

# Discussion of Possible Design Changes to Improve Tor Hidden Service Performance

Karsten Loesing and Christian Wilms\*

July 15, 2008

## Abstract

This document describes possible design changes aiming at improving the performance of Tor Hidden Services. The ideas for these improvements come from an earlier analysis of possible performance bottlenecks<sup>1</sup>. All sections start with the design idea as stated in the previous analysis and evaluate the possible performance gain and possibly unwanted side-effects. While the first 4 design changes aim at improving connection establishment, the latter 3 affect the process of making a hidden service available in the network. Optimally, the results of the discussion of these ideas will be included in a Tor proposal by August 15, 2008.

## 1 Rendezvous Protocol Simplifications

*Overlier and Syverson proposed two simplified rendezvous protocols.<sup>2</sup> Their first protocol aims at using a single circuit on client-side to contain the rendezvous point and connect to an introduction point. The second protocol goes the extra mile to unite the roles of introduction point and rendezvous point and save another circuit. It can be assumed that these simplifications improve connection establishment times. However, they have yet unclear effects on anonymity, given that they are implemented without the authors' proposed valet node approach which requires a major redesign of hidden services.*

A possible design of the second protocol that combines the roles of introduction and rendezvous point is described in proposal 142<sup>3</sup> and in a revised rendezvous specification<sup>4</sup>. Applying the new design would have the following effects:

---

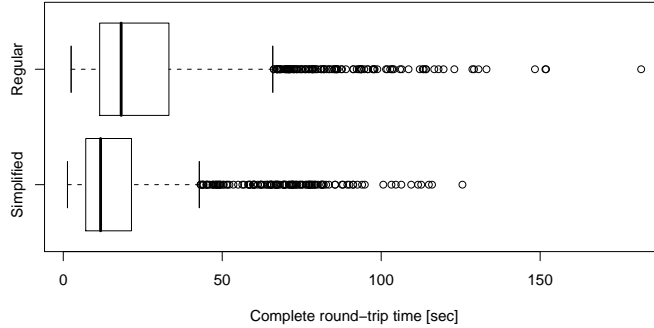
\*Please direct questions and comments either to [tor-assistants@torproject.org](mailto:tor-assistants@torproject.org) or [or-dev@freehaven.net](mailto:or-dev@freehaven.net).

<sup>1</sup><http://freehaven.net/~karsten/hidserv/perfanalysis-2008-06-15.pdf>

<sup>2</sup>Lasse Overlier and Paul Syverson, Improving Efficiency and Simplicity of Tor Circuit Establishment and Hidden Services, <http://www.freehaven.net/anonbib/#overlier-pet2007>

<sup>3</sup><https://tor-svn.freehaven.net/svn/tor/trunk/doc/spec/proposals/142-combine-intro-and-rend-points.txt>

<sup>4</sup><https://tor-svn.freehaven.net/svn/tor/branches/121-hs-authorization/doc/spec/rend-spec-v3draft.txt>



Protocol	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Regular	2.479	11.450	18.210	27.370	33.220	181.800
Simplified	1.324	7.070	11.740	19.130	21.450	125.600

Figure 1: Connection establishment times for combining introduction and rendezvous points

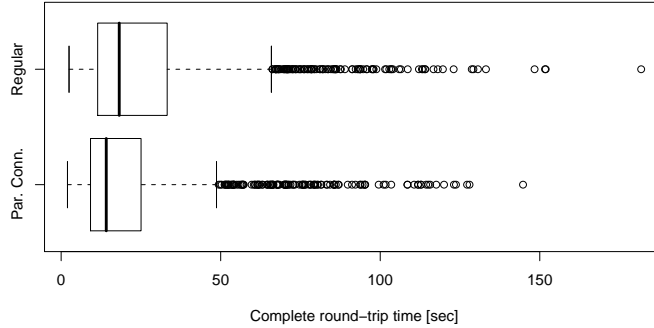
**Performance Improvement of 8.2 Seconds** A pre-evaluation<sup>5</sup> of this design change has shown that mean connection establishment times can be reduced from 27.4 seconds in the original protocol to 19.1 seconds in the changed design. Figure 1 shows a comparison of connection establishment times.

**Responsibility of Serving a Hidden Service** One of the original reasons for the separation of introduction and rendezvous points is that a relay shall not be made responsible for relaying data on behalf of a certain hidden service. The changed design needs to make sure that a combined introduction and rendezvous points cannot learn easily on behalf of which hidden service it is working.

**Scalability** On the one hand, the simplified protocol removes the need for a hidden service to establish new circuits on demand. On the other hand, the service needs to maintain a number of open circuits proportional to the expected number of client requests.

**Attack Resistance** An attacker who can take down a combined introduction and rendezvous point does not only eliminate an access point to the hidden service, but also breaks current client connections to the hidden service using that contact point.

<sup>5</sup>Christian Wilms, Improving the Tor Hidden Service Protocol Aiming at Better Performance, diploma thesis, June 2008



Protocol	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Regular	2.479	11.450	18.210	27.370	33.220	181.800
Par. Conn.	1.987	9.212	14.140	22.450	24.990	144.800

Figure 2: Connection establishment times for parallel connections to introduction points

## 2 Parallel Connections to Introduction Points

*A client could attempt to establish two introduction circuits to two different introduction points simultaneously and only use the first that succeeds. The slower circuit could still be used for another purpose. However, there is a possible anonymity issue here that needs to be taken under consideration, because the circuit can now be linked to one of the hidden service’s introduction points. Further, it needs to be evaluated whether the extension of two circuits at the same time has a negative effect on clients with low bandwidth.*

**Performance Improvement of 4.9 Seconds** A pre-evaluation resulted in a possible reduction of connection establishment times from 27.4 seconds in the original protocol to 22.5 seconds in the changed design. Figure 2 shows the possible improvement of connection establishment times.

**More Internal Circuits Required** Clients need to maintain a higher number of internal circuits in order to use two parallel circuits to connect to an introduction point. This number should have a fixed lower bound that is by 1 higher than the current limit and could be increased based on the history of hidden service accesses.

**Increase of Overall Network Load** If the slower circuit is discarded after the first introduction point has been established, there will be an additional load on the network. In particular,  $3 + 1 = 4$  circuit extensions will be lost.

**Dealing with Loser Circuits** If slower circuits are not discarded, they can be kept in a pool for later internal circuit usage. However, in addition to being

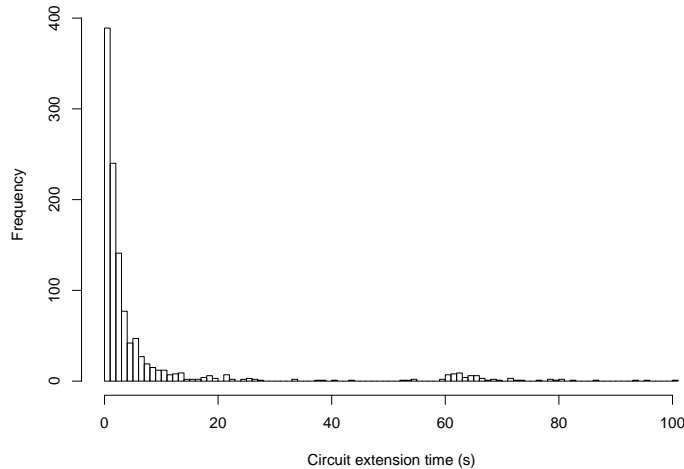


Figure 3: Client-side introduction circuit establishment times

slow anyway, these circuits might need to be extended for another usage which will make them even slower. Therefore, they should be discarded after they exceed a certain length, e.g. 6 hops.

**Anonymity Issues with Loser Circuits** Other than stated in the analysis, there should not be any anonymity issues with slower circuits, because they were never told to be used as introduction circuits.

### 3 Individual Cannibalization Timeouts

An idea that was not described in the last report is to introduce individual timeouts for cannibalizing a circuit and extending it to an introduction point afterwards on client side. The current timeout for this task is 60 seconds. Assuming that the timeout would be reduced to a lower value, a second (or third) attempt to cannibalize and extend a circuit would be started earlier. Figure 3 contains a histogram of times that are required for this task. Here, the current timeout of 60 seconds becomes visible at the second peak of values shortly after 60 seconds.

**Performance Improvement of 2 Seconds** Figure 4 shows simulated mean introduction circuit cannibalization and extension times for timeouts between 10 and 60 seconds. For a timeout of 30 seconds the performance gain would be approximately 2 seconds in the mean as opposed to the current timeout of 60 seconds.

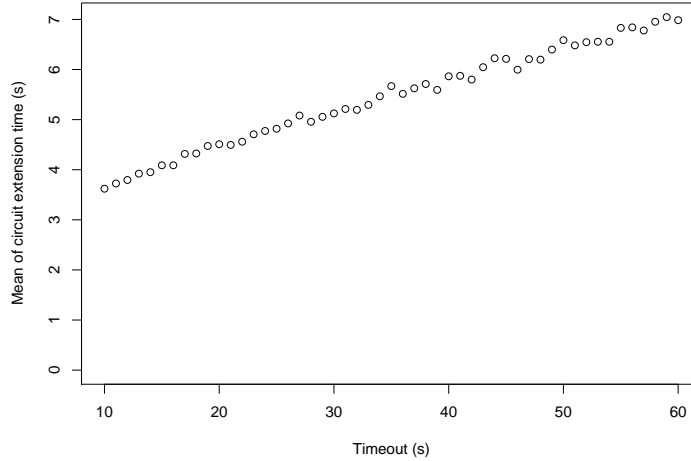


Figure 4: Mean client-side introduction circuit establishment times as a function of timeout

**Effect on Low-Bandwidth Clients** All measurements so far have been performed on high-bandwidth clients. If there is a significant influence of bandwidth on cannibalization time, relays should keep a history of past circuit establishment times and adapt their individual timeouts by applying exponential smoothing on historical values.

**Increased Network Load** There will be an increased share of circuits that are discarded after the new timeout expires, but would have been completed within the existing timeout of 60 seconds. Figure 5 visualizes the simulated increased number of circuit cannibalization and extension attempts for lower timeouts.

## 4 Increase Count of Internal Circuits

*The number of preemptively built internal circuits for later cannibalization should be increased. Really popular hidden services require more than two internal circuits in the pool to answer multiple client requests at the same time. This scenario was not yet analyzed, but will probably exhibit even worse performance as measured here. The number of preemptively built internal circuits should be a function of connection requests in the past to adapt to changing needs. Furthermore, an increased number of internal circuits on client side would allow clients to establish connections to more than one hidden service at a time.*

**Performance Improvement of 4.7 Seconds** It is assumed that a popular

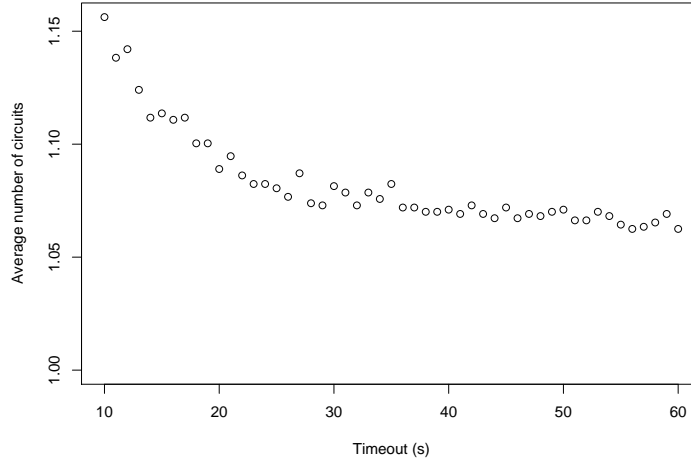


Figure 5: Number of client-side introduction circuit establishment attempts as a function of timeout

hidden service cannot make use of cannibalization for connecting to rendezvous points, but needs to create new circuits. Therefore, the circuit creation time needs to be added to the current results. An evaluation of internal circuit creating times is shown in the table below. In the mean the connection establishment time to a popular hidden service would increase by 4.7 seconds.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.139	1.203	2.504	4.659	5.234	56.090

**Dependency on Regular Protocol** This design change only makes sense if the design change described in Section 1 is *not* applied.

**Higher Number of Internal Circuits** A hidden service would have to maintain a higher number of internal circuits to respond to client requests. Each service needs to maintain a history of client requests to have a good estimate for the number of required internal circuits.

## 5 Build More Introduction Circuits

*When establishing introduction points, a hidden service could launch 5 instead of 3 introduction circuits at the same time and use only the first 3 that could be established. The remaining two circuits could still be used for other purposes afterwards.*

**Performance Improvement of 3.7 Seconds** The effect of choosing the 3 fastest out of 5 (4, 6) circuit creations as introduction points has been simulated using previously measured data, too. Therefore, circuit establishment times were derived from log files and written to an array. Afterwards, a simulation with 10,000 runs was performed picking 5 (4, 6) random values and using the 3 lowest values in contrast to picking only 3 values at random. The result is that the mean time of the 3-out-of-3 approach is 8.1 seconds, while the mean time of the 3-out-of-5 approach is 4.4 seconds. The results are given in the table below.

Intro. Pnts.	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3	1.023	5.017	6.820	8.060	9.718	40.490
4	0.882	3.811	4.997	5.421	6.472	30.950
5	0.786	3.211	4.188	4.430	5.367	19.230
6	0.578	2.826	3.702	3.865	4.705	12.080

**Network Load** The effect on network load is minimal, because the hidden service can reuse the slower internal circuits for other purposes. The only change is that a hidden service starts establishing more circuits at once instead of subsequently doing so.

## 6 Descriptor Upload Timing

*The choice to wait for 30 seconds for a service to have a stable set of introduction points is rather arbitrary. An analysis of typical delays in establishing introduction points might help to apply a more suitable algorithm here.*

When finding a useful algorithm for uploading rendezvous descriptors, there are two conflictive objectives: a) upload descriptors as early as possible and b) avoid uploading descriptors that need to be replaced shortly after. An evaluation of introduction point establishments and abandonings within the first 30 minutes of providing a hidden service can help determining the optimal delay. One can determine the number of uploaded descriptors from empirical data for every possible delay.

**Descriptor Publication Delay** Figure 6 shows the delays in descriptor publication for stabilization times between 1 and 90 seconds.

**Number of Published Descriptors** Figure 7 displays the mean number of published descriptors. These results show that there is a clear trade-off between the two objectives for delays lower than 50 seconds. A delay of 60 seconds significantly reduces the mean number of published descriptors, but at the price of almost doubling the mean upload time of the first descriptor as compared to the current choice of 30 seconds.

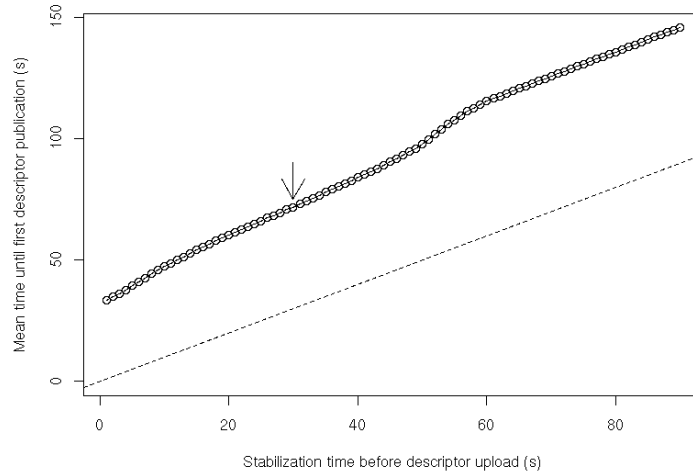


Figure 6: Time until first descriptor upload as a function of stabilization time (arrow denotes current stabilization time of 30 seconds)

## 7 Descriptor Upload Failures

*The current logic to upload rendezvous service descriptors does not handle failures in a reasonable way. In case of a failure, Tor waits for a solid hour before making the next attempt. There should either be a smaller timeout or an individual handling of failures per directory.*

This bug is described in detail in task 767<sup>6</sup>. The fix, however, requires on-demand downloading of router descriptors, which might have anonymity issues. Therefore, this bugfix requires discussion of the possible consequences.

**Higher Reliability** An evaluation of log statements has shown that 14.7% of all descriptor uploads to the three central hidden service authorities failed.

A subsequent analysis has confirmed these results for the distributed storage for hidden service descriptors with a failure rate of 13.7%.

**Anonymity Issue** A hidden service needs to download a specific set of router descriptors in order to store its hidden service descriptor. This might reveal the hidden service identity to everyone observing a request for router descriptors.

<sup>6</sup><http://bugs.noreply.org/flyspray/?do=details&id=767>

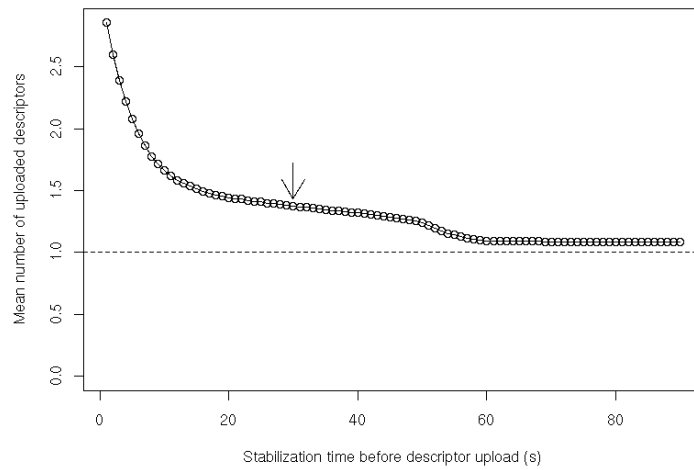


Figure 7: Number of uploaded descriptors as a function of stabilization time (arrow denotes current stabilization time of 30 seconds)