# Tor growth rates and improving Torperf throughput

Virgil Griffith
i@virgil.gr

Tor Tech Report 2014-10-001
October 4, 2014

## 1 Preliminaries

Despite data being available from metrics.torproject.org for sometime, there's been little statistical analysis of that data. Let's fix that. From the Metrics data, the most obvious thing to plot is the number of relays over time, see Figure 1. Plotting in logscale (so a straight line means exponential growth) reveals that the number of relays increases exponentially. Good to know. The "stable relays" are plotted in purple because they are fabulous.

Next in Figure 2 we chart the total network bandwidth over time. Tor's total network bandwidth doubles at a darn impressive 13–14 months! Moore's Law, doubling every 18 months, is downright torpid by comparison.

Since 2010 the doubling rates for both relays and bandwidth have been remarkably consistent. Although recognizing that there are unaccounted for sinusoidal trends, the fact remains that a simple fit of $y = m \ \log(x) + b$ accounts for $\sim$90% of the variance! Additionally, the 99% confidence intervals on the predicted data are barely visible without a magnifying glass. Extrapolation from statistics is a dangerous game, but realistically we can't expect these growth rates to be more predictable. With this statistical bedrock under our feet, let's go deeper. In Figure 3 we see how the mean relay bandwidth grows over time. We see that the mean relay bandwidth doubles about every two years. This is akin to Nielsen's Law which states that for high-end home users, bandwidth doubles every two years. Good job operators—those Tor-shirts are well earned!

We see that the mean relay bandwidth increases by Nielsen's Law, but how does this impact client experience? Fortunately, we have Torperf data to answer this. Simple things first, and in Figure 4 we plot Torperf bandwidth over time. Torperf's fitted line isn't nearly as good a fit as the number of relays or total bandwidth (Figures 1 and 2), but it conveys enough of the trend to be useful. We see that, depending on file size, Torperf throughput doubles every 25–35 months.[1] Given such a wide spread in Figure Figure 4, we will separately consider the Torperf bandwidth for downloading a 50 KiB and 5 MiB file. Lets go deeper.

Absolute Torperf improvements are great to see, but the key measure is how Torperf throughput compares with clients' non-Tor throughput. From OOKLA bandwidth data we calculate the composite mean download rate for the three countries with the greatest number of Tor clients: United States, Germany, and Russia (Figure 7). With the composite non-Tor bandwidth in hand, we plot Torperf bandwidth normalized (divided) by the composite non-Tor bandwidth arriving at Figure 5.

For smaller files (50 KiB), we see that although absolute Torperf has been doubling every 35 months, normalized Torperf has been essentially flat. For larger files (5 MiB), we see a gradual uptick in normalized Torperf.

From the doubling rates of Torperf and composite non-Tor bandwidth we can derive the normalized Torperf growth rates analytically. Taking the ratio of two exponentials of the form $y = 2^{(1/n)x}$ where $n$ is the doubling rate, we get $y = 2^{(1/n-1/m)x}$ where $n$ and $m$ are the doubling rates of Torperf bandwidth and composite non-Tor bandwidth respectively. This results in normalized Torperf doubling every 20 years for small files and doubling every 5 years for large files. To put a five year doubling rate in perspective, this means Torperf will reach 5% of non-Tor bandwidth around year 2022. Internal optimizations like the KIST scheduler are great steps to improve this.

## 2 Will adding advertised bandwidth improve Torperf?

There have been various proposals for improving client speeds by adding operator incentives beyond the established T-shirts and financial grants. Our final analysis is an attempt to predict whether adding more advertised relay bandwidth would reliably improve Torperf throughput.

We've established that absolute Torperf improves on its own due to the increasing bandwidth of relays. Our first step to blunt the influence of increasing relay bandwidth is to always look at the normalized Torperf performance. We explored several different predictors of normalized Torperf, and the most promising was proportion of total read bandwidth to total advertised bandwidth, or the Network Utilization Ratio (NUR). We plot normalized Torperf as a function of NUR in Figure 6.

We see that NUR doesn't predict much of the normalized bandwidth for small (50 KiB) files. However, for large files (5 MiB), there's a fuzzy yet definite trend of "lower NUR means higher normalized Torperf". But there's a risk, we see that the lowest NUR data points (purple) are all from 2014. Therefore NUR could be acting as a mere proxy for the gradual (yet slow per Figure 5) improvement of normalized Torperf over time.

We control for this using a two-factor ANOVA using `DATE` and `NUR` as the two factors and normalized Torperf as the dependent variable. For the stats-literate, the full ANOVA tables are given in Table 1, but the take-home message is that `NUR` provides substantial predictive

[1] It's not obvious that Torperf bandwidth increases exponentially, but given that bandwidth and CPU are the primary factors in Torperf and that each of these follow their respective exponential curves, it's reasonable to err on the side of an exponential fit over a linear one. Statistical modeling often leverages domain knowledge.

power for normalized Torperf even after accounting for DATE. Concretely, while the single-factor model using DATE has an $r^2$ of 0.02 (50 KiB) and 0.14 (5 MiB), the two-factor model using DATE and NUR yields an $r^2$ of 0.17 and 0.44—a 750% and 208% improvement respectively. This allows us to tentatively conclude that a sudden uptick in advertised bandwidth would improve normalized Torperf beyond the glacial ascent seen in Figure 5.[2]

| | df | Sum Sq | Mean Sq | F-value | p-value |
|---|---|---|---|---|---|
| DATE | 1 | 0.04039 | 0.040390 | 38.418 | 7.309E$^{-10}$ |
| NUR | 1 | 0.29005 | 0.290045 | 275.887 | 2E$^{-16}$ |
| Residuals | 1546 | 1.62534 | 0.001051 | | |

(a) 50 KiB. For aggregate model $r^2 = 0.17$.

| | df | Sum Sq | Mean Sq | F-value | p-value |
|---|---|---|---|---|---|
| DATE | 1 | 27.395 | 27.395 | 401.09 | 2E$^{-16}$ |
| NUR | 1 | 56.750 | 56.750 | 830.87 | 2E$^{-16}$ |
| Residuals | 1546 | 105.595 | 0.068 | | |

(b) 5 MiB. For aggregate model $r^2 = 0.44$.

Table 1: ANOVA tables predicting normalized Torperf for downloading a 50 KiB and 5 MiB file.

## 3 Summary

We've learned a few things.

1. Many aspects of Tor follow exponential growth. Table 2 summarizes these results. Additionally, Tor bandwidth currently sits at < 2% of mean non-Tor bandwidth.

2. Tor clients' absolute throughput is steadily improving. However, after normalizing by mean non-Tor bandwidth, this improvement is greatly diminished. For small files, normalized Torperf has been essentially flat since records have been kept.

3. An intervention to increase advertised bandwidth would noticeably improve normalized Torperf for large as well as small files.

[2] Unsurprisingly, there's some caveats to this conclusion. Our argument presumes that the distribution of advertised bandwidth across relays is constant—for example, Torperf would not improve if $10^{12}$ new relays joined the consensus but each provided only 1 B/s. We're aware of no evidence indicating this assumption is unrealistic.

|  | Doubling rate | $r^2$ |
|---|---|---|
| Total advertised bandwidth | 1.2 years | 0.96 |
| Mean relay bandwidth | 2 years | 0.91 |
| Number of relays (all) | 3 years | 0.94 |
| Absolute Torperf (5 MiB) | 2 years | 0.46 |
| Absolute Torperf (50 KiB) | 3 years | 0.55 |
| Mean RU download bandwidth | 3.1 years | 0.95 |
| Mean US download bandwidth | 3.4 years | 0.97 |
| Mean DE download bandwidth | 3.9 years | 0.88 |
| Composite download bandwidth | 3.5 years | 0.97 |
| Normalized Torperf (5 MiB) | 5 years | - |
| Normalized Torperf (50 KiB) | 19.9 years | - |

Table 2: Summary of growth rates

# 4 Future Work

Some natural extensions to this work are:

- Instead of looking at the mean relay bandwidth, instead separately calculate the expected bandwidth for the guard, middle, and exit node positions.

- It'd be nice to characterize the distribution of advertised bandwidth. Does it follow a Gaussian? Pareto? It'd be nice to know.

- When computing the composite non-Tor bandwidth, instead of doing an unweighted average of the United Staes, Germany, and Russia, it'd be better to do a weighted average among all countries in which each country is weighted by its number of originating Tor clients. We doubt this would change the conclusions.

- Tor's Network Utilization Ratio (NUR), shown in Figure 8, has clear drops of unclear cause. Given how predictive NUR is of normalized Torperf, we'd like to know the causes of the two drops in NUR on 2013-10-09 and 2014-06-06.

**Acknowledgements.** We thank Roger Dingledine and Karsten Loesing for their help and review. All analyses were done in R.

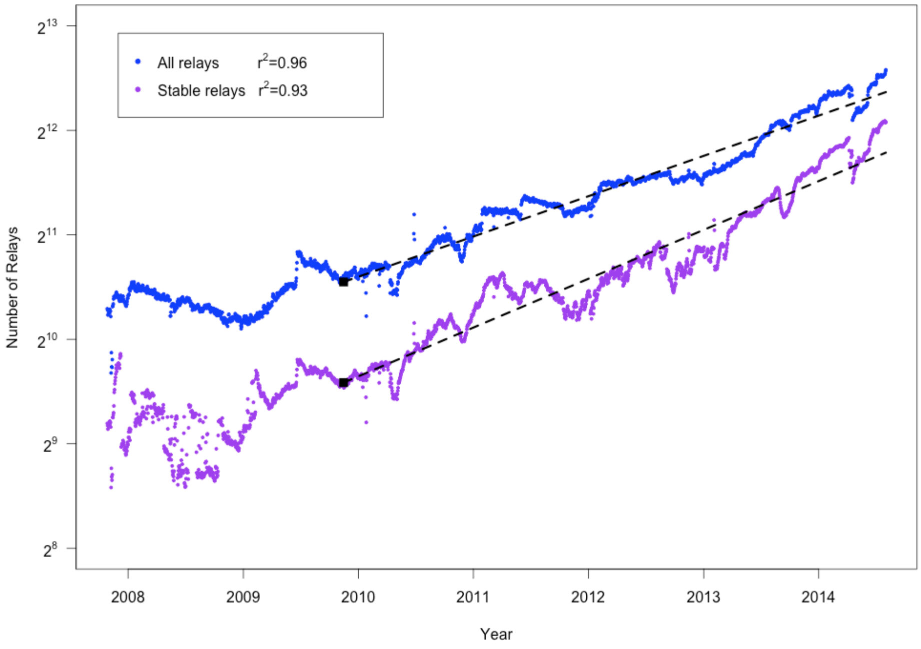


Figure 1: The number of Tor relays increases exponentially, doubling every 2 years (stable) to 2.5 years (all).

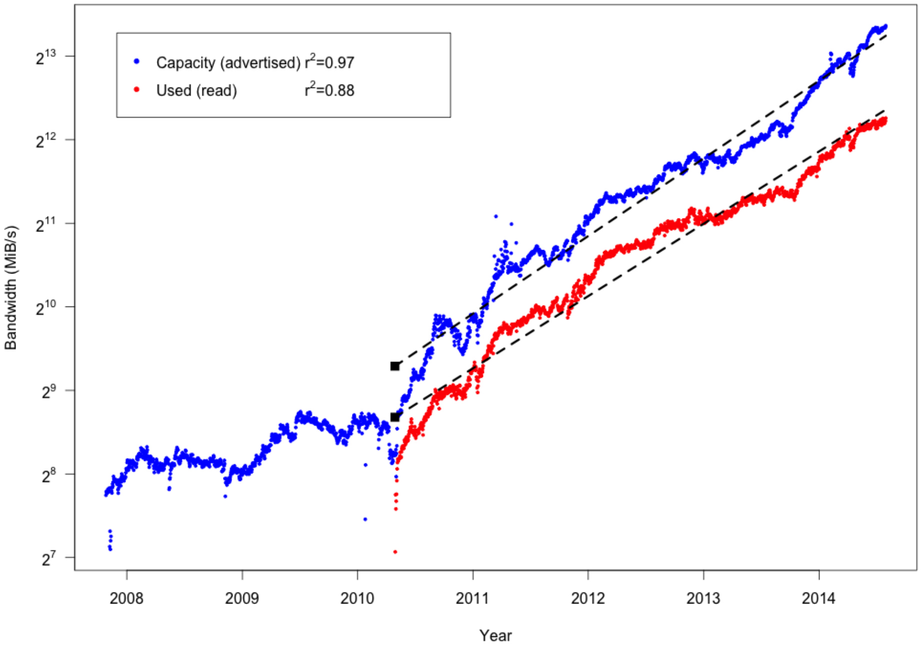


Figure 2: Total network bandwidth also increases exponentially.

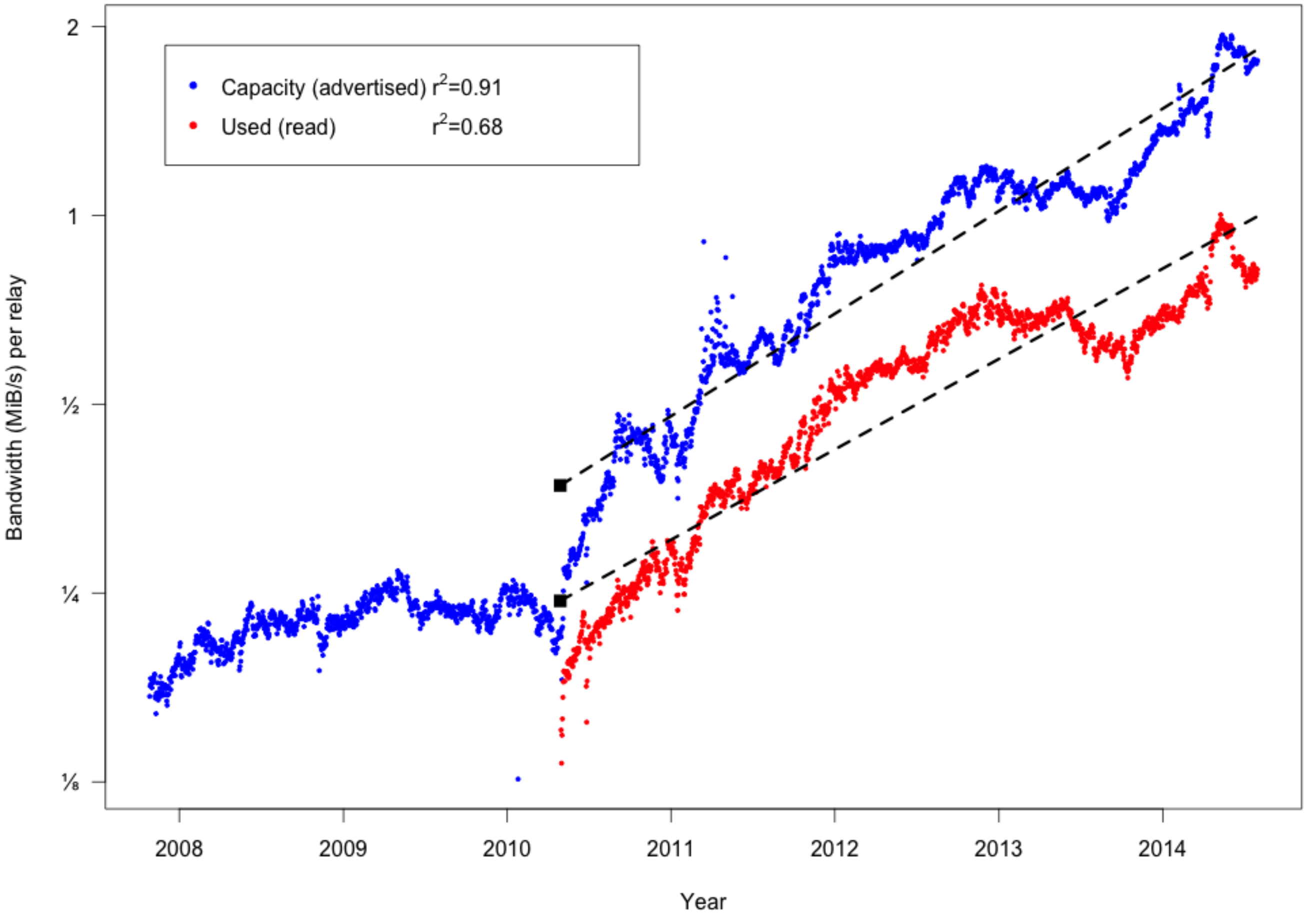


Figure 3: Mean relay bandwidth increases exponentially and doubles approximately every 24 months.

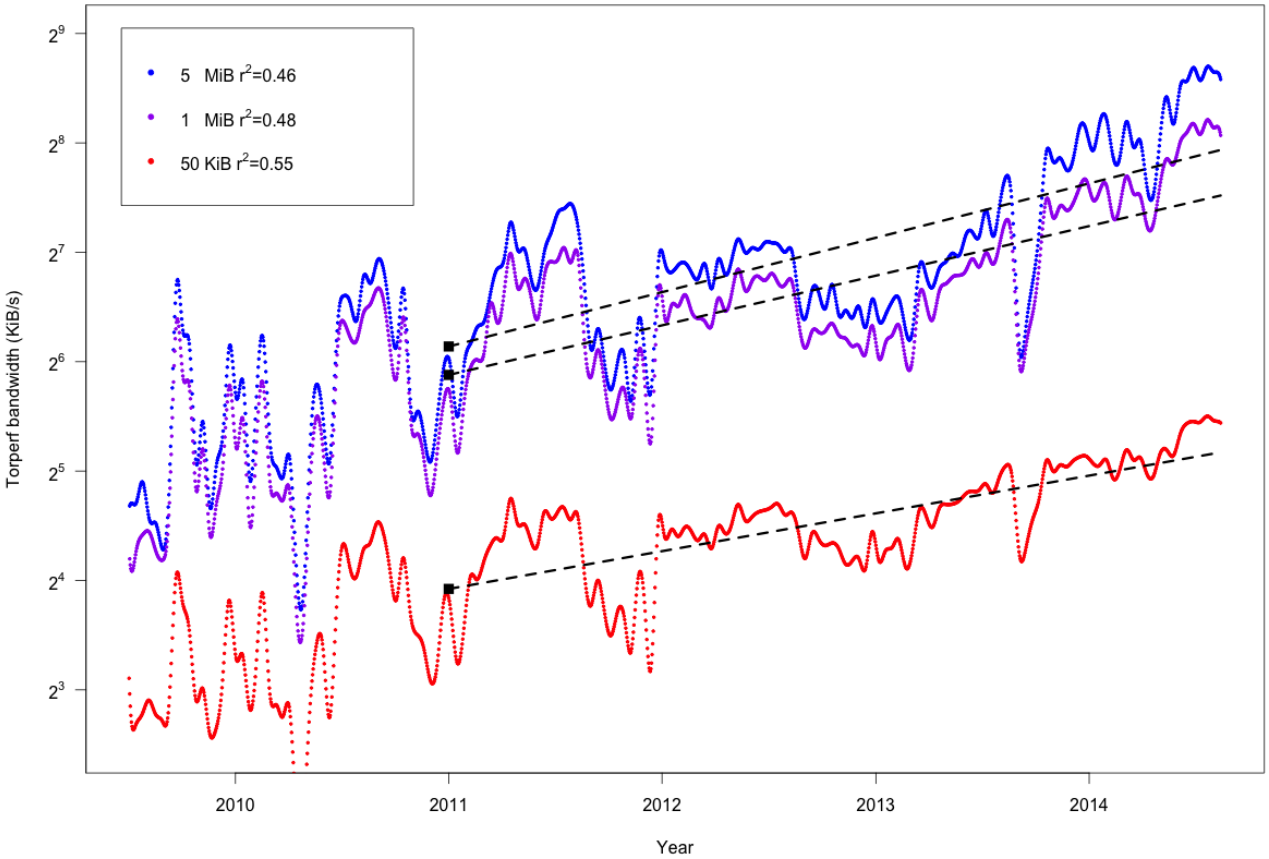


Figure 4: Absolute Torperf throughput increases exponentially, doubling every 25 months for 5 MiB files and every 35 months for 50 KiB files. Unfortunately, the throughput when downloading a 50 KiB file is ∼8x slower than downloading a 5 MiB file. These trends imply that these two rates will continue to diverge.

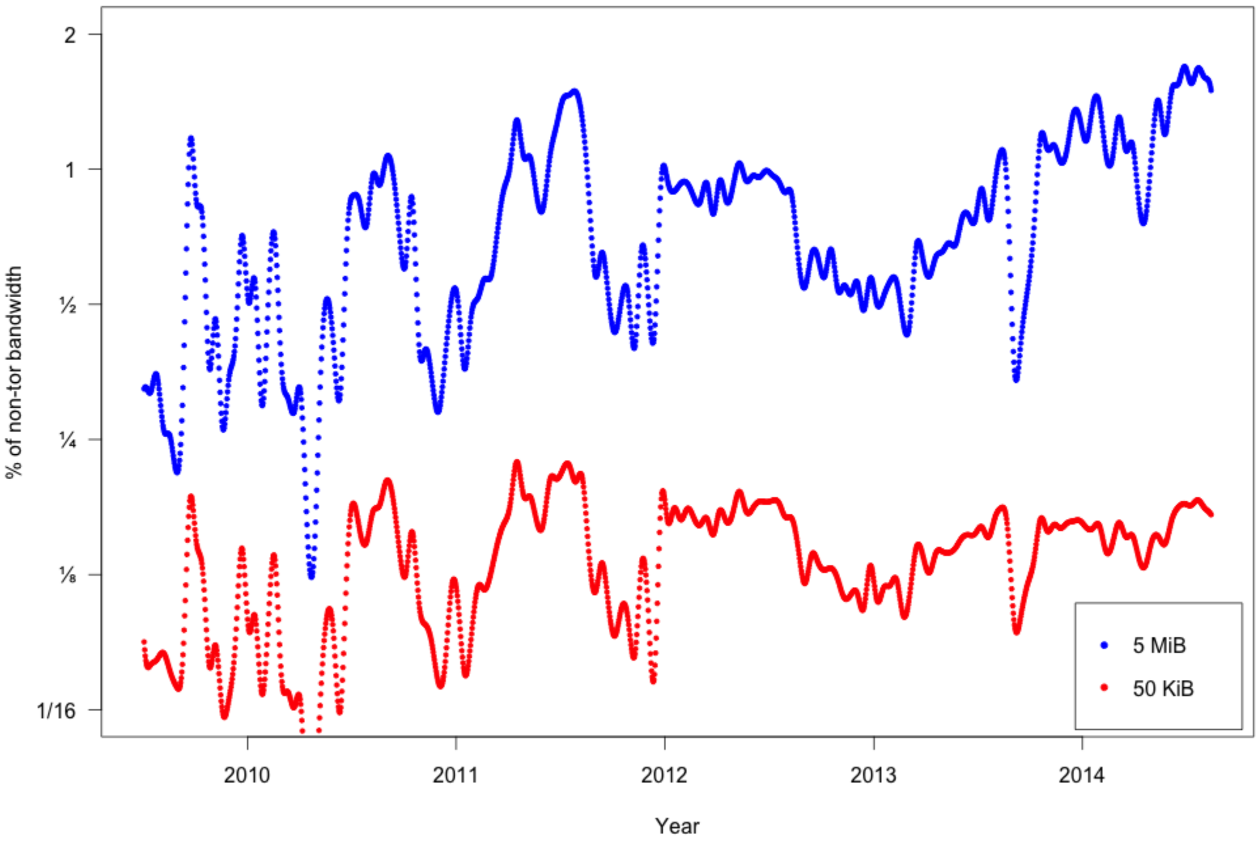
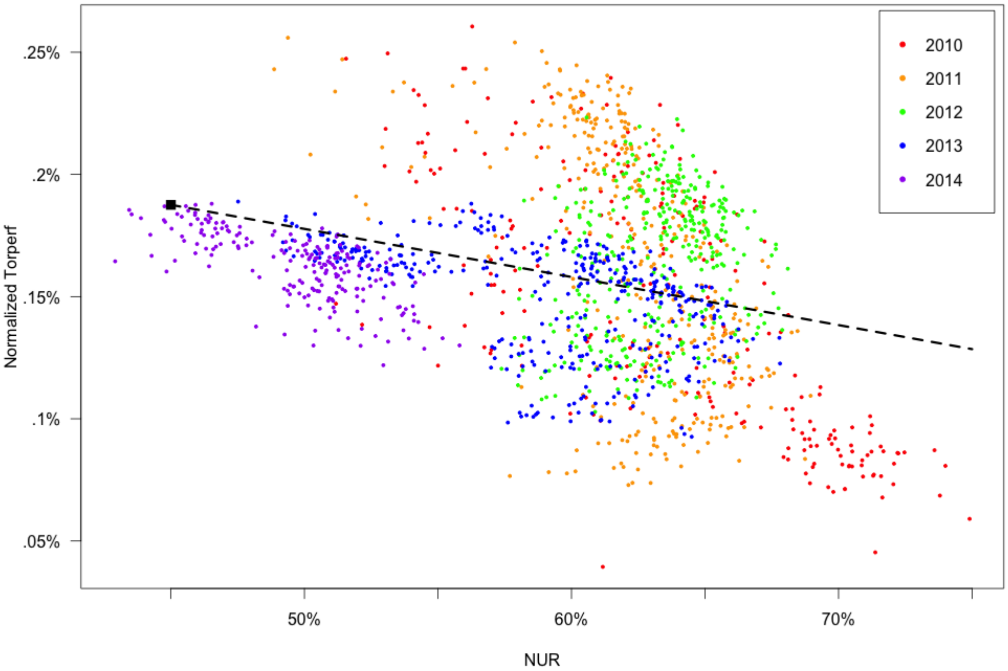
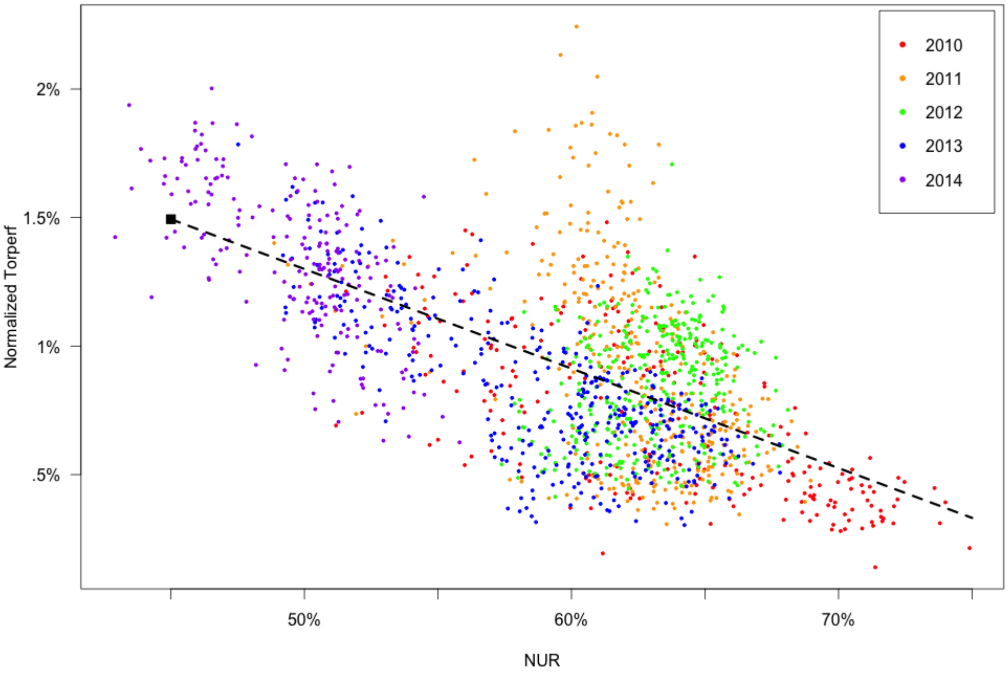


Figure 5: The normalized Torperf for 50 KiB and 5 MiB files.



(a) 50 KiB; $r^2 = 0.15$.



(b) 5 MiB; $r^2 = 0.44$.

Figure 6: Low NUR imples higher normalized Torperf—especially so for larger files.

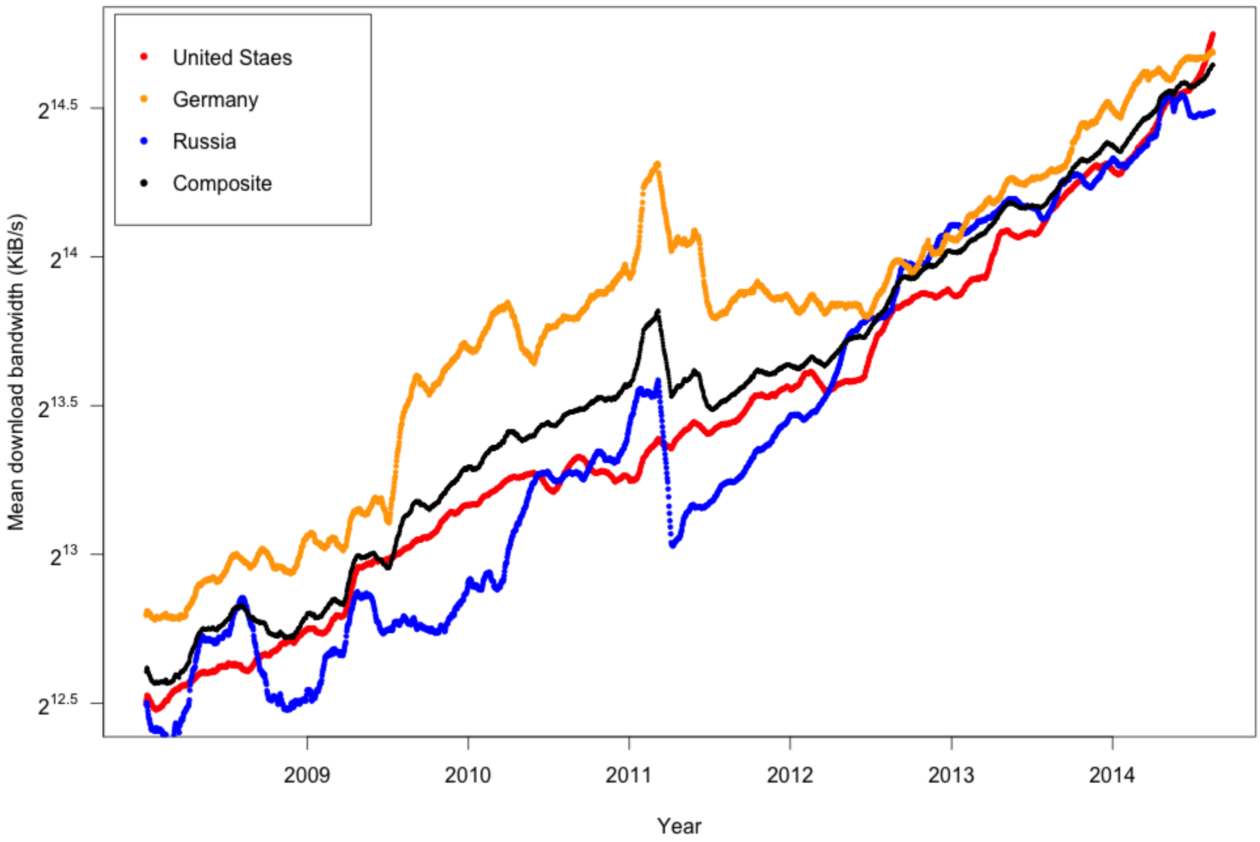


Figure 7: Mean download bandwidth for United States, Germany, and Russia according to netindex.com. Composite is the mean of all three.

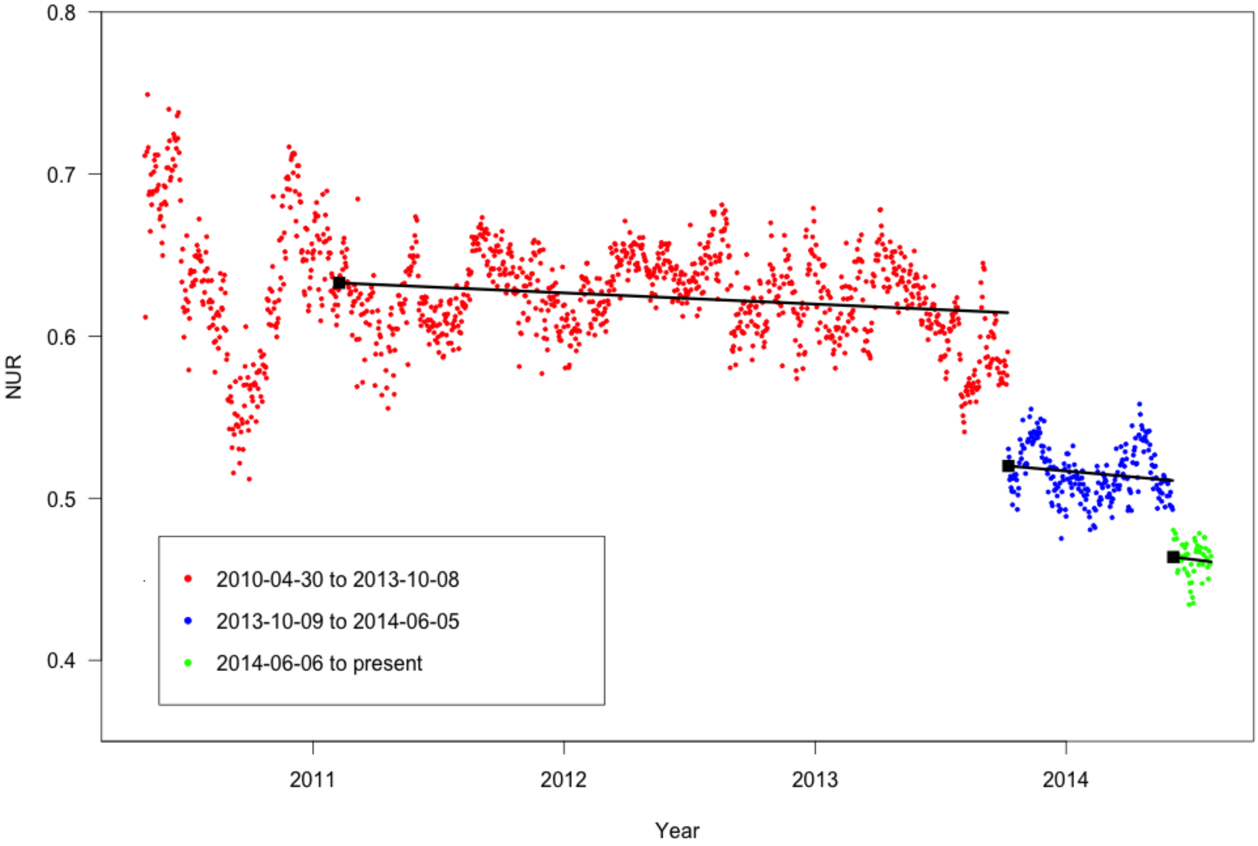


Figure 8: Network Utilization Ratio (NUR) falls into three distinct stages. Within each stage the fitted line is essentially flat. What happened on 2013-10-08 and 2014-06-06!? The only thing we see is that on 2014-06-05 (one day prior) the EFF began their Tor Challenge.