Throttling Tor Bandwidth Parasites

Rob Jansen Paul Syverson Nicholas Hopper
U.S. Naval Research Laboratory University of Minnesota
{rob.g.jansen, paul.syverson}@nrl.navy.mil hopper@cs.umn.edu

Tor's network congestion and performance problems stem from a small percentage of users that consume a large fraction of available network capacity. These users continuously drain relays of excess bandwidth, creating new network bottlenecks and exacerbating the effects of existing ones. This may currently be a performance issue due to unfair resource consumption, but it also shows that the network is vulnerable to malicious service degradation by a relatively low-resource adversary using similar techniques.

There are three general approaches to alleviate Tor's performance problems: optimizing scheduling strategies, increasing network capacity, and reducing network load. Improved path selection and circuit scheduling may shift network load to better utilize the available bandwidth, but do not increase the capacity of the network or provide any defense against DoS. Organizations such as The Tor Project may increase network capacity by joining new relays to the network, however this approach is a short-term solution that does not scale: the bulk users attracted to the faster network will continue to leech the additional bandwidth.

In this work, we present the design of three new algorithms that throttle clients to reduce network congestion and increase web client performance. Unlike existing techniques, our new adaptive throttling algorithms use information local to a relay to dynamically select which connections get throttled and to adjust the rate at which those connections are throttled. Adaptively tuned throttling mechanisms are paramount to our algorithm designs in order to avoid the need to re-evaluate parameter choices as network capacity and load changes. Our *bit-splitting* algorithm adaptively throttles each connection at its fair share of bandwidth, our *flagging* algorithm adaptively throttles only connections whose throughput exceeds the statistically fair throughput, and our *threshold* algorithm adaptively throttles connections above a throughput quantile at a rate represented by that quantile.

We implement our algorithms in Tor and compare significant client performance benefits using network-wide deployments of our algorithms under a range of light to heavy network loads. We test various configurations of our algorithms and compare our results to static throttling under a varied range of network loads. We find that the effectiveness of the existing static throttling approach is highly dependent on network load and configuration whereas our adaptive algorithms work well under various loads with no configuration changes or parameter maintenence: web client performance was improved for every parameter setting we tested. Figure 1 shows that each of our new algorithms provides a significant improvement in web client performance over vanilla Tor.

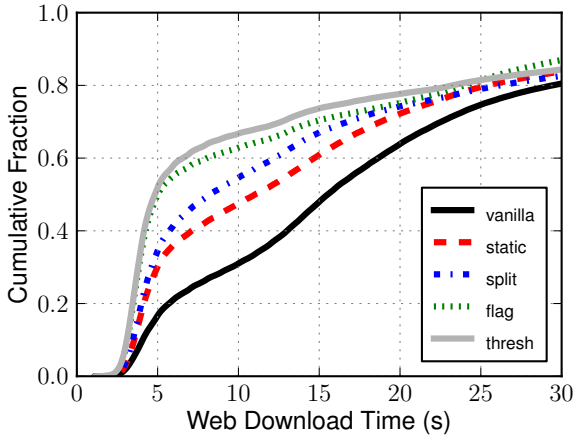

Figure 1: Throttling improves web client performance for each throttling algorithm over vanilla Tor, using 50 relays and a load of 950 web clients and 50 bulk clients.

We also analyze the security of our algorithms under adversarial attack, discussing several realistic attacks on anonymity while comparing the information leaked by each algorithm relative to unthrottled Tor. We find that throttling clients reduces information leakage and improves network anonymity against realistic adversaries, and is an effective defense against practical bulk traffic DoS attacks while minimizing the false positive impact on honest users.

For a more detailed discussion of this work, please see our technical report [1].

References

[1] Rob Jansen, Paul Syverson, and Nicholas Hopper, *Throttling Tor Bandwidth Parasites*. University of Minnesota TR 11-019, 2011.