Different Ways to Use a Bridge

Sebastian Hahn
sebastian@torproject.org

Tor Tech Report 2011-11-002
November 29, 2011

1 Different Ways to Use a Bridge

When some adversary prevents users from reaching the Tor network, our most popular answer is using bridge relays (or bridges for short). Those are hidden relays, not listed along with all the other relays in the networkstatus documents. Currently, we have about 600 of them, and censors are having different luck learning and blocking them—see the 10 ways to discover Tor bridges blog post for more on how discovery approaches may work.[1] China appears to be the only place able to successfully block most bridges consistently, whereas other places occasionally manage to block Tor's handshake and as a byproduct block all bridges too.

Bridge users can be broadly grouped in three camps:

- Tor is blocked, and some way—any way—to reach the network has to be found. The adversary is not very dangerous, but very annoying.
- Tor may or may not be blocked, but the user is trying to hide the fact they're hiding Tor. The adversary may be extremely dangerous.
- Other bridge users: Testing whether the bridge works (automated or manual), probing, people using bridges without their knowledge because they came pre-configured in their bundle.

Here we examine the first two use cases more closely. Specifically, we want to look at properties of a bridge that must exist for it to be useful to a user.

[1] https://blog.torproject.org/blog/research-problems-ten-ways-discover-tor-bridges

2 Bridges—building blocks

First off, it is helpful to understand some basics about bridges and how they are used by normal users.

Bridges are very similar to ordinary relays, in that they are operated by volunteers who made the decision to help people reach the Tor network. The difference to a normal relay is where the information about the bridge is published to—bridges can choose to either publish to the Bridge Authority (a special relay collecting all bridge addresses that it receives), or to not publish their information anywhere. The former are called public bridges, the latter private bridges.

We don't have any information about the number of private bridges, but since the Bridge Authority collects data about the public bridges, we do know that bridges are used in the real world. See the bridge users[2] and networksize graphs[3] for some examples. Not having data about private bridges or their users means some of the analysis below is based on discussions with users of private bridges and our best estimates, and it can't be backed up by statistical data.

The reason we're using a Bridge Authority and collecting information about bridges is that we want to give out bridges to people who aren't in a position to learn about a private bridge themselves.

"Learning about a bridge" generally means learning about the bridge's IP address and port, so that a connection can be made. Optionally, the bridge identity fingerprint is included, too—this helps the client to verify that it is actually talking to the bridge, and not someone that is intercepting the network communication. For a private bridge, the operator has to pass on that information; public bridges wrap up some information about themselves in what is called their bridge descriptor and send that to the bridge authority. The bridge descriptor includes some statistical information, like aggregated user counts and countries of origin of traffic. Our analysis here focuses solely on the data provided by public bridges.

Once a user has learned about some bridges, she configures her Tor client to use them, typically by entering them into the appropriate field in Vidalia. Alternatively, she might use a different controller or put the data into tor's configuration file directly.

3 Learning from bridge descriptor fetches

We've been collecting bridge descriptor fetch statistics on the bridge authority, and are using this data to pose some questions and propose some changes. The statistics collected are how many bridge descriptors were served in total, and how many unique descriptors were served, as well as the 0, 25, 50, 75 and 100 percentiles of fetches per descriptor. Every 24 hours, the current statistics are written to disk and the counters reset. The current statistics are attached to this post, for closer inspection. We've also prepared two graphs in Figures 1 and 2 to easily see the data at a glance.

The first thing to note is that there aren't very many bridge descriptor fetches at all, which isn't a big surprise: The current Tor bundles don't fetch them when they're used in the typical

[2] https://metrics.torproject.org/users.html#bridge-users
[3] https://metrics.torproject.org/network.html#networksize

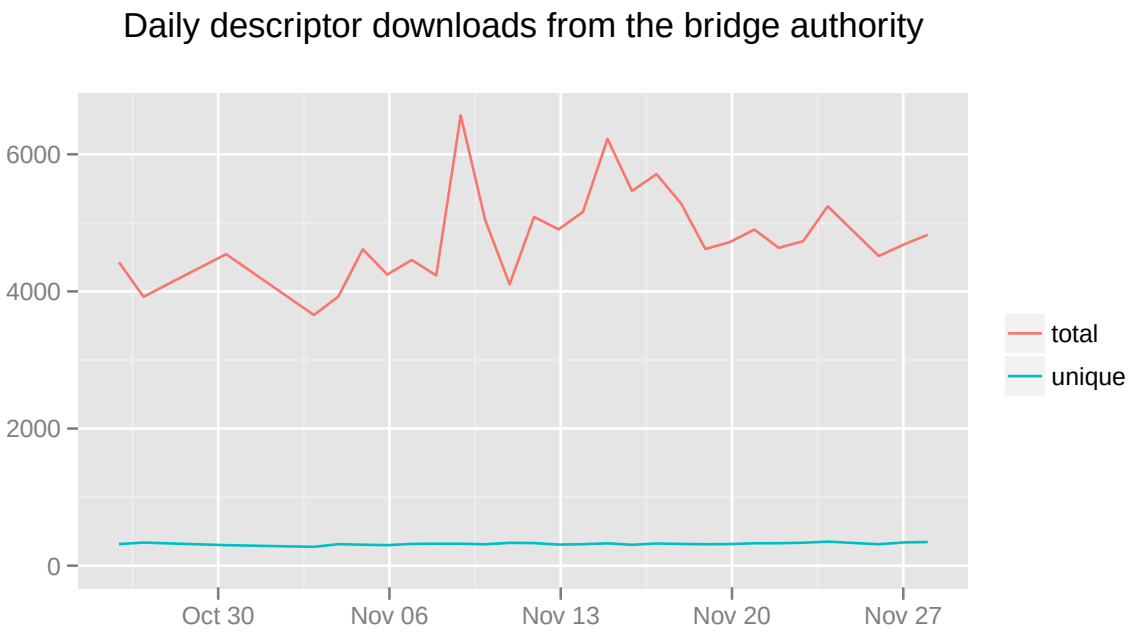

Figure 1: Daily descriptor downloads from the bridge authority

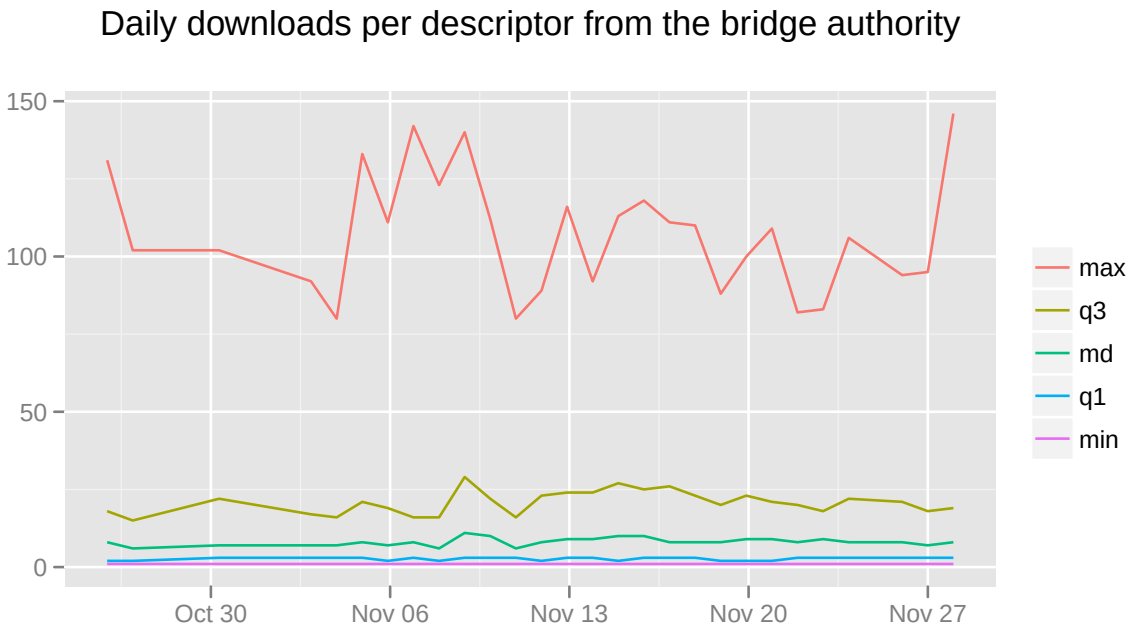

Figure 2: Daily downloads per descriptor from the bridge authority

way, that is by adding some bridges via Vidalia's interface after bridges were discovered via one of our bridge distribution channels. Over the past month, there have been between 3900 and 6600 fetches per day, with a median of 8 fetches per bridge. The most fetched descriptor is fetched up to 350 times per day, indicating that it does indeed belong to a bridge that was given out with a fingerprint and being used by Tor clients. We have gotten some reports that a bundle circulated with pre-configured bridges, and this could account for the many fetches.

Secondly, most bridge descriptors are not even fetched from the authority. This is a clear indication that we can improve our odds of updating bridge clients with current bridge info if we can get them to request the information better.

4 Improving Tor's behaviour for the two user groups

The first group ("Tor is blocked, and some way to reach the network has to be found") is mostly concerned about circumvention, without necessarily hiding that they're using Tor from someone. Typically, access to the Internet is filtered, but circumventing a filter isn't too risky and people are more concerned with access than hiding their tracks from a data-collecting adversary. Speed, bootstrapping performance, and little intervention/maintenance of a setup are the biggest goals.

Adding auto-discovery mechanisms for bridges that changed their IP address will help this group gain a lot more robustness when it comes to maintaining connectivity against an adversary that blocks public relays, but isn't very quick in blocking all bridges. As far as we know, this is currently true for the majority of our bridge userbase.

For the second group ("Tor may or may not be blocked, but the user is trying to hide the fact they're hiding Tor"), precise control over Tor's actions is much more important than constant connectivity, and private bridges might be utilized to that end as well. A user in this group wants to keep the bridges he's using secret, and puts up with frequent updates to the configuration for the added safety of only connecting to a pre-specified IP address:port combination. We can't do very much for a user belonging to this group with regard to bridges, but he will very much benefit from improvements made to our general fingerprintability resistance. Also options like the `DisableNetwork`[4] option (prevent touching the network in any kind of way until this option is changed) that was recently introduced to Tor help him.

Another interesting point here is that we can indirectly improve the behaviour for the first group by not making it too easy to learn about bridges, because censors can use the same data to more effectively block them. This means that we shouldn't, for example, start giving out significantly more bridges to a single user.

We've written a proposal[5] to implement some changes in Tor, to better facilitate the needs of the first group of bridge users.

[4] https://trac.torproject.org/projects/tor/ticket/3644

[5] https://lists.torproject.org/pipermail/tor-dev/2011-November/003097.html