those relays.

Ensafi *et al.* [17, 18] also observed inconsistencies with respect to clients in China being able to communicate over IP with Tor servers. However, their experiments were not designed to locate geographic patterns or answer other key questions about the GFW's filtering of Tor in the routing layer. Specifically:

- Ensafi *et al.* assumed that SYN packets are treated the same as RST packets by the GFW, but had no way to verify this. Winter and Lindskog observed that, from their VPS in Beijing, for Tor relays only the SYN/ACK from the server is blocked, not the SYN from the client to the server. One of our key results in this paper is that this observation also applies to China in general for a lot of different geographic locations. In this paper we present a novel SYN backlog side channel for this purpose.
- Ensafi *et al.*'s method for choosing clients in China was uniform throughout the IP address space of the country, not stratified geographically, so that any geographic patterns in their results could have been biased. We instead use stratified sampling based on longitude and latitude.
- Ensafi *et al.*'s data was not culled to ensure that Tor relays which appeared in the consensus but were actually not accessible did not appear in the measurements. We more thoroughly culled our data to remove these kinds of distortions.
- Ensafi *et al.*'s measurements between clients and servers did not form a full bipartite graph, meaning that not every client was tested with every server and *vice versa*. Our experiments were designed to form a full bipartite graph to ensure completeness and avoid distortions in the results.

So, to return to our key open questions:

- Are there geographic patterns in the cases where the GFW lets through packets that it would otherwise block? *No. Our results indicate that failures occur throughout the country and that there are no conspicuous geographic patterns.*
- Are the GFW's failures on a given route persistent or intermittent? *Both. Some routes see persistent failures throughout that day, and some routes see only intermittent failures.*

In summary, this paper makes the following contributions:

- We answer the two aforementioned key questions, and confirm other key observations (such as that the GFW blocks SYN/ACKs entering the country mostly, and not SYNs leaving the country) that can provide important clues about how the GFW operates.
- We describe the first real-world application of the hybrid idle scan [17, 18] to a large-scale Internet measurement problem, in which we measure the

connectivity between the Tor anonymity network and clients in China over a period of four weeks.

- We present and evaluate a novel side channel based on the Linux kernel's SYN backlog which enables indirect detection of packet loss.

Our results call into question some basic assumptions about the GFW, such as the assumption that China uses the consensus file (a list of all available relays) as their list for blocking Tor or the assumption that the blocking occurs at the border. These assumptions should be researched further, including active probing [46], bridge discovery [4], and the role of routing in the GFW's failures (see discussion in Section 5).

2 Networking Background

The research questions we seek to answer require high geographical diversity of clients in China. Typically, such a study would only be possible if we could find and control vantage points in all of China's provinces. Instead, we exploit side channels allowing us to detect intentional packet dropping—without controlling the two affected machines. In particular, we use *hybrid idle scans* (see Section 2.3) and *SYN backlog scans* (see Section 2.1). The idea behind these side channels as well as their prerequisites are discussed in this section.

2.1 Side Channels in Linux's SYN Backlog

A performance optimization in the Linux kernel's SYN backlog can be used to detect intentional packet dropping. Half-open TCP connections of network applications are queued in the kernel's *SYN backlog* whose size defaults to 256. These half-open connections then turn into fully established TCP connections once the server's SYN/ACK was acknowledged by the client. If a proper response is not received for an entry in the SYN backlog, it will retransmit the SYN/ACK several times. However, if the SYN/ACK and its respective retransmissions are never acknowledged by the client, the half-open connection is removed from the backlog. When under heavy load or under attack, a server's backlog might fill faster than it can be processed. This causes attempted TCP connections to not be fully handled while pending TCP connections time out. The Linux kernel mitigates this problem by *pruning* an application's SYN backlog. If the backlog becomes more than half full, the kernel begins to reduce the number of SYN/ACK retransmissions for all pending connections [2]. As a result, half-open connections will time out earlier which should bring the SYN backlog back into uncritical state. We show that the Linux kernel's pruning mechanism—by design a *shared resource*—constitutes a side channel which can be used to measure intentional packet drops targeting a server. This is possible without controlling said server.

Our key insight is that we can remotely measure the approximate size of a server's SYN backlog by sending SYN segments and counting the number of corresponding SYN/ACK retransmissions. Starting with version number 2.2, the Linux kernel retransmits unacknowledged SYN/ACK segments five times [3]. As a result, we expect to receive the full number of five retransmissions when querying a service whose SYN backlog is less than half full. If, on the other hand, the backlog becomes more than half full, we will observe less than five retransmissions. When applied to the problem of intentional packet dropping, this allows us to infer whether a firewall blocks TCP connections by dropping the client's SYN or the server's SYN/ACK segment.

It is worth mentioning that a server's backlog state can also be inferred by coercing it into using *SYN cookies* [19]. A server using SYN cookies reveals that its SYN backlog is completely full. However, this measurement technique is effectively a SYN flood and TCP connections which were established using SYN cookies suffer from reduced throughput due to the lack of window scaling. In contrast to triggering SYN cookies, our technique has no negative impact on servers or other clients' connections, when applied carefully.

2.2 The Global IP Identifier

IP identifiers (IPIDs) are unique numbers assigned to IP packets in case they are fragmented along a path. The receiving party is able to reassemble the fragmented packets by looking at their IPID field. Most modern TCP/IP stacks increment the IPID field per connection or randomize it, as opposed to *globally incrementing* it. A machine with a globally incrementing IPID keeps a global counter that is incremented by 1 for every packet the machine sends, regardless of the destination IP address. Being a *shared resource*, the IPID can be used by a measurement machine talking to a remote machine to estimate how many packets the remote machine has sent to other machines. Throughout this paper, we refer to machines with globally incrementing IPIDs as simply machines with "global IPIDs."

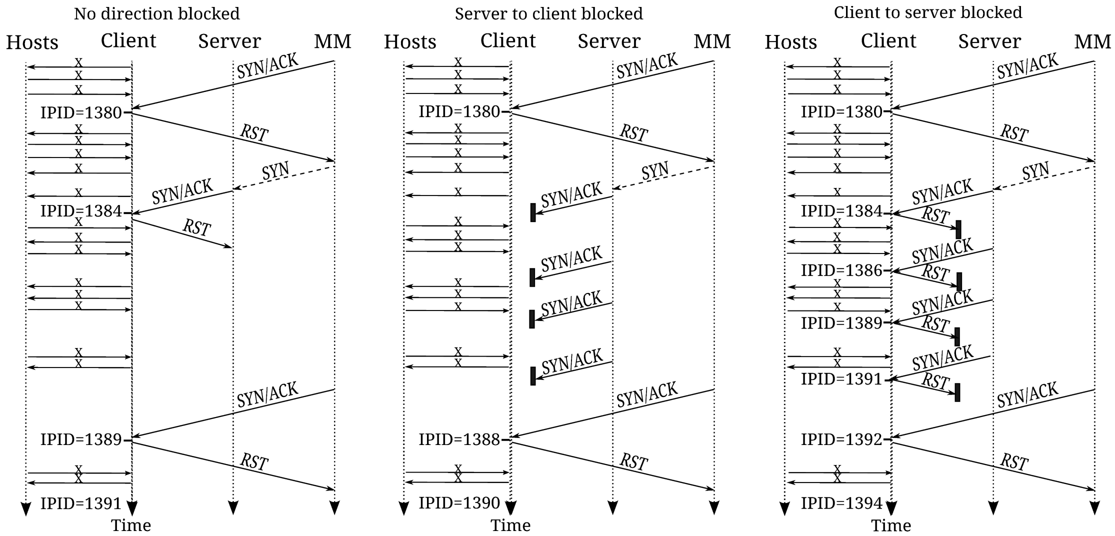


Fig. 2. Three different cases of packet dropping that the hybrid idle scan can detect. MM is our measurement machine. Despite high amounts of noise caused by different hosts communicating with the client, the ARMA modeling can still detect the blocking case correctly.

2.3 Hybrid Idle Scan

Ensafi *et al.* [17, 18] presented a new method for remotely detecting intentional packet drops on the Internet via side channel inferences. Their technique can discover packet drops (*e.g.*, caused by censorship) between two remote machines, as well as infer in which direction the packet drops are occurring. The only major requirements for their approach are a client with a global IPID and a target server with an open port. Access to the client or the server is not required. Conceptually, the hybrid idle scan technique can turn approximately 1% of the total IPv4 address space [18] into vantage points that can be used to measure IP address-based censorship—without having root access on those machines. This is why we employ the hybrid idle scan technique for our geographic study of how Tor is blocked in China.

The hybrid idle scan implementation queries the IPID of the client to create a time series. By sending SYN/ACKs from the measurement machine and receiving RST responses, the IPID of the client can be recorded. The time series is used to compare a base case (when no traffic is being generated other than noise) to a period of time when the server is sending SYN/ACKs to the client (because of our forged SYNs). Recall that the hybrid idle scan assumes that the client's IPID is global and the server has an open port. By comparing two phases, one phase where no SYN packets are sent to the server and one phase where SYN packets are sent to the server with the return IP address spoofed to appear to be from the client, the hybrid idle scan technique can detect *three different cases* (plus an error case), with respect to IP packets being dropped by the network in between the client and the server:

1. **Server-to-client-dropped:** SYN/ACKs are dropped in transit from the server to the client causing the client's IPID to not increase at all (except for noise). See Figure 2.
2. **No-packets-dropped:** If no intentional packet dropping is happening, the client's IPID will go up by exactly one. See Figure 2. This happens because the server's SYN/ACK is unsolicited and answered by the client with a RST segment causing the server to remove the entry from its SYN backlog and not retransmit the SYN/ACK.
3. **Client-to-server-dropped:** The RST responses sent by the client to the server are dropped in transit. In this case, the server continues to retransmit SYN/ACKs and the client's IPID will get incremented by the total number of (re)transmitted SYN/ACKs, which is typically three to six. See Figure 2. This may indicate null routing, the simplest method for blacklisting an IP address.
4. **Error:** A measurement error happens if networking errors occur during the experiment, the IPID is found to not be global throughout the experiment, a model is fit to the data but does not match any of the three non-error cases above, the data contains too much noise and intervention analysis fails be-

cause we are not able to fit a model to the data, and/or other errors.

Auto-regressive moving average (ARMA) models are used to distinguish these cases. This overcomes autocorrelated noise in IPID values (*e.g.*, due to packet loss, packet delay, or other traffic that the client is receiving). More details about the ARMA modeling are described by Ensafi *et al.* [17, 18]. More sophisticated models, such as Hidden Markov Models, are not necessary because we are only looking for level shifts, not complex state structure.

2.4 The Tor Network

The Tor network [15] is an overlay network which provides its users with anonymity on the Internet. As of April 2014, the network consists of approximately 4,500 volunteer-run *relays*, nine *directory authorities*, and one *bridge authority*. While the relays anonymize the network traffic of Tor clients, the authorities' task is to keep track of all relays and to vote on and publish the *network consensus* which Tor clients need in order to bootstrap. It is trivial for censors to download the hourly published network consensus and block all IP address/TCP port pairs found in it. Other circumvention systems suffer from the same problem [32].

All authorities are hard-coded in the Tor client's source code and their IP addresses remain static. As a result, they constitute attractive choke points for censors. In fact, blocking the IP addresses of all nine directory authorities is sufficient to prevent direct connections to the Tor network.[1] Our study focuses on the reachability of the authorities and relays, as it is known that the GFW is blocking them [46]. Our focus is on gathering more details about this blocking and characterizing it with a large-scale spatiotemporal study.

3 Experimental Methodology

In this section, we describe the challenges our experimental methodology was designed to address, the data sets we collected, how our measurements help us to test the open questions enumerated in Section 1, and other issues.

3.1 Encountered Challenges

Over the course of running our experiments and analyzing our data, we faced a number of challenges which we discuss here.

Churn in the Tor network: While the size of the Tor network does not vary considerably over a short period of time, the network's *churn rate* can render longitudinal studies difficult. For example, the median size of Tor's network consensus (*i.e.*, the number of Tor relays in the network) in March 2014 was 5,286. In total, however, March has seen 13,343 *unique relays*, many of which were online for only hours. To minimize the chance of selecting unstable Tor relays for longitudinal studies, only relays having earned the "Stable" flag should be considered. Furthermore, the relay descriptor archives can be examined to calculate a relay's reachability over time [39]. We selected only Tor relays that had an uptime of at least five days, and filtered out all data points where a relay appeared to have left the network. After having run our experiments, we removed one Tor relay in Argentina from our data because its Tor and web ports switched during our experiments.

Diurnal patterns: For most measurements in this paper, we measured once per hour throughout the day. This avoids bias and distortion. For example, if we measured one set of clients in the morning and one set at night, differences between the two sets of clients may be due to different traffic patterns at different times of day and not a property of the different set of clients. Thus we always randomize the order of our experiments when possible and repeat all measurements every hour for at least one full day.

3.2 Experimental Design and Setup

Over the course of our experiments, we made use of two sets of Linux-based measurement machines in the U.S. and China. These two sets of machines correspond to the two data sets that we collected.

Machines in the U.S.: The three machines used for our hybrid idle scans (see Section 2.3) and SYN backlog scans (see Section 2.1) were located at the University of New Mexico (UNM). All machines had a direct link to a research network which is free from packet filtering and does not conduct egress filtering to block spoofed return IP addresses. Furthermore, the UNM measurement machines have IP addresses that are not bound to any interfaces in order to eliminate unsolicited network packets. For example, a measurement machine's kernel should never send a RST when it receives a SYN/ACK. The data set collected using the hybrid idle scan from these machines is a large-scale geographic pairing of many clients (in China and other countries) with many Tor relays and web servers around the world (mostly outside China). It complements the other data sets discussed below because it gives a complete cross-section of censorship between many clients and many servers. This data will be used to test the existence of geographic or other spatial patterns in the GFW's failures.

VPS in China: We rented a VPS in China. The system was located in Beijing (AS 23028) and was used for our SYN backlog scans discussed in Section 2.1. Our VPS provider employed a transparent and stateful TCP proxy in front of our VPS which silently dropped unsolicited segments. We carefully implemented our SYN backlog scans so they first established state whenever necessary to be unaffected by the TCP proxy. These SYN backlog scans provide a data set that speaks to our assumptions about how China blocks Tor. It complements the hybrid idle scan data set because, although the measurements are from a single client in China, it allows us to see exactly how that client experiences the censorship. This data will be used to test what kinds of packets are filtered, specifically whether RSTs are treated the same as SYNs.

3.2.1 Hybrid Idle Scans

Recall that by using hybrid idle scans, we have more freedom in choosing clients in different regions to test their reachability to different servers. Our goal is to determine blocking of Tor relays (outside of China) from the perspective of a large and geographically diverse set of clients (within China). We do not address verification of the hybrid idle scan in this paper, since this was presented in past work [18].

We are interested in knowing whether there exist different experiences of the censorship of Tor for different users in different regions. Past work showed that a small fraction of all Tor relays was accessible from a single vantage point in Beijing [46], but what about the rest of the country? A key question is: how does the GFW's architecture and China's routing affect censorship in different regions?

IP address selection: We selected clients in China (CN), North America (NA), and Europe (EU). In order to be able to select random IP addresses in China without favoring specific locations—especially large cities featuring a vast number of allocated IP addresses—we divided the map of China into 33∗65 cells corresponding to one degree of latitude and longitude. We filled this grid with all IP addresses in MaxMind's database that were documented to be in China.[2] Then, we collected IP addresses by randomly selecting a cell from our grid after checking that they employed global IPIDs. In an analogous manner, clients from the EU and NA were chosen by horizontally scanning these regions. After 24 hours, we gathered a pool of IP addresses that belonged to machines with a global IPID. Then, we continually checked the selected IP addresses for a 24-hour period to discard IP addresses that changed global IPID behavior, went down, or where our ARMA model was unable to distinguish measurement cases [18]. At the end we had 11 NA, 7 EU, and 161 CN clients to use for our measurements.

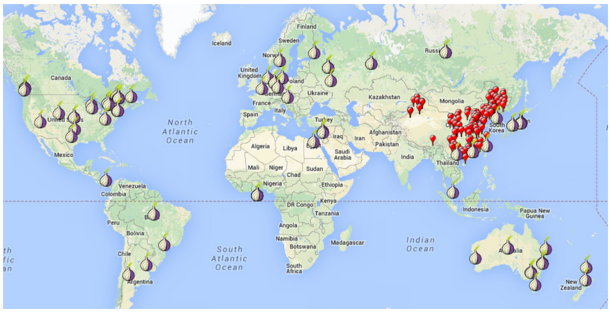


Fig. 3. The geographic distribution of all tested Tor relays (shown as onions) and of our global IPID clients in China (shown as red marks). Note that outside of Xinjiang the west of China has very little Internet penetration, which is why we have few data points in this region and the distribution is biased towards the eastern parts of China. (Map data © 2014 Google, INEGI)

Servers were chosen from three groups: Tor relays, Tor directory authorities, and web servers. Tor relays were obtained from a Tor relay status list [42]. We only selected relays with an uptime greater than five days. In order to select Tor relays in geographically diverse regions, we selected 10 Tor relays from Europe, 13 from the United States, 20 from Russia, and 101 from other countries. This way, our selected Tor relays were not biased towards Europe or the U.S., which exhibit more relays per capita than other regions. The 10 Tor authorities were obtained from the Tor source code. Web servers were chosen randomly from Alexa's top 50 websites in China [5]. All web server and Tor relay IP addresses were checked hourly to make sure that they stayed up for at least 24 hours before being selected for our measurement.

The geographic distribution of our Tor relays as well as all clients in China is illustrated in Figure 3.

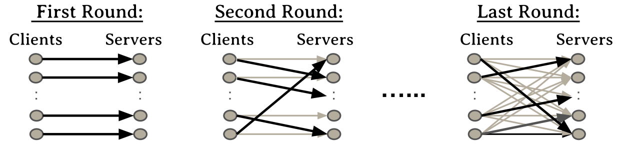
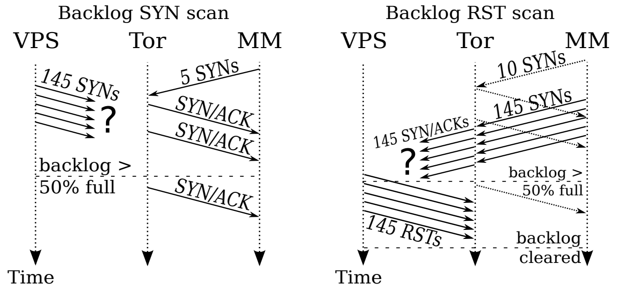


Fig. 4. After 27 days of experiments, the reachability between *all* clients and *all* servers was tested.

Creating a complete bipartite graph: We used three machines at UNM to run the hybrid idle scan experiments. We started the experiments with 180 clients and 176 servers. Each day 20 clients and approximately 20 servers were selected for each of the machines. For 22 hours[3], every hour, we performed the hybrid idle scan for each possible pair of client and server. Every "scan round" performs: *1)* two minutes of hybrid idle scans, *2)* 30 seconds of sending RSTs to clear the server's backlog, and *3)* five seconds of testing the client to assure that they remained online and kept their global IPID. Similar checks are performed to ensure that servers remain online throughout each experiment. At any given time, each IP address (client or server) was involved in only one test. After 27 days, each client's reachability was tested to all servers, *i.e*, *our clients and servers created a bipartite graph* as illustrated in Figure 4.

Pruning the data: We used the selected IP addresses throughout our experiments. Naturally, some of the hosts went down or our ARMA model was unable to distinguish measurement cases. Also, the host behind an IP address can change, *e.g.*, a client with a global IPID might lose its DHCP lease and get replaced with a client running a random IPID. To account for these issues, we performed tests throughout our experiments which cull out data points where basic assumptions are not met. For every server involved in the experiment, we had two checks: liveliness and the stable Tor flag test. After each scan, for five seconds we sent five SYN segments per second using UNM's unbound IP address. The data point passed the liveliness test only if it retransmits three or more SYN/ACKs. Also, if the server was a Tor relay, we verified that the relay was assigned the "Stable" flag (cf. Section 3.1).

For every client, for five seconds, we sent five SYN/ACKs per second using UC's unbound IP address. We expect the client to respond with RST segments totaling in number to more than half the number of sent SYN/ACKs. If this is the case then the data point passes the client's liveliness test. The results of a scan were allowed into the data set only if both the client and server passed their checks. Note that each data point is one client and one server tested one time in a given hour.



(a) SYN scan to infer whether SYN segments from VPS reach Tor. "MM" is our measurement machine.

(b) RST scan to infer whether RST segments from VPS reach Tor. "MM" is our measurement machine.

Fig. 5. The two types of backlog scans we employ. The purpose of these scans is to verify if *1)* SYN segments from China reach a Tor relay and if *2)* RST segments from China reach a Tor relay.

There was a several-hour network outage that caused a hole in a portion of one day of our data.

After culling out data that did not meet our basic assumptions, we were left with 36% of the total data collected. This 36% is the data described in Section 4 and used for our analysis.

3.2.2 Backlog Scans

After having presented the underlying side channel in Section 2.1, we now discuss the implementation of our two backlog scan types which can answer two questions, *1)* "Do SYN segments from China reach a Tor relay?" and *2)* "Do RST segments from China reach a Tor relay?". Basically, we answer both questions by first transmitting crafted TCP segments to a relay, thus manipulating its SYN backlog, and then querying its backlog size by counting the relay's SYN/ACK retransmissions. The conceptual implementation of both scan types is illustrated in Figure 5.

SYN scan: The SYN scan—depicted in Figure 5(a)—is started by MM (our measurement machine) by sending five SYN segments to Tor in order to infer the relay's backlog size when under stress.[4] After a delay of approximately 500 ms, VPS proceeds by sending 145 SYN segments whose purpose is to fill the relay's backlog by more than half. Recall that the backlog size defaults to 256, so we only fill the backlog to 59%. That way, we can make the Tor relay's kernel prune MM's SYN segments, thus reducing their retransmissions. Finally, MM knows that VPS's SYNs reached the relay if the number of SYN/ACK retransmissions for its five SYNs is lower than five. Otherwise, VPS's SYNs did not reach the relay. This type of inference is necessary because, most of the time, China's GFW drops SYN/ACKs from known Tor relays.

Before deciding to fill a relay's backlog to 59%, we obtained an understanding of the load of a typical Tor relay. In particular, we logged the backlog size of our Tor relay at Karlstad University every second for 24 hours. The median backlog size was one, the arithmetic mean was 1.29 and the maximum was 10. As a result, we believe that it is unlikely that a 59% filled backlog is problematic.

RST scan: Our RST scan incorporates an additional step but is based on the same principle. As illustrated in Figure 5(b), MM starts by sending 10 SYN segments whose purpose is, analogous to the SYN scan, to monitor the relay's backlog size. Afterwards, MM proceeds by sending 145 spoofed SYN segments with VPS's source address. Note that we cannot send the SYN segments from VPS as they might be blocked. By sending spoofed SYN segments from an unfiltered network link, we can ensure that the segments reach the Tor relay. Upon receiving the SYN segment burst, the relay replies with SYN/ACK segments which we expect to be dropped by the GFW. In the final step, VPS sends a burst of RST segments to the Tor relay. The RST segments are crafted so that every RST segment corresponds to one of the relay's SYN/ACK segments. The purpose of the RST burst is to terminate all half-open connections, thus clearing the relay's backlog. Based on how many retransmissions we observe for the 10 "probing SYNs", we can infer whether the RST segments were dropped by the GFW or not. Receiving five retransmissions means that the backlog was not cleared and the RST segments were dropped. Receiving less than five retransmissions means that the backlog was successfully cleared and the RST segments were not dropped by the GFW. This kind of inference is necessary because machines outside China cannot measure directly what happens to RST packets sent from China, and machines inside China are very limited in their ability to infer what is happening on blocked IP address/TCP port pairs.

Implementation: We implemented our scans using a collection of bash scripts and a patched version of the tool hping3 [37]. Accurate timing was crucial for our experiments. To keep the clock of our machines synchronized, we used the tool ntp which implements the network time protocol. Recall that the SYN backlog behavior we are exploiting is limited to Linux kernels (see Section 2.1). As a result, our scans targeted the subset of 94 out of our 144 Tor relays which are known to run Linux. Tor relays periodically publish their server descriptors—which includes their operating system—to all directory authorities so there is no need for us to guess the operating system of Tor relays. In general, all modern network stacks have information flow that we can use for these backlog scans, but Windows requires a high packet rate and FreeBSD-based network stacks (such as Mac OS X) were not common enough to warrant a separate implementation.

Pruning the data: By pruning the backlog scan data, we aim to make sure that the relay runs an unmodified Linux TCP/IP stack. After scanning a relay, we send three "baseline SYNs" to it in order to query its original amount of SYN/ACK retransmissions. First, we discard scans in which the relay never sent five SYN/ACK retransmissions, Linux's default value since version 2.2. For example, we found Linux relays which always retransmit SYN/ACK segments four times, regardless of their backlog size. Second, we also discard scans whose SYN/ACK retransmissions do not exhibit Linux's exponential backoff behavior. Third and finally, we discard scans where the relay was offline or other networking problems occurred. These three pruning steps discarded 774 out of all 2,094 scans (37%).

3.3 Good Internet Citizenship

We took several steps to devise our scans to be minimally invasive. First, we set up a web server on our measurement machines whose index page informed visitors about our experiments. The page contained our contact information to provide alarmed network operators with an opportunity to contact us and opt out of our measurements. Furthermore, we carefully designed our measurements so that it is very unlikely that they harmed any computers or networks. Throughout the lifetime of our experiments, we did not receive any complaints. We discuss ethical aspects of our measurements in Section 6.

4 Analysis and Results

We now analyze the two data sets we gathered; the hybrid idle scans and the backlog scans.

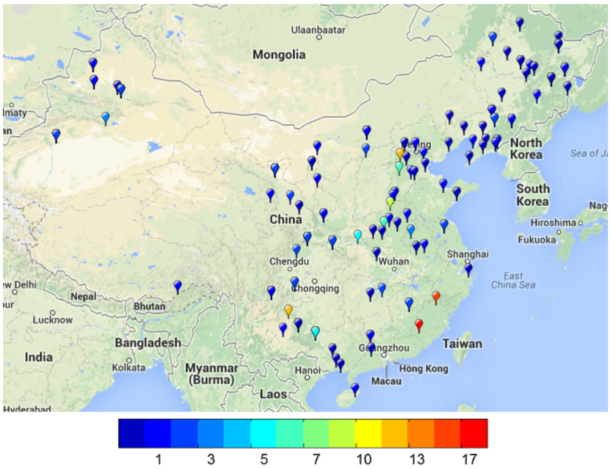


Fig. 6. The color temperature for clients corresponds to the number of observed **No-packets-dropped** cases over the entire experiment. No geographic or topological pattern is visible. Instead, the distribution matches the geographic Internet penetration patterns of China [10]. (Map data © 2014 Basarsoft, Google, ORION-ME, SK planet, ZENRIN)

4.1 Hybrid Idle Scans

The hybrid idle scan data was collected from 15 March 2014 to 10 April 2014. One client was removed from the data because we determined that it was in Hong Kong and as a result not subject to the GFW's filtering.

Table 1 shows the results of our hybrid idle scans. The column $S \rightarrow C$ is short for **Server-to-client-dropped**, *None* means **No-packets-dropped**, $C \rightarrow S$ means **Client-to-server-dropped**, and *Error* simply means **Error**. In the table's rows, *CN* is short for China, *EU* means Europe, and *NA* means North America. As for the server types, *Tor−Dir* is a Tor directory authority, *Tor−Relay* is a Tor relay, and *Web* is a web server. Our results confirm that, in general, SYN/ACKs entering China from blacklisted IP address/TCP port pairs are blocked. Some web servers were censored, and some Tor nodes were censored outside China. This is to be expected because even in countries that do not perform nation-scale Internet censorship, organizations frequently take steps to filter material such as pornography or file sharing sites. Note that highly popular websites often contain material that is subject to censorship.

The most interesting result from the hybrid idle scans is that the **No-packets-dropped** case was measured all over the country without any noticeable geographic pattern. The geographic distribution of observed **No-packets-dropped** cases is shown in Figure 6. The case distribution closely matches the distribution of our clients which, in turn, matches the geographic Internet penetration patterns of China [10]. This means that the failures in China's IP address/TCP port blacklisting mechanisms are not limited to one region or one network block. We provide a more thorough analysis in Section 4.2, which confirms that there are no conspicuous geographic patterns in the GFW's failures. The two outliers near the Taiwan Strait are only two data points, so we cannot draw conclusions from them.

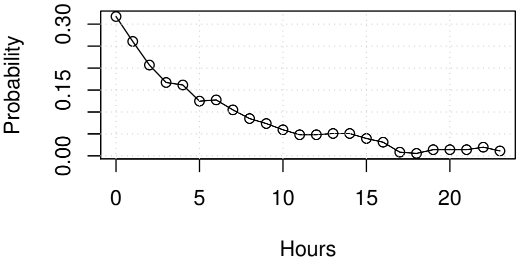


Fig. 7. The temporal association between cases of **No-packets-dropped**. The x axis shows the amount of hours since the last **No-packets-dropped** case whereas the y axis shows the probability of observing another case of **No-packets-dropped**.

We also observed that in many cases these filtering failures are persistent and *last throughout the day*. We witnessed four client/server pairs where all 22 measurements in a day returned **No-packets-dropped**. We redacted the clients' 16 least significant bits:
Client 58.193.0.0 (CN) → server 198.96.155.3 (CA)
Client 58.193.0.0 (CN) → server 161.53.116.37 (HR)
Client 58.193.0.0 (CN) → server 128.173.89.245 (US)
Client 121.194.0.0 (CN) → server 198.96.155.3 (CA)

Clients 58.193.0.0 and 121.194.0.0 are part of the Chinese Educational and Research Network (CERNET). Server 198.96.155.3 is a long-established Tor exit relay at the University of Waterloo. 161.53.116.37 and 128.173.89.245 are Tor relays in Croatia and the U.S., respectively. There were also many instances where client/server pairs showed **Server-to-client-dropped** for most of the day but also showed **No-packets-dropped** once or a handful of times.

4.2 Temporal and Spatial Association

We now seek to answer the question of whether there are any temporal or spatial associations among the **No-packets-dropped** cases observed for Tor relays tested from within China.

Temporal association is shown in Figure 7. The probabilities are computed by a simple counting technique. We have the hourly count of the number of **No-**

Table 1. The results of the hybrid idle scans.

Client	Server	$S \rightarrow C$ (%)	None (%)	$C \rightarrow S$ (%)	Error (%)
CN	Tor−Relay	116,460 (81.52)	555 (0.39)	786 (0.55)	25,061 (17.54)
CN	Tor−Dir	8,922 (64.91)	31 (0.23)	2,696 (19.61)	2,097 (15.25)
CN	Web	306 (1.23)	15,663 (62.95)	2,688 (10.80)	6,226 (25.02)
EU	Tor−Relay	18 (0.20)	8,589 (96.79)	22 (0.25)	245 (2.76)
EU	Tor−Dir	2 (0.25)	776 (96.76)	0 (0.00)	24 (2.99)
EU	Web	19 (1.23)	1,333 (86.28)	95 (6.15)	98 (6.34)
NA	Tor−Relay	45 (0.39)	11,022 (94.48)	33 (0.28)	566 (4.85)
NA	Tor−Dir	4 (0.37)	1,025 (94.73)	3 (0.28)	50 (4.62)
NA	Web	32 (1.52)	1,794 (85.06)	98 (4.65)	185 (8.77)

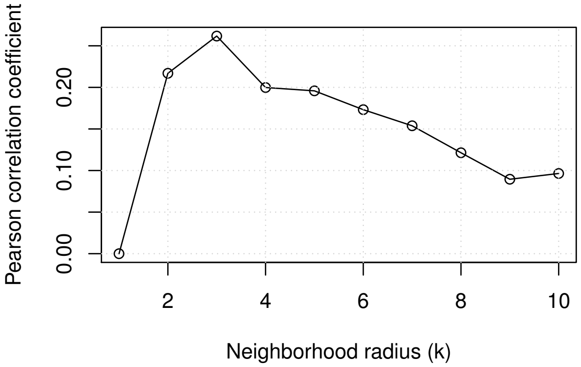


Fig. 8. Spatial association between clients in China. The x axis shows the neighborhood radius (k) and the y axis shows the Pearson correlation coefficient.

packets-dropped cases for each source. For each occurrence of **No-packets-dropped**, we check if there are other **No-packets-dropped** cases in the subsequent hours. We use 151 sources for this calculation, excluding the educational sources, which contained 353 **No-packets-dropped** cases in total. The final probabilities are averaged over all sources. With the increase in the lag amount in the x-axis, the probability decreases. This shows that **No-packets-dropped** cases generally happen in bursts of hours.

Spatial association is shown in Figure 8. We use the latitude and longitude of the sources as two-dimensional coordinates. The curvature of the earth is ignored while computing the distance between sources. For every source, we find the geographically K-nearest neighboring sources and average their count. We compute the Pearson's correlation coefficient between the count of **No-packets-dropped** cases for a source and the average of the same for the neighboring sources. Note that Pearson's correlation has a range of −1.0 to 1.0. Our maximum observed correlation value of 0.26 is, therefore, a very weak positive correlation and supports the fact that there is no significant geographical association between sources and their neighbors. With the increase of the neighborhood radius, the correlation decreases to below 0.1. Together with the fact that the cases of **No-packets-dropped** are distributed fairly evenly in all geographic regions (see Figure 6), this is strong support that there are no conspicuous geographic patterns in the GFW's failures. In other words, failures can occur in any part of the country.

A key assumption we are making here is that spatial association between sources should be observed as high correlation between their **No-packets-dropped** cases. The significance of the result is in the fact that the maximum correlation among all sources is only 26%, with the average even lower. We would need a statistical significance test if there were some cases of high correlation (>50%) in order to claim that those cases are not statistically significant in the distribution. No such cases were observed.

4.3 SYN Backlog Scans

We began our backlog scans on 24 March 2014 and ran them twice a day with approximately 12 hours in between the scans until 10 April 2014. We gathered a total of 2,094 scans and after pruning, this effort yielded 1,320 scans (63%).

4.3.1 Reachable Tor Relays

Out of all 1,320 backlog scans, 33 scans (2.5%) to 12 unique IP addresses contained the respective Tor relay's SYN/ACK segments, indicating that no filtering was

Table 2. The results of the backlog scans. Most of the time, both the SYN and the RST traversed the GFW.

	RST passes	RST dropped
SYN passes	666 (80%)	39 (4.7%)
SYN dropped	68 (8.2%)	53 (6.4%)

happening. Interestingly, 19 of these 33 scans targeted the directory authority 128.31.0.39 on port 9131. Only the RST scan and not the SYN scan yielded SYN/ACKs from the directory authority.

The results in Table 2 show that, in general, if a RST packet passes through the GFW then a SYN packet also will. This confirms one of the basic assumptions behind the hybrid idle scan, that for any client to any destination if a SYN packet is filtered by the GFW then a RST with the same source, destination, and port numbers will also be filtered. Also, the fact that most SYNs were allowed to pass through the GFW confirms that the GFW blocks Tor relays by dropping SYN/ACK segments with IP address and port information that matches known Tor relays. Other types of filtering seen for Tor relays in China (*e.g.*, dropping SYN segments) are a negligible fraction of the censorship.

5 On Topology

Our hybrid idle scan results showed that there are no obvious geographic patterns in the GFW's failures. In this section, we analyze some additional data we collected that suggests directions for future study of the GFW and why it sometimes fails at IP/port blacklisting.

5.1 Data Collected from Tor Relay Traceroutes

We used a long-established Tor relay at Karlstad University in Sweden for our traceroute measurements discussed in this section. The relay had been part of the Tor network for several months, and using our VPS we manually verified it to be blocked in China. This data set shows blocking between one Tor relay and many clients in China. Note that running more than just one Tor relay would have given us a more diverse picture but due to operational constraints, we were running only one relay.

We want to learn about the *unfiltered routes* leading into China. To investigate this question, we used our Tor relay in Europe to run traceroutes to numerous destinations in China. After a country-wide scan, we obtained a list of 3,934 IP addresses in China that responded to SYN/ACKs and were distributed geographically in a diverse way, which served as our traceroute destinations. For every IP address, we ran two TCP traceroutes; one whose TCP source port was equal to the filtered Tor port 9001 and one whose TCP port was set to the unused and unfiltered port 9002. The traceroutes had both their SYN and ACK bit set. We used a slightly modified version of the tool hping3 [37] to run the traceroutes as it allowed us to send TCP segments with a source port which is bound by the Tor process and to keep incrementing the TTL even when one hop failed to respond. Starting on 25 May 2014 and continuing until 7 June 2014, we ran the traceroutes on an hourly basis. Exactly two days worth of data were discarded because of an anomaly that we have not fully investigated where there was apparently a massive failure of the GFW. The result was a total of $3,934 \cdot 24 \cdot (14 - 2) \cdot 2 = 2,265,984$ traceroutes. We determined where the traceroutes entered China using whois and round-trip time information. We culled out a small amount of data that did not enter China through a known backbone network. In particular, some whois records list the country as China but the network is actually a point of presence (POP) in Pasadena, California, USA.

5.2 Why and Where the GFW Fails

In the traceroute data, we consider a *failure* of the GFW to be when a Tor port traceroute reaches all the way to its destination, *i.e.*, the Tor packets are not dropped anywhere along the route to the destination. Of the 3,934 destination IPs, only 135 experienced a failure at any time. All networks with at least 10 failures are shown in Table 3.

CERNET (the Chinese Educational and Research Network) is by far the network with the most failures. On only 13 occasions in our data did a Tor traceroute not reach a CERNET destination when the corresponding non-Tor port traceroute did. These 13 data points are probably due to chance. CERNET has its own backbone and international links that are separate from commercial networks. It is managed by Tsinghua University in Beijing, where much of the academic research on the GFW (also known as the Golden Shield project) is carried out [43]. It is not clear if this represents a lack of

Table 3. Number of GFW failures (#) by network in the traceroute data.

#	Network name (from whois information)
503	CERNET
81	CNC Group CHINA169 Shanxi Province Network
78	China Unicom Henan province network
58	Anhui Informationg [sic] Center
41	CHINANET
37	CNC Group CHINA169 Xinjiang Province Network
35	CNC Group CHINA169 Neimeng Province Network
31	China Unicom Heilongjiang Province Network
25	China Unicom Shandong Province Network
22	China Unicom Shanxi Province Network
20	China Mobile
17	China Unicom Hebei province network
14	China Unicom Liaoning province network
13	China Unicom Shandong province network
13	China unicom InnerMongolia province network
10	CHINANET ningxia province network

censorship of Tor on CERNET, or simply a more sophisticated (potentially at a higher layer in the network stack) implementation of censorship of Tor for this network.

After CERNET, many provincial networks experienced a significant number of failures. Note that the provinces are spread all over China, which is congruent with our conclusion from the hybrid idle scan results that the failures are spread geographically throughout China without major patterns.

5.3 Topology of the GFW

Table 4 was generated by considering each link that appears in our data separately. That is, each data point is a pair of routers where the first router appeared in a traceroute one hop before the second router. For each link we calculated:

- p: The number of times a non-Tor port packet traversed this link.
- t: The number of times a Tor port packet traversed this link.
- f: The number of times traversing this link led to a Tor port traceroute eventually reaching its destination.
- c: The number of times this link was the one where a Tor packet did not traverse it but traversed the link before it in a traceroute (*i.e.*, the "missing link" that represents censorship by one of the two routers).

We aggregated all links based on the major ISPs of the two routers. Censorship is more likely to occur within China than at the border. The majority of "missing links" (c), where one of the two routers on the link was probably performing the censorship, happen within the CHINANET or CNC Group networks. However, the number of links traversed by Tor packets that eventually reached their destination (f) within these networks is quite high, and the ratios ($r = t/p$) of links traversed by Tor (t) to links traversed by non-Tor port packets is much higher than at the borders of networks. Borders between networks may be indicators of Internet Exchange Points (IXPs) [25], meaning that we cannot discount the possibility that IXPs play a significant role in how routing issues affect the failures of the GFW. Note that no non-zero values of f occur for links between different Chinese ISPs unless one of them is CERNET. This is evidence that whether a packet gets exchanged between two Chinese ISPs is a strong indicator of if there could be a failure to censor it, which may be due to the role of IXPs.

Note also that small values of c may be due to chance, so not all non-zero values for c are evidence of censorship on that type of link.

To summarize, it appears that routing is a major factor in the GFW's failures.

6 Ethical Discussion

Our work has two ethical considerations that need to be discussed. First, our SYN backlog scans briefly fill a Tor relay's backlog in order to be able to observe packet drops. A full backlog can prevent a relay from accepting new TCP connections or cause the use of SYN cookies which can lead to reduced throughput. To prevent relays from using SYN cookies, we adapted our scan parameters to minimize the risk of completely filling a relay's SYN backlog. SYN cookies typically do not support scaled flow control windows, which is why we made every effort to avoid them. In general, the rate at which we are sending SYN packets, without intention of completing a connection, is not enough to create a denial-of-service condition on any modern network stack. Even a much higher rate of SYNs would probably not cause any issues with the service, but we kept our SYN rate as low as possible just to be conservative. For an interest-

Table 4. Analysis of links in the traceroute data.

$r = t/p$	Link type	p	t	f	c
1.0037	(Outside China)→CHINANET	471317	473068	399	18
1.0000	CERNET→ChinaTelecom	281	281	96	0
0.9989	(Outside China)→China Mobile	4518	4513	20	19
0.9985	(Outside China)→CNC Group	205184	204869	354	568
0.9952	CERNET→CHINANET	413	411	157	3
0.9930	(Outside China)→Other China	286	284	56	0
0.9896	CERNET→CERNET	8352	8265	2877	28
0.4030	CHINANET→CHINANET	1941761	782602	832	661243
0.2717	CNC Group→CNC Group	801178	217681	1523	245158
0.0439	Other China→Other China	6519	286	57	10
0.0175	CNC Group→(Outside China)	228	4	0	37
0.0115	ChinaTelecom→CHINANET	866	10	0	0
0.0099	China Mobile→China Mobile	25974	256	103	10174
0.0030	CHINANET→ChinaTelecom	331	1	0	192
0.0011	ChinaTelecom→ChinaTelecom	2828	3	0	0
0.0000	Other China→CHINANET	67	0	0	0
0.0000	CNC Group→CHINANET	8255	0	0	2247
0.0000	ChinaTelecom→Other China	1116	0	0	0
0.0000	CHINANET→Other China	456	0	0	17
0.0000	CHINANET→CNC Group	252	0	0	251

ing discussion about ethical issues related to port scans in general, we refer the reader to Durumeric *et al.* [16].

Second, our idle scans create unsolicited traffic between a client and a server. This traffic—which can be observed by the censor—is only SYN/ACKs from the server to the client and RSTs from the client to the server. As a result, we are not causing any meaningful communication other than background noise as it is also caused by port scanning activity. While one may conceptualize the hybrid idle scan technique as providing the ability to conscript a client into performing tests for us, in reality the traffic between the server and the client is no different from if the server chose to send SYN/ACKs to the client. Thus, in terms of the traffic that the censor sees, the hybrid idle scan technique is no different from if Tor relay operators performed simple connectivity measurements by directly sending SYN/ACKs.

7 Related Work

We divide related work in two subsections: network inference techniques and the Great Firewall of China.

7.1 Network Inference Techniques

There has been a fair amount of work on utilizing side channels in TCP/IP network stacks. Antirez's seminal IPID idle scan from 1998 [8, 23] and other work on idle scans [19] focus on network security. Qian *et al.* 2010 [36] used the IPID to infer the blocking direction of mail server ports for spam blocking purposes. Qian *et al.* 2012 [35] show that some firewalls exhibit behavior that can be used to infer sequence numbers and hijack connections. Chen *et al.* [9] use the IPID field to perform advanced inferences, such as the amount of internal traffic generated by a server, the number of servers in a load-balanced setting, and one-way delays. Morbitzer [31] explores idle scans in IPv6. Reverse traceroute [20] is an interesting application of indirect methods for Internet measurement.

iPlane [24] sends packets from PlanetLab nodes to carefully chosen hosts, and then compounds loss on specific routes to estimate the packet loss between arbitrary endpoints. The view of the network is fundamentally limited to the perspective of the measurement machines, however. Queen [44] utilizes recursive DNS queries to measure the packet loss between a pair of DNS servers, and extrapolates from this to estimate the packet loss rate between arbitrary hosts.

To the best of our knowledge, our work is the first use of idle scan inference techniques for a large-scale Internet measurement study where the data collected gives a view of the network from the perspective of a very large number of clients distributed over a large country. Platforms such as DIMES [28], M-Lab [29], PlanetLab [27], and RIPE Atlas [6, 26] have traditionally been the only way to measure from the perspective of a large number of clients, but they can be very limited, especially in non-Western regions of the Internet such as China.

7.2 The Great Firewall of China

The Great Firewall of China was first described in an article in 2600 magazine [41]. In 2006, Clayton, Murdoch, and Watson investigated the firewall's keyword filtering mechanism and demonstrated that it can by circumvented by simply ignoring the firewall's injected RST segments [11]. Clayton *et al.*'s study was limited to how the filtering works. *What* it filters was covered by Crandall *et al.* in 2007 [12], along with more details about routing. Using latent semantic analysis, the authors bootstrapped a set of 122 keywords which were used to probe the firewall over time. The study also shows that filtering is probably not happening at the border of China's Internet. Xu, Mao, and Halderman made an effort to pinpoint where exactly the filtering is happening [48]. The authors came to the conclusion that most filtering is happening in border ASes but some filtering is also happening in provincial networks. Park and Crandall revisited the GFW's keyword filtering mechanism and discussed why the filtering of HTML responses was discontinued in late 2008 [33].

In addition to topology and HTTP filtering, another direction of research focused on how the GFW operates on the TCP/IP layer. In 2006, Clayton *et al.* already showed that the GFW is terminating suspicious HTTP requests using injected RST segments. Weaver, Sommer, and Paxson showed that it is possible to not only distinguish genuine from injected RST segments but also to fingerprint networking devices injecting the segments [45]. More recently in 2013, Khattak *et al.* probed the GFW in order to find evasion opportunities on the TCP/IP layer [21]. Resorting to techniques first discussed by Ptacek and Newsham in 1998 [34], the authors showed that there are numerous evasion opportunities when crafting TCP and IP packets. Similarly, Winter and Lindskog showed in 2012 that packet fragmentation used to be sufficient to evade the GFW's deep packet inspection [46].

In addition to the design and topology of the GFW, some work focused on how the GFW blocks application protocols other than HTTP. In 2007, Lowe, Winters, and Marcus showed that the GFW is also conducting DNS poisoning [22]. A more comprehensive study was conducted by anonymous authors in 2012 [38] and 2014 [7]. The authors sent DNS queries to several million IP addresses in China, thereby demonstrating that the GFW's DNS poisoning causes collateral damage, *i.e.*, interferes with communication outside China. A follow-up study was conducted in 2014—also by anonymous authors [7]. The authors probed a large body of domain names to determine how filtering changes over time. Furthermore, the authors approximated the location of DNS injectors.

Most work discussed so far treated the firewall as a monolithic entity. Wright showed in 2012 that there are regional variations in DNS poisoning, thus suggesting that censorship should be investigated on a more fine-grained level with attention to geographical diversity in measurements [47]. In addition to DNS and HTTP, the GFW is known to block the Tor anonymity network. Using a VPS in China, Winter and Lindskog [46] investigated how the firewall's active probing infrastructure is used to dynamically block Tor bridges.

Novel Internet censorship measurement techniques include Dainotti *et al.* [13], who analyze several Internet disruption events that were censorship-related, and Dalek *et al.* [14], who present a method for identifying externally visible evidence of URL filtering.

8 Conclusion

In this paper, we have characterized the mechanism that the Great Firewall of China uses to block the Tor network using a hybrid idle scan that can measure connectivity from the perspective of many clients all over China. We have also presented a novel SYN backlog idle scan that can infer if packet loss is taking place without causing denial of service. These novel Internet measurement techniques open up whole new possibilities in terms of being able to measure the Internet from the perspective of arbitrary clients and servers. This is important when it comes to characterizing and documenting Internet censorship around the world, because of the difficulty in finding volunteers geographically dispersed throughout a country.

Acknowledgments

We would like to thank our shepherd, Dali Kaafar, as well as the anonymous reviewers. We would also like to thank Kasra Manavi. This material is based upon work supported by the National Science Foundation under Grant Nos. #0844880, #1017602, #0905177, #1314297, and #1420716. The author from Karlstad University was supported by a research grant from Internetfonden.

Some code and data is available at:
http://cs.unm.edu/~royaen/projects/gfw/
For the availability of other code and data, please email the authors.

References

[1] Censorship Wiki. https://censorshipwiki.torproject.org.

[2] Linux kernel source tree. http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/net/ipv4/inet_connection_sock.c?h=4d0fa8a0f01272d4de33704f20303dcecdb55df1#n562.

[3] tcp(7) - Linux man page. http://linux.die.net/man/7/tcp.

[4] Extensive Analysis and Large-Scale Empirical Evaluation of Tor Bridge Discovery. In *INFOCOM*, Orlando, FL, USA, 2012. IEEE.

[5] Alexa. Alexa top sites in China. http://www.alexa.com/topsites/countries/CN.

[6] C. Anderson, P. Winter, and Roya. Global censorship detection over the RIPE Atlas network. In *Free and Open Communications on the Internet*. USENIX, 2014.

[7] Anonymous. Towards a comprehensive picture of the Great Firewall's DNS censorship. In *Free and Open Communications on the Internet*. USENIX, 2014.

[8] Antirez. new TCP scan method, 1998.

[9] W. Chen, Y. Huang, B. F. Ribeiro, K. Suh, H. Zhang, E. de Souza e Silva, J. Kurose, and D. Towsley. Exploiting the IPID field to infer network path and end-system characteristics. In *Passive and Active Network Measurement*. Springer, 2005.

[10] China internet and mobile phone users. Available at http://www.procurasia.com/china-industrial-sourcing/china-statistics-corner/china-internet-users/.

[11] R. Clayton, S. J. Murdoch, and R. N. M. Watson. Ignoring the Great Firewall of China. In *Privacy Enhancing Technologies*. Springer, 2006.

[12] J. R. Crandall, D. Zinn, M. Byrd, E. Barr, and R. East. ConceptDoppler: A weather tracker for Internet censorship. In *Computer and Communications Security*. ACM, 2007.

[13] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé. Analysis of country-wide Internet outages caused by censorship. In *Internet Measurement Conference*. ACM, 2011.

[14] J. Dalek, B. Haselton, H. Noman, A. Senft, M. Crete-Nishihata, P. Gill, and R. J. Deibert. A method for identifying and confirming the use of URL filtering products for censorship. In *Internet Measurement Conference*. ACM, 2013.

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

[16] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: fast Internet-wide scanning and its security applications. In *USENIX Security Symposium*. USENIX Association, 2013.

[17] R. Ensafi, J. Knockel, G. Alexander, and J. R. Crandall. Detecting intentional packet drops on the Internet via TCP/IP side channels: Extended version. *CoRR*, abs/1312.5739, 2013. Available at http://arxiv.org/abs/1312.5739.

[18] R. Ensafi, J. Knockel, G. Alexander, and J. R. Crandall. Detecting intentional packet drops on the internet via TCP/IP side channels. In *Passive and Active Measurement Conference*. Springer, 2014.

[19] R. Ensafi, J. C. Park, D. Kapur, and J. R. Crandall. Idle port scanning and non-interference analysis of network protocol stacks using model checking. In *USENIX Security Symposium*. USENIX Association, 2010.

[20] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. Anderson, and A. Krishnamurthy. Reverse Traceroute. In *Networked Systems Design & Implementation*. USENIX Association, 2010.

[21] S. Khattak, M. Javed, P. D. Anderson, and V. Paxson. Towards illuminating a censorship monitor's model to facilitate evasion. In *Free and Open Communications on the Internet*. USENIX Association, 2013.

[22] G. Lowe, P. Winters, and M. L. Marcus. The Great DNS wall of China. Technical report, New York University, 2007.

[23] G. Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.Org LLC, Sunnyvale, CA, USA, 2009.

[24] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *Operating Systems Design and Implementation*. USENIX Association, 2006.

[25] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate AS-level traceroute tool. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 365–378, New York, NY, USA, 2003. ACM Press.

[26] Global RIPE Atlas Network Coverage. Available at https://atlas.ripe.net/results/maps/network-coverage/.

[27] World map of PlanetLab nodes. Available at https://www.planet-lab.org/generated/World50.png.

[28] The DIMES project: Active Agents by Countries in Last 7 Days. Available at http://www.netdimes.org/new/?q=node/52.

[29] M-Lab Platform: Server Map. Available at http://www.measurementlab.net/infrastructure.

[30] MaxMind – GeoIP2 City Accuracy. Available at https://www.maxmind.com/en/geoip2-city-database-accuracy.

[31] M. Morbitzer. TCP idle scans in IPv6. Master's thesis, Radboud University Nijmegen, The Netherlands, 2013.

[32] D. Nobori and Y. Shinjo. VPN gate: A volunteer-organized public vpn relay system with blocking resistance for bypassing government censorship firewalls. In *Networked Systems*

Design and Implementation. USENIX, 2014.

[33] J. C. Park and J. R. Crandall. Empirical study of a national-scale distributed intrusion detection system: Backbone-level filtering of HTML responses in China. In *Distributed Computing Systems*. IEEE, 2010.

[34] T. H. Ptacek and T. N. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, Secure Networks, Inc., 1998.

[35] Z. Qian and Z. M. Mao. Off-path TCP sequence number inference attack. In *Security & Privacy*. IEEE, 2012.

[36] Z. Qian, Z. M. Mao, Y. Xie, and F. Yu. Investigation of triangular spamming: a stealthy and efficient spamming technique. In *Symposium on Security and Privacy*. IEEE, 2010.

[37] S. Sanfilippo. hping. http://www.hping.org, 2006.

[38] Sparks, Neo, Tank, Smith, and Dozer. The collateral damage of internet censorship by dns injection. *SIGCOMM Computer Communication Review*, 42(3):21–27, 2012.

[39] The Tor Project. Relay descriptor archives. https://metrics.torproject.org/data.html#relaydesc.

[40] The Tor Project. Tor metrics – direct users by country. https://metrics.torproject.org/userstats-relay-country.html?graph=userstats-relay-country&start=2014-01-01&end=2014-07-01&country=cn&events=off.

[41] Tokachu. The not-so-great firewall of China. 2600 Magazine, Winter 2006–2007.

[42] TorStatus. Tor network status. http://torstatus.blutmagie.de.

[43] G. Walton. *China's golden shield : corporations and the development of surveillance technology in the People's Republic of China*. International Centre for Human Rights and Democratic Development, 2001.

[44] Y. A. Wang, C. Huang, J. Li, and K. W. Ross. Queen: Estimating packet loss rate between arbitrary internet hosts. In *Passive and Active Network Measurement*. Springer, 2009.

[45] N. Weaver, R. Sommer, and V. Paxson. Detecting Forged TCP Reset Packets. In *Network and Distributed System Security*. The Internet Society, 2009.

[46] P. Winter and S. Lindskog. How the Great Firewall of China is blocking Tor. In *Free and Open Communications on the Internet*. USENIX Association, 2012.

[47] J. Wright. Regional variation in Chinese internet filtering. Technical report, University of Oxford, 2012.

[48] X. Xu, Z. M. Mao, and J. A. Halderman. Internet censorship in china: Where does the filtering occur? In *Passive and Active Measurement Conference*. Springer, 2011.

Notes

[1] Note that the Tor Project designed and implemented bridges to tackle this very problem but the details are outside the scope of this work.

[2] MaxMind claims a 69% accuracy for identifying the correct city in China [30]. We observed that MaxMind almost always gets at least the province correct, based on whois records.

[3] Two hours per day were reserved for server data synchronization.

[4] We transmit five SYN segments rather than just one to account for packet loss.