Christoph Döpmann, Florian Tschorsch

CircuitStart: A Slow Start For Multi-Hop Anonymity Systems

CircuitStart: A Slow Start For Multi-Hop Anonymity Systems

Christoph Döpmann
Technische Universität Berlin, Germany
christoph.doepmann@campus.tu-berlin.de

Florian Tschorsch
Technische Universität Berlin, Germany
florian.tschorsch@tu-berlin.de

ABSTRACT

In order to improve the performance of anonymity networks like Tor, custom transport protocols have been proposed to efficiently deal with the multi-hop nature of such overlay networks. In this work, we tackle the issue of quickly, but safely, ramping up the congestion window during the initial phase of a circuit's lifetime. We propose a tailored startup mechanism called CircuitStart that transfers the idea of a traditional slow start to the multi-hop scenario by effectively compensating potential overshooting, improving performance compared to existing approaches.

CCS CONCEPTS

• **Networks** → **Transport protocols**; *Network privacy and anonymity*; • **General and reference** → **Performance**; • **Security and privacy** → *Pseudonymity, anonymity and untraceability*;

KEYWORDS

Tor, multi-hop communication, slow start algorithm

ACM Reference Format:
Christoph Döpmann and Florian Tschorsch. 2018. CircuitStart: A Slow Start For Multi-Hop Anonymity Systems. In *SIGCOMM Posters and Demos '18: ACM SIGCOMM 2018 Conference Posters and Demos, August 20–25, 2018, Budapest, Hungary*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3234200.3234211

1 INTRODUCTION

Anonymity networks like Tor [3] provide anonymous communication by relaying data, packaged into fixed-size cells, over a virtual *circuit*, consisting of typically three nodes. The so-called *onion routing* protocol introduces a fundamental challenge regarding performance. The *multi-hop* nature, i.e., using a sequence of TCP connections, has to be taken into account when carrying out congestion control. While approaches like *Split TCP* [2] deliberately chain multiple low-latency TCP connections to improve performance, Tor generally exhibits high RTTs that negate any potential benefit from such constructions. In general, previous research has shown that several issues arising from Tor's design can be tackled with tailored transport protocols [1]. Most tailored approaches, however, neglect the protocol dynamics, particularly the question of how to ramp-up the congestion window during the initial phase of a circuit. As Tor is designed for interactive use, this is of special importance. Approaches like JumpStart [4], which follow a strategy similar to our approach, are not suitable for multi-hop scenarios. In addition, it is desired that Tor traffic behave much like background traffic, i.e., avoiding aggressive traffic patterns. We tackle these challenges by introducing *CircuitStart* and present initial evidence drawn from simulations that shows its capability to improve Tor circuits' ramp-up behavior.

SIGCOMM Posters and Demos '18, August 20–25, 2018, Budapest, Hungary

ACM ISBN 978-1-4503-5915-3/18/08…$15.00
https://doi.org/10.1145/3234200.3234211

2 CIRCUITSTART

While applicable more broadly, we focus specifically on Tor, the most prominent anonymity network today. We assume a custom, window-based transport protocol that allows low-latency communication between neighboring relays. Making such an assumption is valid, because research has shown that this could be an approach to tackle many of Tor's performance deficiencies [1]. Our main goal is to provide a start-up scheme for such a transport protocol.

Algorithm Description. The main challenge arising from the multi-hop scenario consists in the source's need to react to network conditions (bandwidth and delay) it cannot observe directly. A potential bottleneck may be located some hops away in the circuit. We leverage a combination of delay-based congestion control and special feedback messages to cope with this issue.

When forwarding a cell to its successor, each relay issues a feedback message to its predecessor, signaling cells are "moving". This way, the existence of network congestion *and* overloaded relays is propagated back to the sender.

We exploit this feedback to realize our CircuitStart algorithm. Each relay starts with an initial congestion window (cwnd) of two cells. Upon reception of the respective feedback from the successor, it doubles the cwnd and sends an according number of cells. Note that this behavior differs from a traditional slow start. Firstly, an increase of the cwnd is not triggered by the reception of an ACK, but by feedback messages indicating that the cell has been *forwarded* by the successor relay. Therefore, cwnd growth also captures the state of the successor relay, not only the network in between. Secondly, the window growth does not happen continuously, but in discrete rounds, carried out once per RTT after having received an appropriate number of feedback messages. The resulting *packet trains* allow for a more elaborate analysis of the timing information gathered.

To this end, relays also compute the RTTs between transmission of a cell and reception of the respective feedback message. These values are used to estimate the successor's queue length, along the lines of TCP Vegas. We calculate the difference between the expected and actual window size as

$$diff = cwnd \cdot \frac{currentRtt}{baseRtt} - cwnd,$$

where *baseRtt* denotes the overall minimum RTT and *currentRtt* corresponds to the latest round. Similarly to TCP Vegas, we define a threshold γ, currently set to 4. If $diff > \gamma$, this hints at a growing queue at the successor relay. Thus, we exit the ramp-up and perform congestion avoidance. However, the cwnd can still massively "overshoot", especially if the bottleneck is distant from the source.

Therefore, CircuitStart carries out what we call *overshooting compensation*. When leaving the slow start phase, traditional start-up schemes would halve the cwnd before entering congestion avoidance. In contrast, CircuitStart carries out a more sophisticated calculation: the cwnd is set to *the amount of data acknowledged within the current round so far*. This window re-calculation is motivated by the observation that the length of the packet train that could be forwarded by the successor without additional delay, is a good estimation for the optimal window, because it is the minimal value that suffices to fully utilize the network.

Backpropagation. CircuitStart implicitly propagates the minimum cwnd seen in the circuit back to the source. Thus, it roughly estimates the optimal cwnd at the source. Intuitively, this is achieved because when a bottleneck relay reduces its cwnd, its predecessor will observe that at most this amount of data can be forwarded at once, setting its cwnd to the same value. This continues until the source is reached.

Note that, if network delay differs significantly between relays, the optimal window may be underestimated. However, this is in line with our goal of being safe to avoid aggressive traffic.

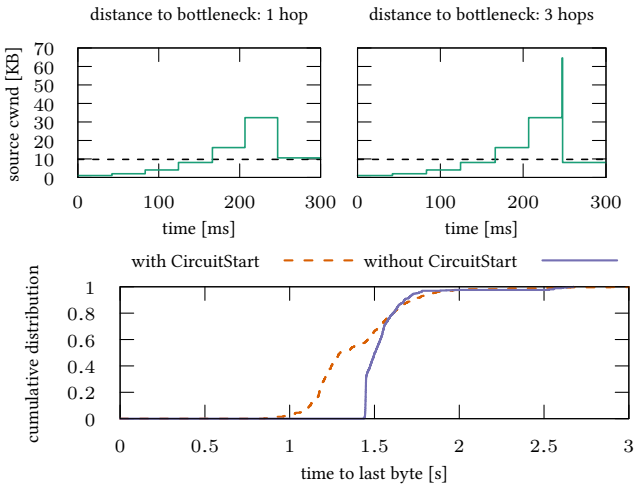

Figure 1: Evaluation (single traces and aggregated performance metrics).

Implementation & Evaluation. We implemented CircuitStart in nstor [5], a simulation framework for the network-level behavior of Tor, based on ns-3. While CircuitStart is not tied to a specific transport protocol as long as the aforementioned assumptions are met, for evaluation purposes, we chose to implement it for the BackTap protocol [5]. As a baseline, we developed a model to calculate the source's *optimal congestion window* in a multi-hop scenario.

Figure 1 (upper plots) shows the cwnd development of a source, where the bottleneck is located at different positions in the circuit. These single runs are representative for CircuitStart's behavior: CircuitStart manages to accurately estimate the optimal cwnd (dashed line), completely compensating the temporary overshooting of the cwnd. At the same time, our approach is able to quickly adjust the cwnd independently of the bottleneck's location.

Moreover, we measured the overall download times when transferring a fixed amount of data over a randomly generated network of Tor relays, connected in a star topology. We simulated 50 concurrent circuits. The results, as depicted in Figure 1, show that CircuitStart is able to improve the achieved performance when compared to BackTap, by up to 0.5 seconds.

3 CONCLUSION & FUTURE WORK

With CircuitStart, we presented a novel startup scheme that is tailored to application in a multi-hop anonymity overlay network, such as Tor. As such, it allows for rapid adaption of the congestion window to non-local network bottlenecks. Our future work will include expanding the scope of the algorithm to not only the initial phase of a circuit, but to enable it to quickly respond to changing network conditions during the congestion avoidance phase.

REFERENCES

[1] Mashael AlSabah and Ian Goldberg. 2016. Performance and Security Improvements for Tor: A Survey. *ACM Compututing Surveys* 49, 2 (2016).

[2] John Border, Markku Kojo, Jim Griner, Gabriel Montenegro, and Zach Shelby. 2001. RFC 3135: Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. http://www.ietf.org/rfc/rfc3135.txt

[3] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *USENIX Security '04: Proceedings of the 13th USENIX Security Symposium*.

[4] Dan Liu, Mark Allman, Shudong Jin, and Limin Wang. 2007. Congestion control without a startup phase. In *PFLDnet '07: Proceedings of the 2007 International Workshop on Protocols for Fast Long-Distance Networks*.

[5] Florian Tschorsch and Björn Scheuermann. 2016. Mind the Gap: Towards a Backpressure-Based Transport Protocol for the Tor Network. In *NSDI '16: Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation*.