An Analysis of Tor Relay Stability

Karsten Loesing
karsten@torproject.org

Tor Tech Report 2011-06-001
June 30, 2011

1 Introduction

The Tor network consists of around 2,000 relays and 500 bridges run by volunteers, some of which are on dedicated servers and some on laptops or mobile devices. Obviously, we can expect the relays run on dedicated servers to be more "stable" than those on mobile phones. But it is difficult to draw a line between stable and unstable relays. In most cases it depends on the context which relays count as stable:

1. A stable relay that is supposed to be part of a circuit for a *long-running stream* should not go offline during the next day. If only 1 of typically 3 relays in a circuit goes away, the stream collapses and must be rebuilt.

2. A stable relay that clients pick as *entry guard* doesn't have to be running continuously, but should be online most of the time in the upcoming weeks. In addition to being stable, an entry guard should have reasonable bandwidth capacity in order not to slow down clients.

3. A stable relay that acts as *hidden-service directory* should be part of a relay subset that mostly overlaps with the subsets 1, 2, or even 3 hours in the future. That means that the relays in this set should be stable, but also that not too many new relays should join the set at once.

4. A stable relay that clients use in a *fallback consensus* [4] that is already a few days or even weeks old should still be available on the same IP address and port, but doesn't necessarily have to run without interruption.

5. A stable *bridge relay* should be running on the same IP address a few days after a client learns about the bridge and should be available most of the time in the upcoming weeks.

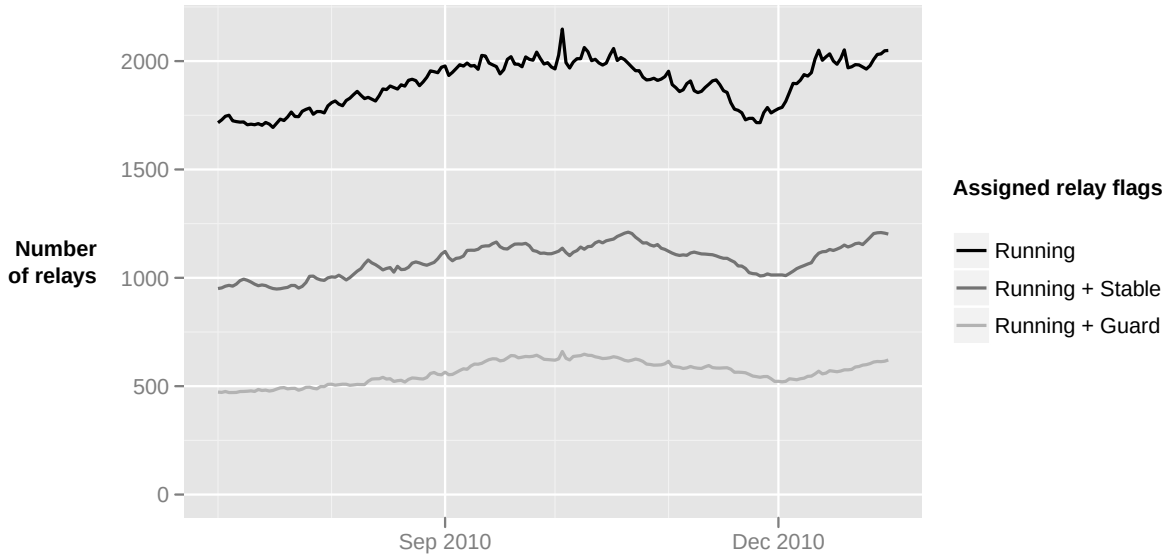

Figure 1: Number of relays with relay flags Running, Stable, and Guard assigned by the directory authorities between July and December 2010

The Tor directory authorities measure relay stability for the first three contexts listed above and assign the Stable, Guard, and HSDir flags that clients use to select relays. Figure 1 shows the number of relays with relay flags assigned between July and December 2010 which will be our analysis interval.

In this analysis, we use a simulator to resemble how the directory authorities assign the Stable and Guard relay flags. This simulator uses archived directory data to decide when a relay was running and when it failed or was restarted. We further use this simulator to calculate future stability metrics that help us evaluate the quality of a given relay flag assignment. We modify the input parameters to Tor's stability metrics to see whether we can improve how the directory authorities should assign relay flags. The analysis data and simulator used in this report are available on the Tor metrics website at https://metrics.torproject.org/.

2 Requirements for Stable and Guard flags

The requirements for assigning the Stable and the Guard flags are defined in the Tor directory protocol specification [1]. These definitions are revised here to better reflect what is implemented in the code:

> "A router is 'Stable' if it is active, and either its Weighted MTBF is at least the median for known active routers or its Weighted MTBF corresponds to at least [5] days. [...] A router is a possible 'Guard' [if it is 'familiar',] if its Weighted Fractional Uptime is at least the median for 'familiar' active routers [or its Weighted Fractional Uptime is at least 98 %], and if its [advertised] bandwidth is at least median [of all active routers] or at least 250KB/s. [...] A node is 'familiar' if 1/8 of all active

nodes have appeared more recently than it, or it has been around for [a weighted time of 8 days]."

These definitions entail four requirements for assigning relay flags, one for the Stable flag and three for the Guard flag:

Weighted Mean Time Between Failure (WMTBF): The WMTBF metric is used to track the length of continuous uptime sessions over time. The (unweighted) MTBF part measures the average uptime between a relay showing up in the network and either leaving or failing. In the weighted form of this metric, which is used here, past sessions are discounted by factor 0.95 every 12 hours. Current uptime sessions are not discounted, so that a WMTBF value of 5 days can be reached after 5 days at the earliest.

Weighted Fractional Uptime (WFU): The WFU metric measures the fraction of uptime of a relay in the past. Similar to WMTBF, WFU values are discounted by factor 0.95 every 12 hours, but in this case including the current uptime session.

Weighted Time: The Weighted Time metric is used to calculate a relay's WFU and to decide whether a relay is around long enough to be considered 'familiar.' The Weighted Time is discounted similar to WMTBF and WFU, so that a Weighted Time of 8 days can be reached after around 16 days at the earliest.

Advertised Bandwidth: The Advertised Bandwidth of a relay is the minimum of its configured average bandwidth rate and the maximum observed bandwidth over any ten second period in the past day. It should be noted that the advertised bandwidth is self-reported by relays and has been misused in the past to attract more traffic than a relay should see. In theory, using the advertised bandwidth value has become discouraged for anything critical.

All four requirements have in common that they consist of a dynamic part that relies on the current network situation (e.g., median of all WMTBF values for the Stable flag) and a static part that is independent of other relays (e.g., WMTBF value of 5 days or higher). The dynamic part ensures that a certain fraction of relays get the Stable and Guard flags assigned even in a rather unstable network. The static part ensures that rather stable (or fast) relays are not denied the flags even when most of the other relays in the network are highly stable (or very fast).

3 Obtaining data for relay stability analysis

The internal data that the directory authorities use to decide whether a relay should get the Stable or Guard flag assigned is not publicly available. Instead, we rely on the network status consensuses and server descriptors that are archived and publicly available since October 2007. The network status consensuses tell us which relays were available with a granularity of 1 hour, and the server descriptors help us figure out whether a relay was restarted in the 1 hour before being listed in a network status. We further learn whether a relay got the Stable and/or Guard flag assigned and what bandwidth it advertised and actually used for relaying data at a given time. In detail, we obtain the following data for this analysis:

Valid-after time: The time when a network status consensus was published.

Fingerprint: A relay's unique public key fingerprint.

Restarted: Was a relay restarted within the last hour before the valid-after time based on its self-reported uptime?

Stable: Did the directory authorities assign the Stable flag to a relay?

Guard: Did the directory authorities assign the Guard flag to a relay?

Advertised bandwidth: A relay's self-reported advertised bandwidth.

Bandwidth history: How many bytes did the relay write on a given day?

The choice for using these archived data instead of, e.g., trying to obtain the internal relay stability data that the directory authorities use has a few advantages and disadvantes. The main disadvantage is that our data has a granularity of one hour and can only detect a single failure or restart within this hour. It's unclear which effect this lack of detail has on less stable relays.

But there are a few advantages of using the archived network data: We have a few years of data available that we can use for the analysis. If we started obtaining original stability data from the directory authorities, we'd have to wait a few months before conducting this analysis. Also, it's still unclear whether the Tor code that assigns relay flags is correct, so it makes sense to use a different data basis. Finally, we need the directory archives anyway to evaluate how stable or fast a relay turned out to be after being selected.

4 Simulating current Stable/Guard assignments

The first step before varying the requirements for assigning relay flags is to run a simulation with the current default requirements. Only if we manage to come up with similar results as the directory authorities, we can hope that our suggested parameter changes will have similar effects in the real Tor network.

Figure 2 shows the fractions of relays that got the Stable and Guard flags assigned by the directory authorities ("observed" lines) and in our simulation with default requirements ("simulated" lines). Ideally, the observed lines should match the simulated lines. The two Guard lines overlap for about half of the observed time interval and the difference is always below 5 % of running relays, which seems to be acceptable. The fraction of Guard relays in the simulation is rather stable at 28 % whereas the observed fraction goes up to 33 % for three months. The observed Stable line seems to be systematically 5 to 10 % higher than the simulated line.

We first investigate the differences by looking in how far the sets of observed and simulated sets overlap. After all, it could be that our simulated 30 % of Guard relays are a completely different subset of relays than the observed 30 %. Figure 3 shows the absolute number of relays in the intersection and the sets of only observed or only simulated relays. This graph shows that the majority of relay assignments overlap for both flags. The sets of Stable relays that are only found in the simulation is rather small with around 50 relays. Compared to that, there are between 100 and 200 relays on average that got the Stable flag assigned by the directory

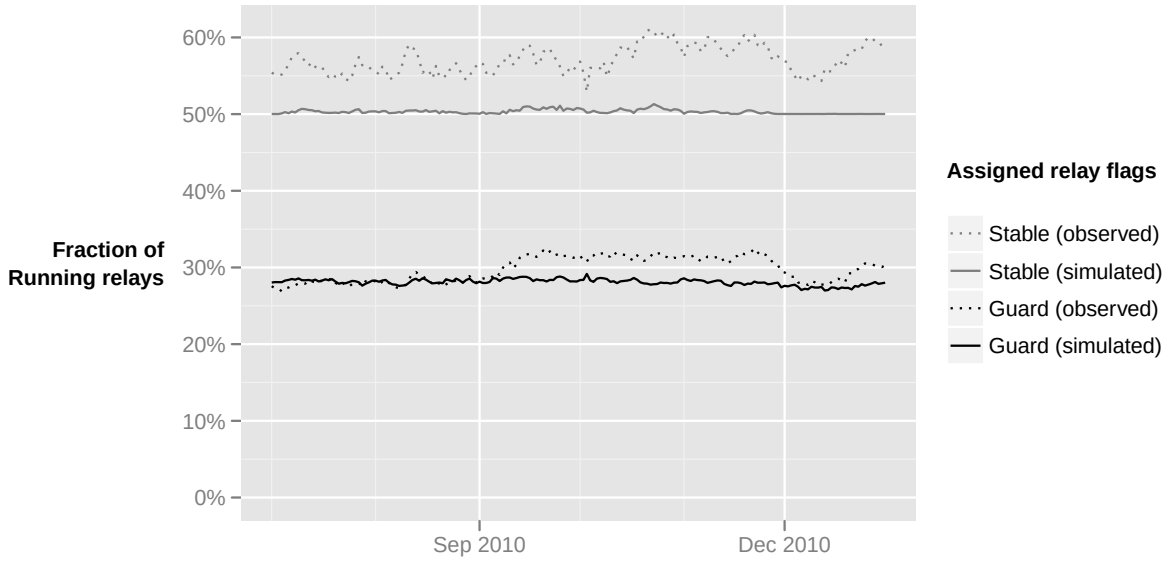

Figure 2: Fraction of relays that got the Stable and Guard flag assigned by the directory authorities and in our simulation with default requirements

authorities but which did not qualify for that flag in the simulation. The situation of assigned Guard flags is somewhat different. In July and August 2010, there are about 50 relays in the only-observed and only-simulated sets. From September 2010 on, there are hardly any relays in the only-simulated set, but the number of relays in the only-observed set goes up to nearly 100.

Before accepting these deviations as a given, we take a quick look at the actual requirements to learn whether the dynamic or the static part of requirements was more relevant. Figure 4 shows the dynamic parts as daily means (dark gray lines) and daily min-max ranges (light gray ribbons) as well as the static parts (dashed lines). The dashed lines act as a limit here: if a dynamic requirement based on the other relays exceeds the dashed line, the requirement is cut to the fixed value and the fraction of relays meeting the requirement increases. Whenever the dynamic requirement remains below the dashed line, the fraction of relays that get flags assigned should remain more or less the same. For example, the top left graph indicates that it's quite possible that more than 50 % of relays got the Stable flag assigned from July to November, but that this fraction should have dropped to 50 % in December 2010. However, this was not the case. The graph also indicates that the 250 KiB/s limit was never reached, so that relays were always selected based on the comparison to the advertised bandwidth of the other relays.

Possible reasons for deviation of simulation and actual assignments by the directory authorities are:

1. The simulator code may contain bugs.
2. Our data with a granularity of 1 hour lack the details for detecting restarts and failures and thereby make it hard to observe uptime sessions correctly.
3. The directory authorities have slightly different views on the network which then get mixed in the voting process.

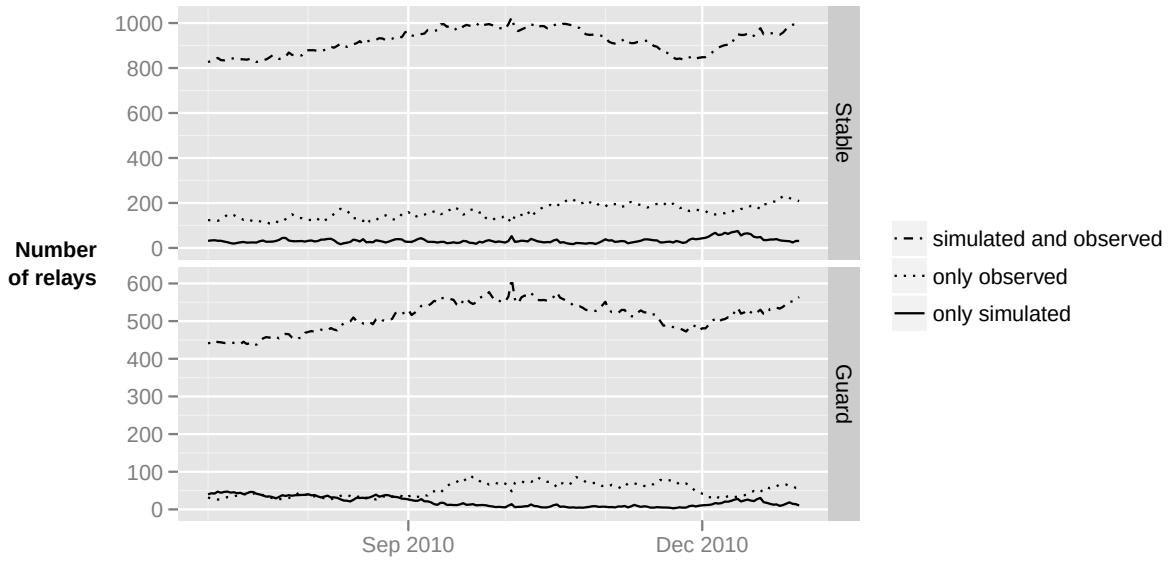

Figure 3: Number of relays that got the Stable and Guard flag assigned by both the directory authorities and by our simulator or by just one of them

4. The implementation of relay history on the directory authorities may contain bugs.

For the remainder of this analysis we will assume that the simulated number of Stable relays is 5 to 10 % lower than in reality and that the simulation of Guard is more or less accurate.

5 Choosing relays for long-lived streams

After learning that our simulation of relay stability is at least not totally off, we want to take a closer look at the Stable flag. Whenever clients request Tor to open a long-lived stream, Tor should try to pick only those relays for the circuit that are not likely to disappear shortly after. If only a single relay in the circuit fails, the stream collapses and a new circuit needs to be built. Depending on how well the application handles connection failures this may impact usability significantly.

The WMTBF metric as the requirement for assigning the Stable flag has already been discussed above. Now we want to find out how useful WMTBF is to predict which relays are not likely to leave or crash in the near future. In order to evaluate future relay stability we measure the *time until next failure*. There is no need to measure more than the current uptime session length, and hence there is no need to weight measurements. For a given set of relays with the Stable flag we determine the 10th percentile of time until next failure. This is the time when 10 % of relays have failed and 90 % are still running. Under the (grossly simplified) assumption that relays are chosen uniformly, $1 - 0.9^3 = 27.1$ % of streams using relays from this set would have been interrupted up to this point.

The default requirement is that a relay's WMTBF is at least the median for known relays or at least 5 days. We simulate times until next failure for the 30th, 40th, 50th, 60th, and 70th

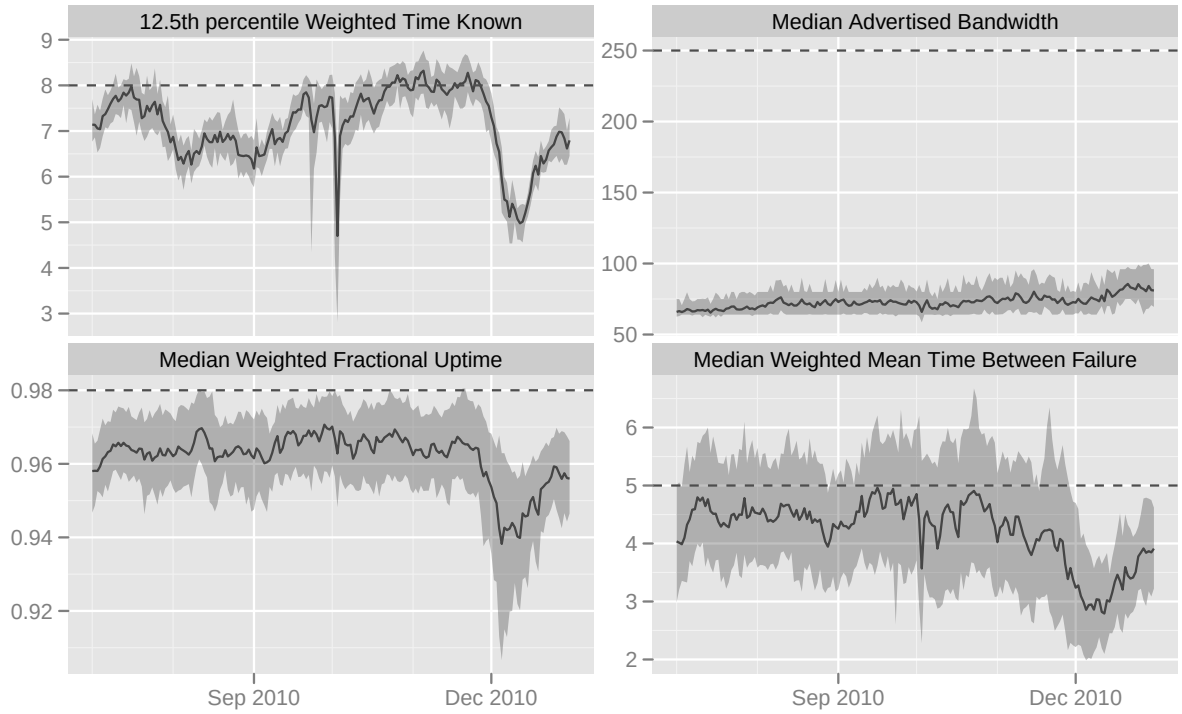

Figure 4: Simulated requirements for assigning the Stable and Guard flags

percentiles of WMTBF values in the network while leaving the fixed 5-day constant unchanged. We also look up times until next failure for the set of Stable relays that got their flags from the directory authorities. Figure 5 shows the results. The artificial steps come from our limitation to measure time until next failure in more detail than in full hours. The further right a line is the higher is the time until next failure for the considered consensuses from July to December 2010.

The first finding that comes to mind is that the observed line is much farther left than the simulated 50th percentile line. In theory, both lines use the 50th percentile as requirement and should therefore have similar results. However, since the directory authorities assign the Stable flag to almost 60 % of all relays, it's likely that their overall stability is lower than the stability of the most stable 50 % of all relays. It's unclear why observed results are even worse than those from the simulated 40th percentile. It might be that the directory authorities don't just assign the Stable flag to too many relays, but also to the wrong relays.

Apart from that, the graph shows the large influence of different WMTBF percentiles on relay stability. If we lowered the requirement to the 30th WMTBF percentile, thus assigning the Stable flag to at least 70 % of all relays, the 10th percentile of time until next failure would be reached after 6 to 24 hours in most cases. The 40th and 50th percentile requirements move this point to 12 to 36 and 18 to 54 hours, respectively. If we changed the requirement to the 60th WMTBF percentile, this point would only be reached after 24 to 60 hours. The 70th percentile requirement is not even shown in the graph, because this requirement is so high that the fixed 5-day constant applies in most cases here, making the dynamic percentile requirement

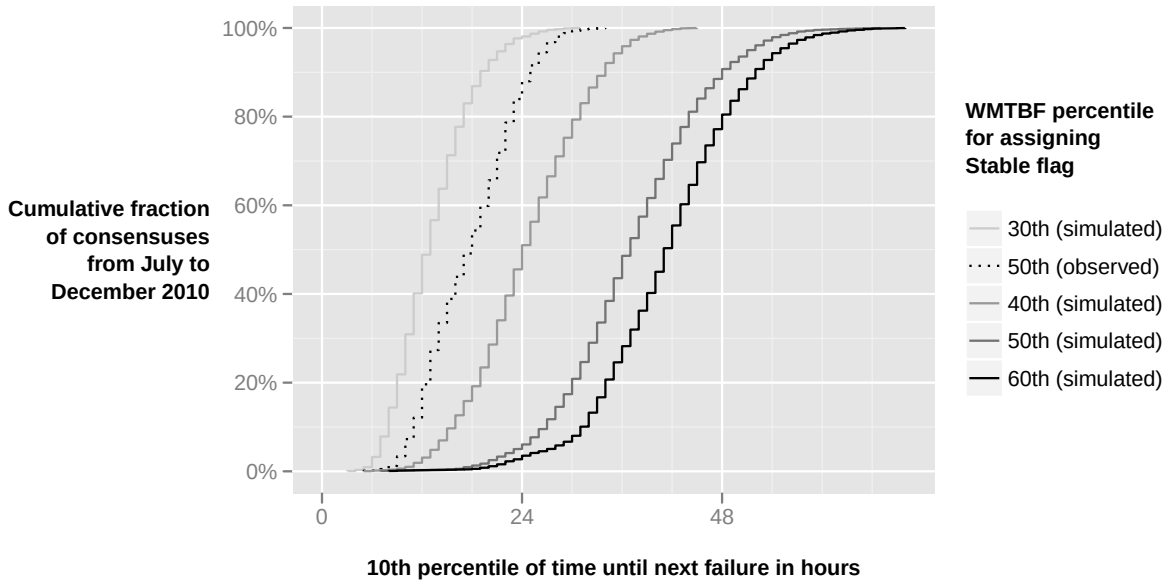

Figure 5: Impact of changing the WMTBF percentile requirement for assigning the Stable flag on the expected time until next failure

meaningless.

As an intermediate conclusion, we could improve relay stability a lot by raising the WMTBF percentile requirement a bit. A good next step might be to find out why the directory authorities assign the Stable flag to 55 to 60 % of all relays and not 50 %. Of course, a higher requirement would result in fewer relays with the Stable flag. But having 40 % of relays with that flag should still provide for enough diversity.

6 Picking stable entry guards

The second flag that we're going to analyze in more detail is the Guard flag. Clients pick a set of entry guards as fixed entry points into the Tor network. Optimally, clients should be able to stick with their choice for a few weeks. While it is not required for all their entry guards to be running all the time, at least a subset of them should be running, or the client needs to pick a new set.

As discussed above, Tor's primary metric for deciding which relays are stable enough to be entry guards is *weighted fractional uptime (WFU)*. WFU measures the fraction of uptime of a relay in the past with older observations weighted to count less. The assumption is that a relay that was available most of the time in the past will also be available most of the time in the future.

In a first analysis we simulate the effect of varying the percentile requirements for becoming an entry guard on the relay stability in the future. We measure future stability by using the same WFU metric, but for uptime in the future. We similarly weight observations farther in the future less than observations in the near future. The rationale is that a high fractional uptime

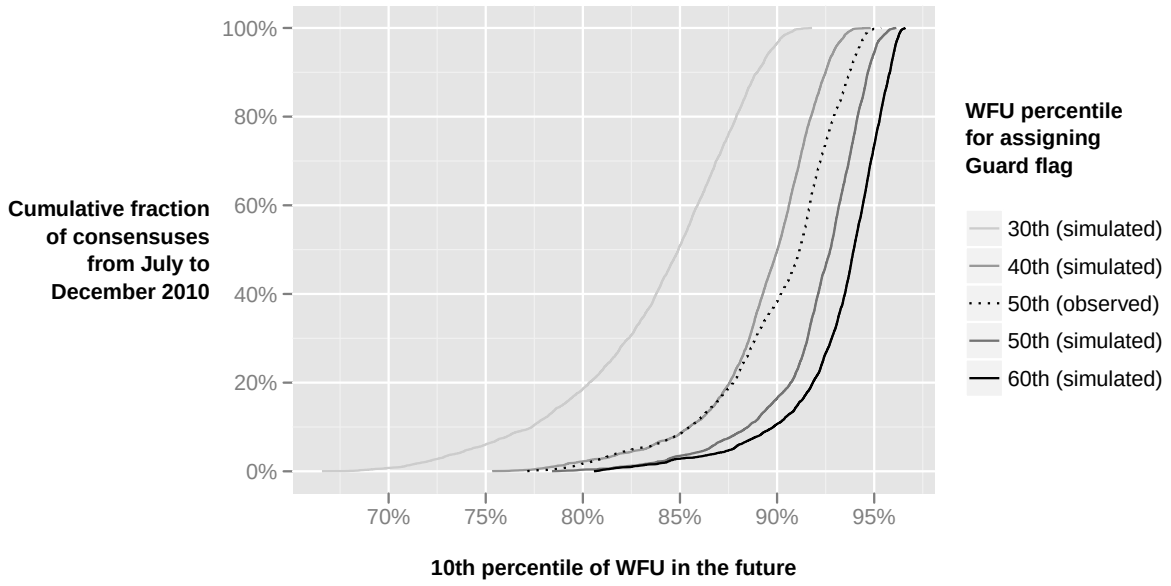

Figure 6: Impact of changing the WFU percentile requirement for assigning the Guard flag on WFU in the future

in the next few days is slightly more important than in a month. For a given set of relays we measure the 10th percentile of WFU in the future as an indication of relay stability. The result is that 10 % of relays will have a lower uptime than this value and 90 % of relays will have a higher uptime.

Figure 6 shows the 10th percentile of WFU for simulations using the 30th, 40th, 50th, and 60th WFU percentile as requirement for assigning the Guard flag. This graph also shows the future WFU of relays that got their Guard flag assigned by the directory authorities. Here, the simulation using the default 50th percentile is much closer to the flags assigned by the directory authorities than in the case of the Stable flag. Unsurprisingly, the 30th percentile requirement has the worst stability, because it includes up to 70% of all relays, minus the non-familiar ones and those that don't meet the bandwidth requirement. Relay stability increases for raising the WFU requirement to the 40th, 50th, and 60th percentile, but in most cases the outcome is an uptime of more than 85 % or even more than 90 %. Under the assumption that a client needs only one or two working guards at a time and can pick a new set of guards easily, these stabilities seem to be sufficiently high.

7 Selecting high-bandwidth entry guards

A second question regarding Guard flag assignments is whether we can raise the advertised bandwidth requirement to end up with faster entry guards. The fixed set of entry guards determines to a certain extent what overall performance the client can expect. If a client is unlucky and picks only slow guards, the overall Tor performance can be bad, in particular because clients don't drop slow guards, but only failing ones.

We introduce a new metric to evaluate how much bandwidth capacity a relay will provide in

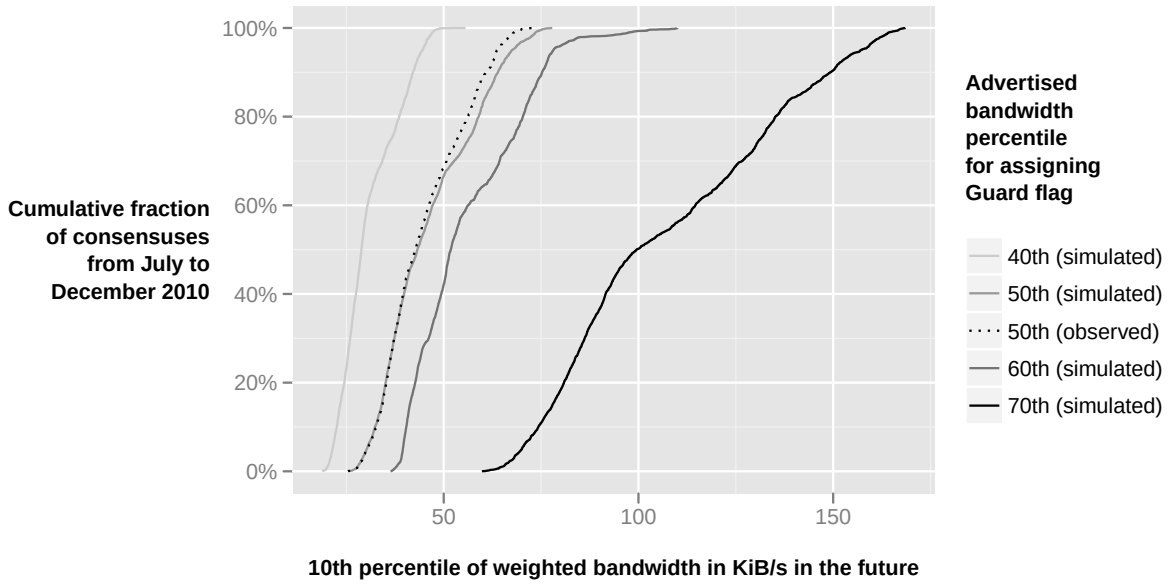

Figure 7: Impact of changing the advertised bandwidth percentile for assigning the Guard flag on bandwidth capacity in the future

the future: the *weighted bandwidth*. This metric is not based on a relay's advertised bandwidth, but on the actual written bytes as reported by the relay. Again, we're more interested in the bandwidth in the near future, so we discount observations 12 hours further in the future by factor 0.95.

Figure 7 shows the effect of changing the advertised bandwidth requirement from the 50th percentile to the 40th, 60th, or even 70th percentile on the 10th percentile of weighted bandwidth. Similar to the graphs above, 10 % of relays have a lower weighted bandwidth and 90 % have a higher weighted bandwidth. Here, the observed 50th percentile is almost identical to the simulated 50th percentile. The graph shows that raising the requirement to the 60th percentile of advertised bandwidth would shift this percentile line by roughly 10 KiB/s to the right. The 70th percentile would ensure that 90 % of the selected Guard relays have a weighted bandwidth of at least between 60 and 170 KiB/s depending on the current network situation.

Of course, raising the advertised bandwidth requirement for becoming a guard node results in having fewer possible guard nodes. Figure 8 shows the effect of raising the advertised bandwidth requirement from the 50th to the 60th or 70th percentile. The 60th percentile requirement would reduce the fraction of relays with the Guard flag from 28 % to around 25 %, and the 70th percentile even to a value below 20 %. There's a clear trade-off here between raising the bandwidth capacity of entry guards and having fewer guards to distribute one third of the network load to. Having fewer than 20 % of all relays being possible Guard nodes is probably not enough and will generate new performance problems.

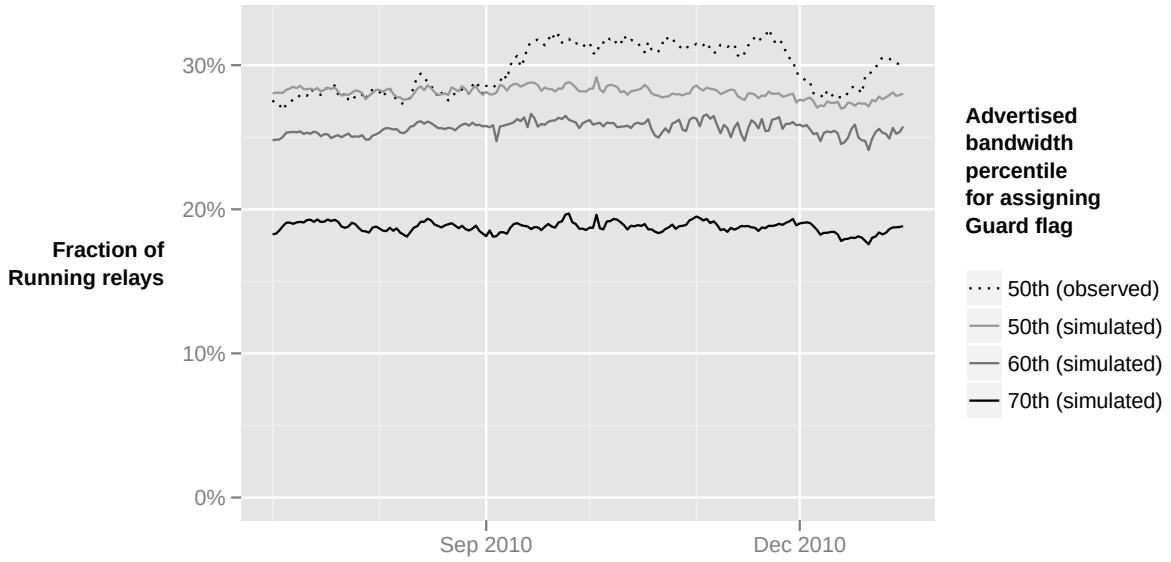

Figure 8: Influence of changing the advertised bandwidth percentile on the fraction of relays getting the Guard flag assigned

8 Discussion and future work

In this report we used a simulator to evaluate Tor's relay stability metrics for assigning Stable and Guard flags to relays. We introduced three metrics to evaluate how stable or fast a set of relays is that got the Stable or Guard flag assigned. Our simulator uses the same algorithms to decide whether a relay is stable as the directory authorities and can be parameterized to analyze different requirements. We could further add new algorithms to the simulator and see what subsets of Stable and Guard relays that would produce.

Using our simulator we found that the fraction of relays with the Stable flag in the current network is higher than it probably should be. We also learned that the WFU requirements for assigning the Guard flag are quite reasonable and lead to stable guards. But we might consider raising the advertised bandwidth requirement a bit to have higher-bandwidth guard nodes. Medium-term, we should get rid of a requirement that is based on the self-reported advertised bandwidth.

Possible next steps are to review the Tor source code for assigning flags and compare the internal relay stability data from the directory authorities to simulated values. It would be interesting to know why the directory authorities assign the Stable flag so generously. Also, it would be interesting to compare the internal stability data from multiple directory authorities to see in how far they agree. Another possible next step might be to turn the four requirement percentiles (WMTBF, WFU, Weighted Time, and Advertises Bandwidth) into consensus parameters to be able to test parameter changes in the real network.

We also left the analysis of relay stability for hidden-service directories, fallback consensuses, and bridge relays as future work. Possible starting points are an earlier analysis of hidden-service directory stability [2], the Tor proposal describing fallback consensuses [4], and a tech

report explaining how bridge descriptors are sanitized to make them publicly available [3].

References

[1] Roger Dingledine and Nick Mathewson. Tor directory protocol, version 3. https://gitweb.torproject.org/tor.git/blob_plain/HEAD:/doc/spec/dir-spec.txt.

[2] Karsten Loesing. *Privacy-enhancing Technologies for Private Services*. PhD thesis, University of Bamberg, May 2009. http://www.opus-bayern.de/uni-bamberg/volltexte/2009/183/pdf/loesingopusneu.pdf.

[3] Karsten Loesing. Overview of statistical data in the Tor network. Technical Report 2011-03-001, The Tor Project, March 2011.

[4] Nick Mathewson. Add new flag to reflect long-term stability. https://gitweb.torproject.org/torspec.git/blob_plain/HEAD:/proposals/146-long-term-stability.txt.