[1 Measuring network-layer discrimination 2](#_Toc163421460)

[1.1 Overview of measurements and block detection 2](#_Toc163421461)

[1.2 Mitigating the Impact of Packet Loss 3](#_Toc163421462)

[1.3 Data 3](#_Toc163421463)

[1.4 Assessing Network-Layer Discrimination 4](#_Toc163421464)

[2 Mathematical Analysis 6](#_Toc163421465)

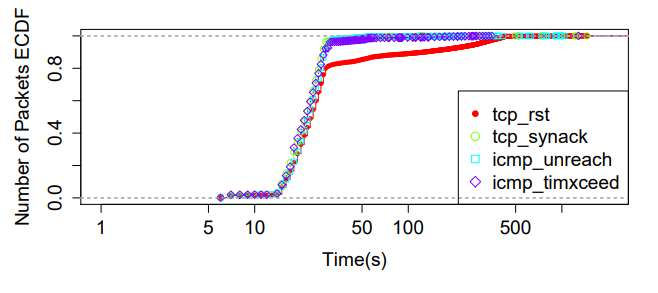
[2.1 Long-Lived Flows 6](#_Toc163421466)

# **Measuring network-layer discrimination**

As we discuss in Section II, a straightforward technique for services to block Tor is to filter traffic from publicly listed exit nodes. To broadly assess this, we measure Tor filtering using ZMap probing from both Tor exit nodes and from control (non-Tor) nodes to see how their access to remote addresses differs. For convenience we term these measurements as assessing ‘network-layer’ discrimination, though from a technical perspective they combine measurement of layer-3 and layer-4 blocking, since we restrict our measurements to attempts to connect to TCP port 80 services.

## Overview of measurements and block detection

We run our scans from Tor exit nodes and from two sets of control nodes: university nodes and a Tor middle node. We compare responses to our Tor scans with those from the baseline control scans and flag deviations as potentially reflecting discriminatory blocking. Target hosts respond to ZMap probes (TCP SYNs) in one of three ways: a) sending a SYN-ACK, which we term a successful response; b) sending a RST, which we term an unsuccessful response; or c) not responding, which we also deem an unsuccessful response. ZMap, by default, only records successful responses; we modified it to record RSTs as well. We note that for an individual probe it is not possible to distinguish a lack of response from packet loss.



**Figure 1** – Distribution of time until receiving a response packet since the last probe was sent for a full scan of IPv4.

In addition, we need to configure a timeout for ZMap to deem that a packet did not receive a response. Figure 1 shows the distribution of the time measured between sending the last scan packet and receiving a response for a full scan of IPv4. To generate this plot, ZMap logged response packets for 25 minutes after sending the last scan packet. More than 95% of all replies (excluding RSTs), and 80% of RSTs arrive within the first 30 seconds, while the rest trickle in up until 500 seconds. Though unusual, late responses could arise due to backed-off timers in the case of SYN-ACKs, huge bufferbloat, or initial latency incurred by extensive setup requirements of cellular wireless devices. Given this data, we chose a conservative cooldown value of 10 minutes for responses to come in.

## Mitigating the Impact of Packet Loss

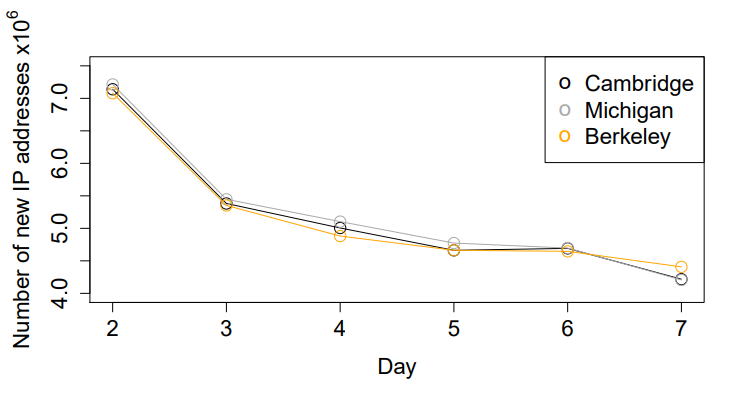
As noted above, ZMap does not allow us to distinguish between a single non-response and a packet loss event. To account for this limitation, we take care to minimize measurement loss in our measurements and to account for potential packet loss in the network.

Table 1 – Summary of control and exit node data.

|  |  |
| --- | --- |
| Control nodes | |
| Number of control nodes | 3 |
| Number of IPv4 scans | 7 per control node |
| Time span of scans | Aug 7–13 |
| Scanned IP addresses per measurement | 3,662,744,599 |
| Average hit-rate per measurement | 1.91% (σ=0.01%) |
| Average (estimated) network loss | 0.84% (σ=0.18%) |
| Tor exit nodes | |
| Number of exit node | 4 |
| Number of scans | 4 per exit node |
| Time span of scans | Aug 10–13 |
| Average hit-rate per measurement | 1.87% (σ=0.03%) |

## Data

We run our measurements from a set of three control nodes and a set of four Tor exit nodes. Two control nodes are located in US universities and one in a European university. The control node measurements serve a dual purpose: they allow us to calibrate and understand our measurement method and the data, and they serve as the baseline measurements against which we compare the Tor exit node measurements.



**Figure 2** – Number of new IP addresses each control node sees per day.

Once fully developed and debugged, for our final analysis we gathered 37 full IPv4 scans over a period of 7 days, conducting 16 from four Tor exit nodes. Table 1  shows the breakdown of the measurements run from the control and Tor exit nodes. We now turn to analyzing the final data to understand temporal churn—how the footprint changes across scans spanning multiple days—and spatial churn—how our view of the global web footprint set changes across the three control locations.

For the same location, we see significant differences in the number of IP addresses that successfully respond, even between consecutive days, ranging up to 17%. Figure 2 shows the number of new IP addresses that each site successfully contacts per day. Using the first day as the baseline, this value somewhat gradually drops from a peak of about 7 million on the second day to about 4 million on day 7. The slow convergence rate indicates that temporal churn remains high even for the same location, and that obtaining a true underlying web footprint for a given location may not be well-defined. Temporal churn is likely caused by nodes that only come online occasionally; however we do not investigate the reasons in this paper.

## Assessing Network-Layer Discrimination

Having gained confidence in our measurement methodology, we now turn to analyzing the resulting data. We conducted the scans from four high-bandwidth Tor exit nodes for 4 days (Aug. 10–13, 2015) (Table 2). These represent 3% of aggregate Tor exit bandwidth. We note that each exit node hosts 2–3 Tor processes on the same interface. As our 100 Mbps scans use the same IP address as the Tor exit node, we turned off all but one Tor process on these machines for the duration of the experiment to minimize load on the interface and potential packet loss on the interface and/or the outgoing link. These preventive measures helped reduce our reported pcap loss on the exit nodes to 0.001% of the typical number of responses seen per scan. We also chose Tor instances that use the same IP address for incoming and outgoing Tor traffic to allow our scans to trigger even ‘lazy’ blacklists.1 For three of the exit nodes, we displayed our scan notice page on port 8080 instead of the usual port 80, as the latter already displayed a separate Tor abuse complaint page

Table 2 – Description of Tor exit nodes from which IPv4 scans were conducted.

|  |  |  |  |
| --- | --- | --- | --- |
| Exit Node | Location | Uptime | Bandwidth (MB/s) |
| Axigy1 | USA | 35 days | 31.09 |
| Axigy2 | USA | 76 days | 31.46 |
| NForce2 | Netherlands | 35 days | 31.46 |
| Voxility1 | Romania | 1 day 17 hr | 16.99 |

Up through July 20, 2015, the raw OONI http requests data consist of 2,505 reports, 2,574,326 (Tor, non-Tor) request pairs, and 102,865 distinct URLs. We applied a number of restrictions to the raw data to obtain a subset useful for our analysis:

* We discarded reports before September 2014. Reports from after this date (82% of the total) occur more regularly than those from before it;
* We discarded URLs with a small number (less than 100) of request pairs. The great majority of distinct URLs were tested only a handful of times and thus not appropriate for our analysis. Though only 2% of URLs occur often enough, they account for 89% of all request pairs;
* We discarded request pairs where one or both responses were missing. A response to an http requests probe is supposed to be either an HTTP response (i.e., with a status code such as 200), or else an indication of timeout or rejection. About 20% of request pairs are anomalous and are missing a response data structure, but they are concentrated in a tiny fraction of reports and URLs.

# **Mathematical Analysis**

## Long-Lived Flows

We first consider a simplified model, in which all flows start at the same known time and have the same known duration (basically, is our observation window). The only factor distinguishing the flows is their (unknown) rate . We get:

|  |  |
| --- | --- |
|  | (1) |

where  – our prior information about the rate .

Since is a positive parameter, we express our complete lack of prior knowledge by using the scale-invariant Jeffrey’s ignorance prior . This basically says that is distributed uniformly: the probability of is proportional to . For example, and have the same probability.

|  |  |
| --- | --- |
|  | (2) |

We used ; for integer we have .

Similarly,

|  |  |
| --- | --- |
|  | (3) |

Lack of source/destination addresses in NDN helps privacy, since NDN packets carry information only about what is requested but not who is requesting it. However, a closer look reveals that this is insufficient. In particular, NDN design introduces three important privacy challenges:

1. Name privacy: NDN content names are incentivized to be semantically related to the content itself. Similar to HTTP headers, names reveal significantly more information about content than IP addresses. Moreover, an observer can easily determine when two requests refer to the same (even encrypted) content;
2. Content privacy: NDN allows any entity that knows a name to retrieve corresponding content. Encryption in NDN is used to enforce access control and is not applied to publicly available content. Thus, consumers wanting to retrieve public content cannot rely on encryption to hide what they access;
3. Cache privacy: as with current web proxies, network neighbors may learn about each others’ content access using timing information to identify cache hits;
4. Signature privacy: since digital signatures in NDN content packets are required to be publicly verifiable, identity of a content signer may leak sensitive information.