Improving Performance in the Tor Network

Laverne Schrock

Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Why Tor?

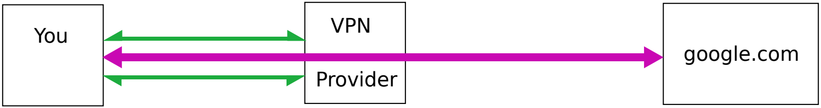
Tor was created to address two problems:

- Your Internet Service Provider can tell what sites you visit.
- Sites can tell from where you are connecting.

Outline

A Simple Solution: Use a VPN

A Virtual Private Network (VPN) routes all your traffic through another system.



VPN: Pros and Cons

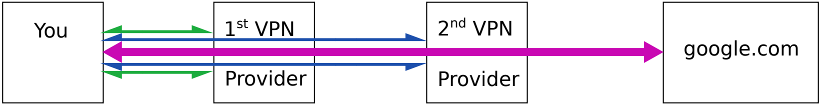
Pros:

- Metadata is now hidden from the ISP.
- Sites now perceive you as connecting from the VPN.

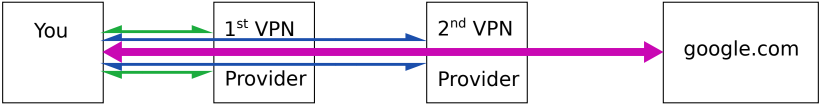
Cons:

- The VPN provider now knows who you are and what sites you visit.

An Improved Idea: Nesting VPNs



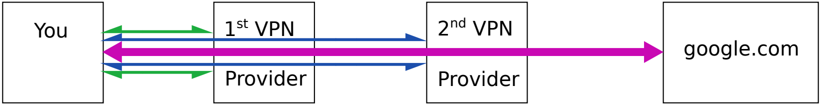
Nesting VPNs: Pros and Cons



Improvements:

- The 1st VPN provider can't tell that you're talking to Google.
- The 2nd VPN provider can't tell where you're connecting from.

Nesting VPNs: Pros and Cons



Problems with this idea:

- This is non-trivial to set up.
- You need an account with each VPN provider.
- You need to trust that the VPN providers are not compromised.

Tor's Solutions

Problem: This is non-trivial to set up.

Solution: Implement software specifically designed for this use case.

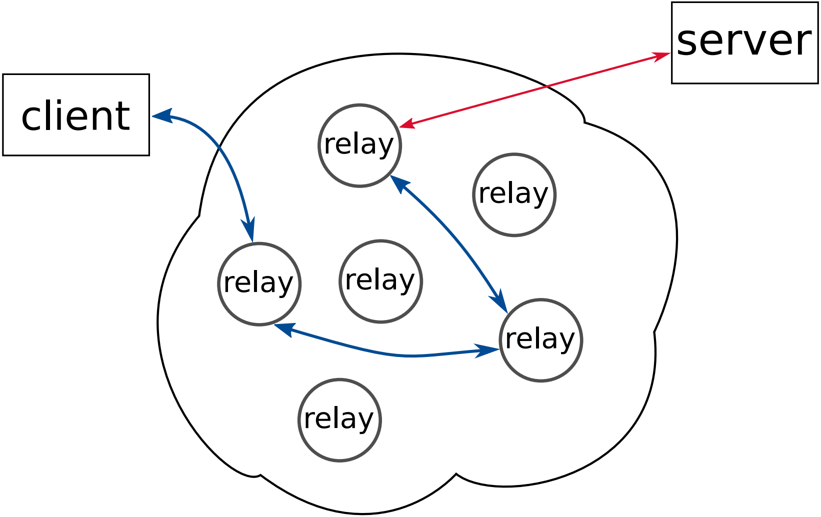
Problem: You need an account with each VPN provider.

Solution: Have volunteers run relays for free.

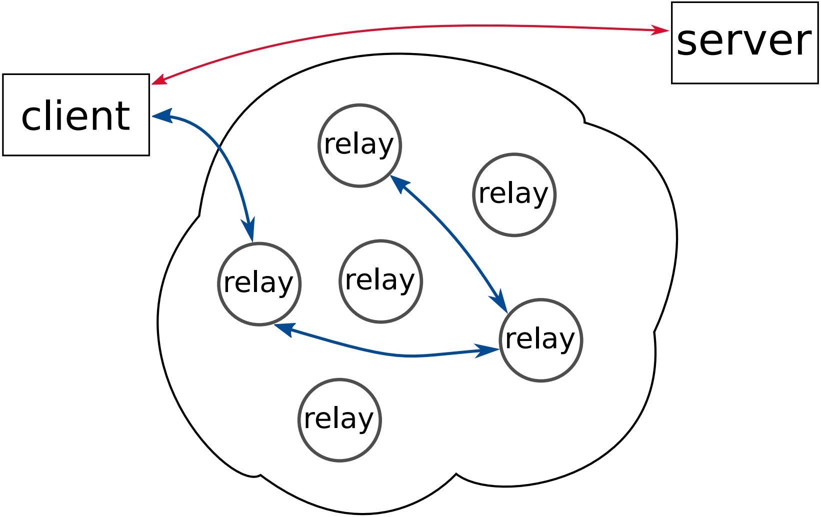
Problem: You need to trust that the VPN providers are not compromised.

Solution: Randomly choose different providers for each connection.

Tor: The Network



Tor: The Network



Lifecycle of a Tor circuit

1. Tor circuits take a while to build, so clients maintain a pool of circuits read to be used.
2. Relays are randomly chosen weighted by the maximum bandwidth of the relay.
3. A quality check is applied to the circuit before it is added to the pool.
4. When the user asks the client to load a web page, the client selects a circuit from the pool to service the request.
5. Circuits are cycled out of the pool and replaced with new ones.

Lifecycle of a Tor circuit

1. Tor circuits take a while to build, so clients maintain a pool of circuits read to be used.
2. Relays are randomly chosen weighted by the maximum bandwidth of the relay.
3. A quality check is applied to the circuit before it is added to the pool. (Circuit-RTT)
4. When the user asks the client to load a web page, the client selects a circuit from the pool to service the request.
5. Circuits are cycled out of the pool and replaced with new ones.

Lifecycle of a Tor circuit

1. Tor circuits take a while to build, so clients maintain a pool of circuits read to be used.
2. Relays are randomly chosen weighted by the maximum bandwidth of the relay.
3. A quality check is applied to the circuit before it is added to the pool. (Circuit-RTT)
4. When the user asks the client to load a web page, the client selects a circuit from the pool to service the request. (ABRA)
5. Circuits are cycled out of the pool and replaced with new ones.

Outline

Current Optimization: Circuit Build-Time

- Some circuits take longer to build, presumably because they are slower circuits.
- We have knowledge of what the distribution of build times looks like on the Tor network.
- Hence, clients only use circuits having a CBT in the fastest 80%.

New Optimization: Circuit RTT

Circuit-RTT (Circuit Round Trip Time) uses the exact same idea, but uses round trip time as the metric quality instead of build time.

- What is RTT?
- How is RTT measured in Tor?
- What distribution do round trip times follow?

Measuring RTT

- Relays can prohibit various types of connections.
- Some connection requests are hard-coded to be prohibited by all relays.
- To measure RTT, the client can send a prohibited request and measure the time until an error message is returned.

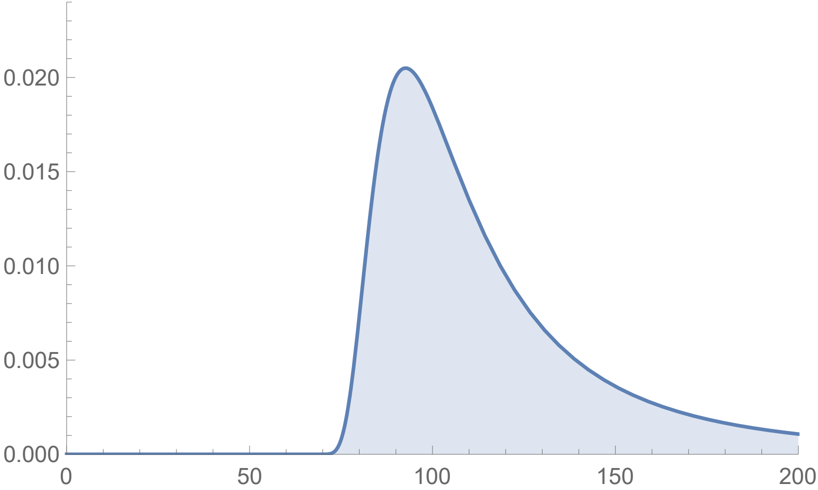
Distribution of RTTs

- Annessi and Schmiedecker used 19 computers scattered across Europe.
- Each computer built 1 million circuits and measured the RTT for each one.
- The measurements were found to follow a Generalized Extreme Value (GEV) distribution.

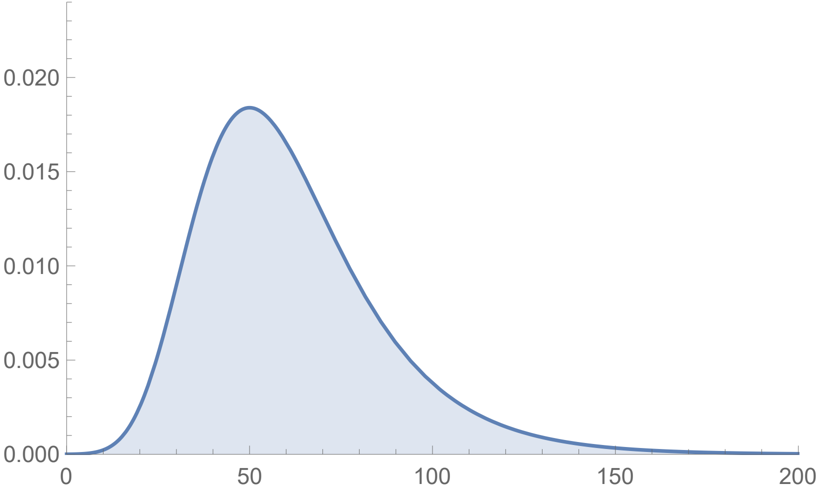
GEV distribution

- Allows one to make statements about how an individual number relates to the other numbers in the data set.
- In order to do so, we need to know the *parameters* of the distribution.

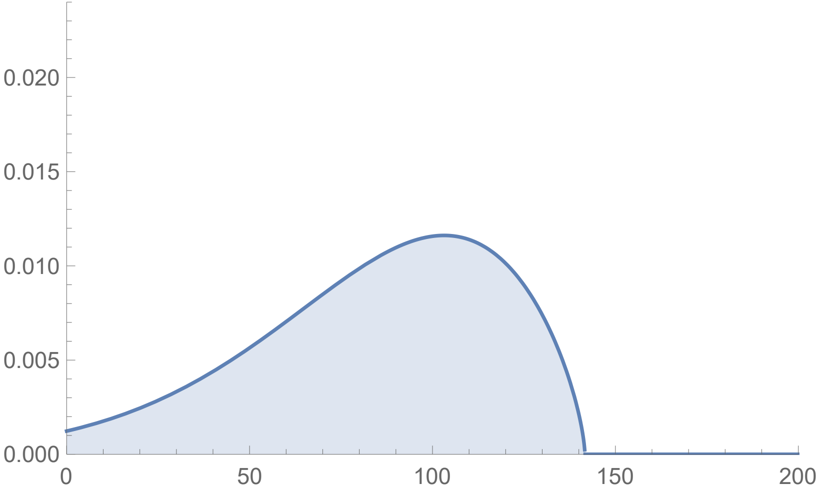
GEV distribution



GEV distribution



GEV distribution



GEV distribution

- Allows one to make statements about how an individual number relates to the other numbers in the data set.
- In order to do so, we need to know the *parameters* of the distribution.

Circuit-RTT Implementation

- Start building circuits and take an RTT measurement before adding to the pool
- Use the last 1000 RTT measurements to estimate the parameters of the distribution
- Only keep circuits having an RTT in the fastest 80% of circuits

Experiment Setup

1. Create a circuit
2. Measure Build Time
3. Measure Round Trip Time
4. Measure latency or bandwidth

Then, perform analysis on these results to see what choices CBT and CRTT would have made.

Results

Impact on latency

16 million measurements spread across 19 hosts

Results

Impact on latency

Table: TTFB when using the best 80% of circuits

	CBT (ms)	Circuit-RTT (ms)	Difference
Median	309	300	-2.9%
Worst*	541	499	-7.8%

*Ignoring the worst 10% to eliminate outliers.

Results

Impact on bandwidth

700 thousand measurements spread across 7 hosts

Results

Impact on bandwidth

Table: Bandwidth under top 80% CBT and Circuit-RTT

	CBT (Mb/s)	Circuit-RTT (Mb/s)	Difference
Median	3.24	3.30	+1.9%
Worst*	1.10	1.14	+3.6%

*Ignoring the worst 10% to eliminate outliers.

Outline

Relays: Allocating Bandwidth

- If too many circuits try to use a relay at the same time, the relay won't be able to provide all the bandwidth they desire.
- A relay will only increase the bandwidth of a circuit by allocating free bandwidth or bandwidth from higher-usage circuits.
- When two circuits are using different amounts of bandwidth at a relay, one of them is not being restricted at this point.
- Thus, at a relay we expect to see a cluster of high-usage circuits and a spread of lower-usage circuits.

Relays: Computing the metric

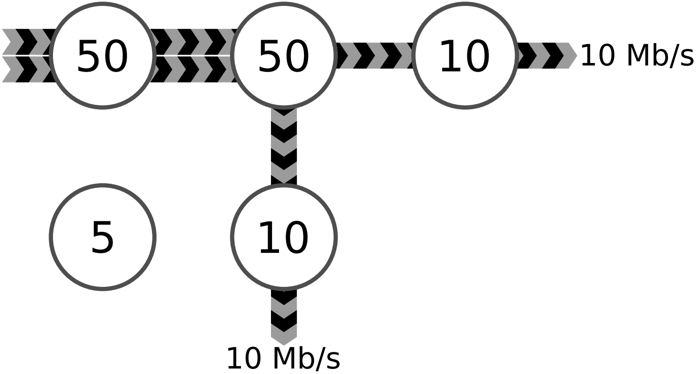
- Use a statistical grouping method to select bandwidths of the limited circuits, *bandwidths*.
- Let the *weight* of the relay be the sum of the inverses of these bandwidths,

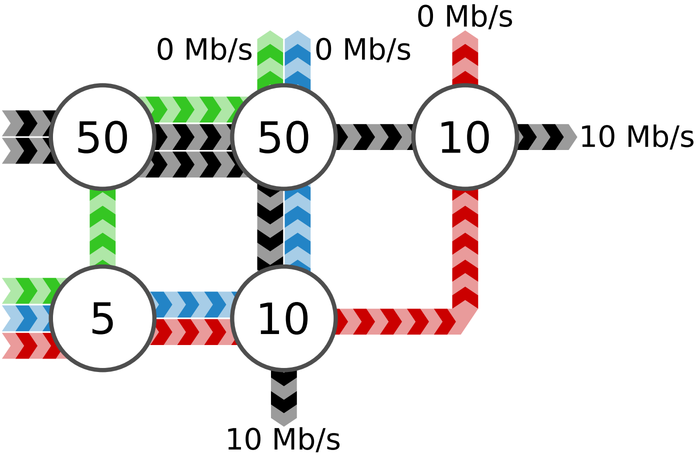
$$weight = \sum_{b \in bandwidths} \frac{1}{b}.$$

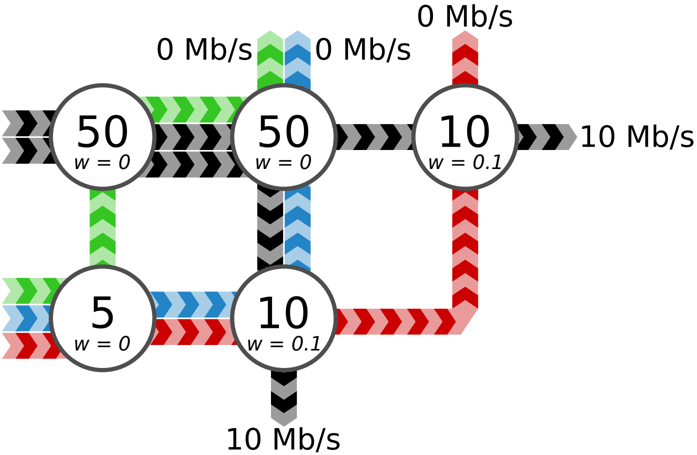
- If the relay is not using all its bandwidth, let $weight = 0$.
- Sign a message declaring the weight and share it with all the clients.

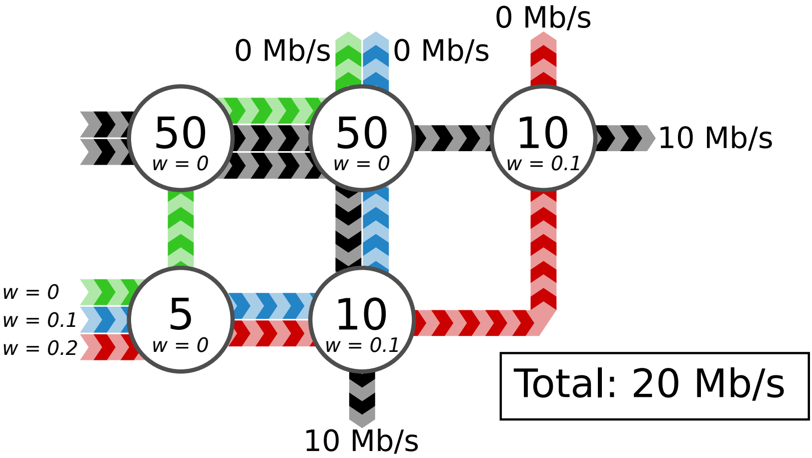
Clients: Using the metric

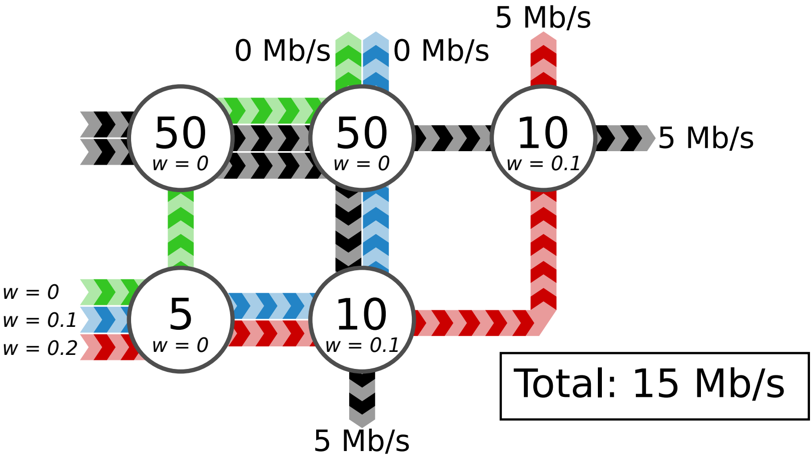
- Sum the weights of relays to get a weight per circuit.
- Choose randomly from all the circuits with a weight of 0.
- If none available, choose randomly based on the inverse weight, so circuits with a lower weight are more likely to be chosen.

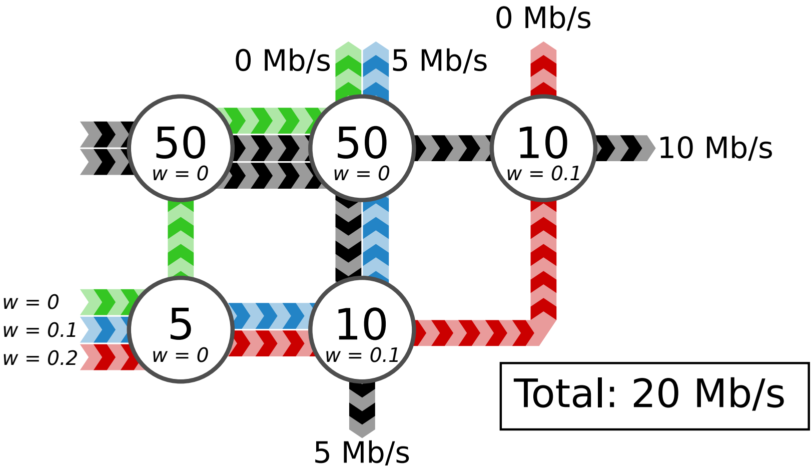

50
50
10
10 Mb/s
5
10
10 Mb/s

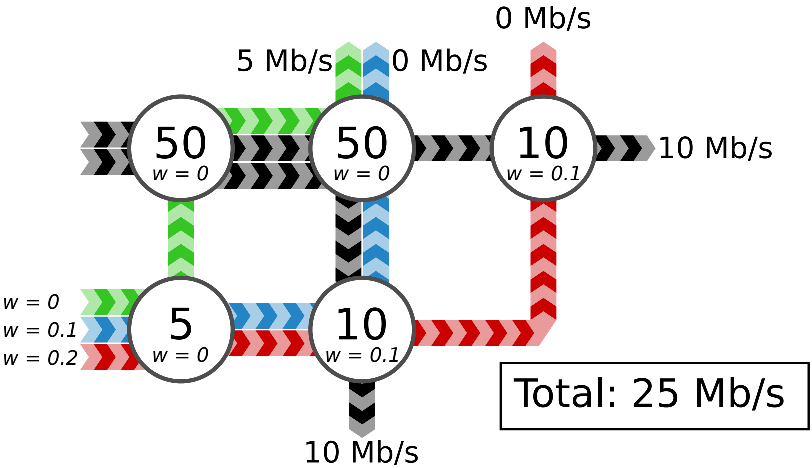

0 Mb/s
0 Mb/s
0 Mb/s
50
50
10
10 Mb/s
5
10
10 Mb/s


0 Mb/s
0 Mb/s
0 Mb/s
50
w = 0
50
w = 0
10
w = 0.1
10 Mb/s
5
w = 0
10
w = 0.1
10 Mb/s


0 Mb/s
0 Mb/s
0 Mb/s
50
w = 0
50
w = 0
10
w = 0.1
10 Mb/s
w = 0
w = 0.1
w = 0.2
5
w = 0
10
w = 0.1
Total: 20 Mb/s
10 Mb/s


5 Mb/s
0 Mb/s
0 Mb/s
50
w = 0
50
w = 0
10
w = 0.1
5 Mb/s
w = 0
w = 0.1
w = 0.2
5
w = 0
10
w = 0.1
Total: 15 Mb/s
5 Mb/s


0 Mb/s
0 Mb/s
5 Mb/s
50
w = 0
50
w = 0
10
w = 0.1
10 Mb/s
w = 0
w = 0.1
w = 0.2
5
w = 0
10
w = 0.1
Total: 20 Mb/s
5 Mb/s


0 Mb/s
5 Mb/s
0 Mb/s
50
w = 0
50
w = 0
10
w = 0.1
10 Mb/s
w = 0
w = 0.1
w = 0.2
5
w = 0
10
w = 0.1
10 Mb/s
Total: 25 Mb/s

Simulation Setup

Because ABRA makes changes to both the server and the client pieces of Tor, Geddes et al. tested the changes in a simulation.

- 500 web servers
- 1300 web clients (320 KB downloads)
- 150 bulk clients (1 MB downloads)
- 500 relays

Experimental Results

- Total network bandwidth was 14% higher.
- Both ABRA and standard Tor had 80% of their TTFB measurements under 1 second, but above that, ABRA's measurements covered a broader range.
- Web download times improved by 5%-10%.

Outline

Conclusion

What we covered:

- What Tor is, problems it addresses, and how improving the performance of Tor helps it achieve its goals.
- Two new strategies for avoiding use of poor circuits:
  - Circuit-RTT
  - ABRA

Acknowledgments

Thanks to Kevin Arhelger and KK Lamberty for their input on this project.

Questions?

References I

R. Annessi and M. Schmiedecker.
Navigator: Finding faster paths to anonymity.
In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 214–226, March 2016.

R. Dingledine, N. Mathewson, and P. Syverson.
Challenges in deploying low-latency anonymity (draft).
https://svn.torproject.org/svn/projects/design-paper/challenges.pdf, 2005.

References II

B. Fung.
Trump has signed repeal of the FCC privacy rules. Here's what happens next.
https://www.washingtonpost.com/news/the-switch/wp/2017/04/04/trump-has-signed-repeal-of-the-fcc-privacy-rules
April 2017.

J. Geddes, M. Schliep, and N. Hopper.
ABRA CADABRA: Magically Increasing Network Utilization in Tor by Avoiding Bottlenecks.
In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, WPES '16, pages 165–176, New York, NY, USA, 2016. ACM.

References III

Tor metrics.
https://metrics.torproject.org/.