DeTor: Provably Avoiding Geographic Regions in Tor

Zhihao Li, Stephen Herwig, and Dave Levin, ***University of Maryland***

https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/li

This paper is included in the Proceedings of the 26th USENIX Security Symposium

August 16–18, 2017 • Vancouver, BC, Canada

ISBN 978-1-931971-40-9

Open access to the Proceedings of the 26th USENIX Security Symposium is sponsored by USENIX

DeTor: Provably Avoiding Geographic Regions in Tor

Zhihao Li, Stephen Herwig, and Dave Levin
University of Maryland

Abstract

Large, routing-capable adversaries such as nation-states have the ability to censor and launch powerful deanonymization attacks against Tor circuits that traverse their borders. Tor allows users to specify a set of countries to exclude from circuit selection, but this provides merely the illusion of control, as it does not preclude those countries from being on the path *between* nodes in a circuit. For instance, we find that circuits excluding US Tor nodes definitively avoid the US 12% of the time.

This paper presents DeTor, a set of techniques for proving when a Tor circuit has avoided user-specified geographic regions. DeTor extends recent work on using speed-of-light constraints to prove that a round-trip of communication physically could not have traversed certain geographic regions. As such, DeTor does not require modifications to the Tor protocol, nor does it require a map of the Internet's topology. We show how DeTor can be used to avoid censors (by never transiting the censor once) and to avoid timing-based deanonymization attacks (by never transiting a geographic region twice). We analyze DeTor's success at finding avoidance circuits through simulation using real latencies from Tor.

1 Introduction

Tor [8] has proven to be an effective tool at providing anonymous communication and combating online censorship. Over time, Tor's threat model has had to adapt to account for powerful nation-states who are capable of influencing routes into and out of their borders—so-called *routing-capable* adversaries [34].

We consider two key threats that the presence of routing-capable adversaries now makes a practical reality. First, routing-capable adversaries can (and regularly do) censor Tor traffic. While it is well-known that some countries block Tor traffic beginning or ending within borders, recent studies have shown that some also block any Tor traffic that happens to transit *through* their borders [4]. Second, routing-capable adversaries can launch deanonymization attacks against Tor. If an adversarial network is on the path of the circuit between source and entry, and between exit and destination, then it can introduce small, detectable jitter between packets to correlate the two connections and therefore uncover the source and destination [19].

In light of increasingly powerful attacks like these, Tor has added the ability for users to specify a set of countries to exclude when selecting circuits. However, as we will demonstrate, this offers users only the illusion of control over where their traffic does not go. Among the circuits that Tor uses to ostensibly ignore the US, we could identify only 12% of them as definitively avoiding the US. Alternative schemes have been proposed that involve using `traceroute` to construct a map of the Internet's topology. However, routing-capable adversaries can easily (and regularly do [35]) provide incomplete responses to `traceroute`, precluding provable security from mapping-based approaches.

In this paper, we present a set of techniques that can *prove* that a circuit has avoided a geographic region. One of the most powerful features of these techniques is how little they require compared to many prior approaches: they do not require modifying the hardware [3] or routing protocols [30] of the Internet, nor do they require a map of the Internet's routing topology [12]. Instead, our work extends recent work on "provable avoidance routing" [24] that uses geographic distances and speed-of-light constraints to prove where packets physically could not have traversed. Users can specify arbitrary geographic regions (our techniques do not rely on any notion of network topology or ownership), and we return per-packet proofs of avoidance, when available.

We construct avoidance in Tor in two applications:

Never-once proves that packets forwarded along a circuit never traversed a given geographic region, even once. With this, users can avoid website fingerprinting attacks [23] and censoring regimes [4].

Never-twice proves that packets forwarded along a circuit do not reveal more information to a geographically constrained adversary than is strictly necessary by ensuring that they do not appear on two non-contiguous legs of the Tor circuit. With this, users can prevent certain deanonymization attacks [2, 17, 29, 10, 15].

In sum, this paper makes the following contributions:

- We introduce the notion of Tor circuits that *provably* avoid regions of the world. Unlike prior approaches, our proofs do not depend on any model of network or

AS-level topologies, and are instead based on round-trip time measurements. Therefore, they are easy to collect, do not require modifications to Tor, and do not depend on Internet measurements that are manipulable by a powerful adversary.

- We present the design, analysis, and evaluation of two novel forms of avoidance: never-once to avoid censors and website fingerprinting attacks, and never-twice to avoid various traffic deanonymization attacks.
- We build these techniques in a system we call *DeTor*, and evaluate it using real Tor latencies collected by the Ting measurement tool [6]. We show that provable, never-once avoidance is possible even when avoiding routing-central countries like the US, and that provable never-twice avoidance works for *98.6%* of source-destination pairs not in the same country.

Collectively, our results show that, with client-side techniques alone, it is possible to achieve greater control over where Tor data does *not* go. We believe this to be a powerful building block in future defenses.

2 Background and Related Work

In this section, we describe some of the attacks that are possible against Tor from a powerful routing-capable adversary. We also discuss prior work that has sought to mitigate these attacks. First, we begin by reviewing the relevant details of the Tor protocol.

2.1 A Brief Review of Tor

Tor [8] is a peer-to-peer overlay routing system that achieves a particular type of anonymity known as unlinkable communication. A source-destination pair is *unlinkable* if no one other than the two endpoints can identify *both* the source and destination. That is, an observer may know the source (or destination) is communicating with *someone*, but cannot identify with whom.

Tor achieves unlinkable communication by routing traffic through a *circuit*: a sequence of overlay hosts. There are typically three hosts in a circuit: an entry node[1] (who communicates with the source), a middle node, and an exit node (who communicates with the destination). The source node is responsible for choosing which Tor routers to include in a circuit, and for constructing the circuit. Tor's default circuit selection algorithm chooses nodes almost uniformly at random to be in a circuit, with three notable exceptions: (1) nodes with greater bandwidth are chosen more frequently, (2) no two nodes from the same subnet are chosen to be in the same circuit, and (3) no nodes are chosen from a user-specified list of countries to ignore.

Circuit construction is done in such a way that the only host who knows all hops on the circuit is the source: each other host knows only the hop immediately preceding and succeeding it. By the end of the circuit construction protocol, the source has established a pairwise secret (symmetric) key with each hop on the circuit.

The salient feature of Tor is the manner in which it performs "onion routing." When sending a packet p to the destination, the source encrypts p with the symmetric key it shares with the exit node; it then encrypts this ciphertext with the key shared with the middle node; and in turn encrypts this doubly-encrypted ciphertext with the key shared with the entry node. Each hop on the circuit "peels off" its layer of encryption, thereby ensuring that anyone overhearing communication between any two consecutive Tor routers learns nothing about the other Tor routers on the circuit.

2.2 Threat Model

We assume a powerful routing-capable adversary [34], e.g., a nation-state. Such an attacker has the ability to make (potentially false) routing advertisements and can therefore attract routes to its administrative domain. Thus, routing-capable adversaries are able to insert themselves onto the path between two communicating endpoints. Once on the path, they can launch various man-in-the-middle attacks by injecting, dropping, delaying, or reordering packets.

Routing-capable adversaries can also mislead or obfuscate attempts to map their networks. For example, one common approach for mapping a network is to use `traceroute`, but even benign networks sometimes refuse to respond to ICMP packets, tunnel their packets through their internal network, or simply do not decrement TTLs. These efforts effectively hide routers from a `traceroute` measurement, and could allow a nation-state adversary to hide its presence on a path. It is because of these kinds of attacks that we choose not to employ `traceroute`-based measurements in our system.

Because we are mainly focused on nation-state adversaries, we assume that the attacker can be geographically localized. For example, to avoid the United States, we assume that a user can download the geographic information (GPS coordinates) that succinctly describe where the US is (including its territories, such as Guam) and that these constitute all of the locations from which the country could launch attacks. This was the same assumption made by Levin et al. [24]. In practice, it may be possible that an adversary could infiltrate other countries' networks, but there are many instances where a nation-

[1] Alternatively, clients can make use of so-called bridge nodes, which are in essence non-public entry nodes. Because they serve the same purpose as traditional entry nodes, they pose no difference in DeTor, and so we refer to them collectively as "entry nodes."

state deploys its censorship mechanisms within its borders [7, 13].

This attack model extends naturally to colluding countries, such as the Five Eyes: one can simply consider them as one large "nation-state" that is constrained to its (potentially noncontiguous) borders. As we will demonstrate, because our techniques apply to noncontiguous geographic regions, they are not restricted to single nation-states, and can be applied to arbitrary sets of countries.

The attacker can also run its own Tor routers or collude with some Tor routers, but, as per the previous assumption, only within its own (or its fellow colluders') borders.

Finally, we make several assumptions about what an attacker *cannot* do. We assume the attacker cannot violate standard cryptographic assumptions, particularly that it cannot invert cryptographically secure hash functions, infer others' private keys, or forge digital signatures or MACs. Also, we note that, while an attacker can lie about having *larger* latencies (by delaying its own packets), it is unable to lie about having *lower* latencies than its links actually permit.

2.3 Attacks

This paper considers three very powerful attacks that are at the disposal of a routing-capable adversary. We review the attacks here, and then describe how prior work has sought to mitigate them.

Censorship A routing-capable adversary can censor traffic that enters its borders. Commonly, with Tor, this involves identifying the set of active Tor routers and simply dropping traffic to or from these hosts. The Tor Metrics project monitors several countries who appear to perform this kind of censorship [37].

Traffic deanonymization Consider an attacker who is able to observe the traffic on a circuit between the source and the entry node and between the exit node and the destination. The attacker can correlate these two seemingly independent flows of traffic in a handful of ways. For instance, a routing-capable adversary operating a router on the path between source and entry could introduce jitter between packets that it could detect in the packets between exit and destination. This works because Tor routers do not delay or shuffle their packets, but rather send them immediately in order to provide low latencies [6].

Website fingerprinting Even an attacker limited to observing only the traffic between the source and entry node can be capable of deanonymizing traffic. In particular, if the destination's traffic patterns (e.g., the number of bytes transferred in response to apparent requests) are well-known and unique, then an attacker may be able to infer the destination by observing the traffic on any leg of the circuit [23]. Such attacks run into challenges when there is sufficient cover traffic, but unfortunately Tor users have little control over how much cover traffic there is.

2.4 Related Work

Sneaking through censors The traditional way of mitigating censoring nation-states is to sneak through them by making would-be-censored traffic look benign. For example, decoy routing [21, 41] uses participating routers that are outside the censoring regime but on a benign path to effectively hijack traffic and redirect it to a destination that would be censored. To the censoring regime, the traffic appears to be going to a destination it permits.

Other approaches employ protocol obfuscation techniques to make one protocol look like another. A slew of systems [26, 40, 27, 38, 39, 18] has explored making Tor traffic appear to be other, innocuous traffic, notably Skype (many censors permit video chat applications, so as to allow their citizens to keep in touch with friends and family abroad). We seek an altogether different approach of *avoiding these nefarious regions altogether*, rather than trying to sneak through them. However, these are somewhat orthogonal approaches, and may be complementary in practice.

AS-aware Tor variants The work perhaps closest to ours in terms of overall goals is a series of systems that try to avoid traversing particular networks once (or twice). To the best of our knowledge, these have focused almost exclusively on using autonomous system (AS)-level maps of the Internet [2, 19, 10]. Like DeTor, the idea is that if we can reason about and enforce where our packets may (or may not) go *between* hops in the circuit, we can address attacks such as censorship and certain forms of traffic deanonymization.

As described in §2.2, however, we assume in this paper an adversary who has the ability to manipulate an observer's map of the Internet, for instance by withholding some routing advertisements, withholding `traceroute` responses, and so on. Instead of relying on these manipulable data sources, we base our proofs on *physical, natural* limitations: the fact that information cannot travel faster than the speed of light. As an additional departure from this line of work, we focus predominately on nation-state adversaries, which are easier to locate and geographically reason about than networks (which may have points of presence throughout the world).

DeTor's proofs of avoidance come at a cost that systems that do not offer provable security do not have to

pay. DeTor discards any circuit for which it cannot obtain proof of avoidance, but it is possible that there are circuits that achieve avoidance that do not meet the requirements of the proof. As a result, DeTor clients have fewer circuits to choose from than more permissive systems that rely only on AS maps, potentially leading to greater load imbalance in DeTor. We believe this to be a fundamental cost of security with provable properties; however, we also believe that future work can reduce DeTor's "false negatives."

Avoidance routing Recent work has proposed not to sneak through attackers' networks, but to avoid them altogether. Edmundson et al. [11] propose to use maps of the Internet's routing topology to infer through which countries packets traverse, and to proxy traffic through those who appear to avoid potential attackers. However, as with AS-aware Tor variants, this approach relies on data that can be significantly manipulated by the kind of powerful routing-capable adversaries that we consider. In this paper, we seek techniques that yield provable security, even in the presence of such adversaries.

Alibi Routing [24] uses round-trip time measurements and speed of light constraints to provably avoid user-specified, "forbidden" geographic regions. The Alibi Routing protocol only uses a single relay; we generalize this approach to apply to Tor's three-hop circuits. Moreover, our never-twice application avoids doubly-traversing any region of the world, and does not require an *a priori* definition of a forbidden region. We review Alibi Routing next.

3 Background: Alibi Routing

We build upon techniques introduced in Alibi Routing [24] to achieve provable avoidance in Tor. In this section, we briefly review how Alibi Routing achieves its proofs of avoidance, and we outline the challenges we address in translating it to Tor.

3.1 Proofs of Avoidance

Alibi Routing [24] is a system that provides proof that packets have avoided a user-specified geographic region. Specifically, a source node s specifies both a destination t and a *forbidden region* F. Node s trusts all nodes that are provably outside F (we return to this point at the end of this subsection). Alibi Routing then seeks to identify a relay a that is not in F and that satisfies the following property. Let $R(x,y)$ denote the round-trip time (RTT) between hosts x and y, and let R_{e2e} denote the end-to-end RTT that s observes; then for a user-configurable $\delta \geq 0$:

$$(1+\delta)\cdot R_{\mathsf{e2e}} \;<\; \min \begin{cases} \min_{f\in F}\left[R(s,f)+R(f,a)\right]+R(a,t) \\ R(s,a)+\min_{f\in F}\left[R(a,f)+R(f,t)\right] \end{cases} \quad (1)$$

When this inequality holds, it means that the RTT for s forwarding packets through a to t is significantly less than the smallest round-trip time that would *also* include a host in the forbidden region. In other words, if s can verify that its traffic is going through a and t, then the traffic could not also go through F without inducing a noticeable increase in end-to-end round-trip time. With such a relay, s can prove that his packets avoided F with two pieces of evidence:

1. A MAC (or digital signature) from a attesting to the fact that it did indeed forward the packet, and
2. A measured end-to-end round-trip time that satisfies Eq. (1).

These two pieces of evidence form an "alibi": the packets went through a and could not also have gone through F, therefore it avoided F. As a result, Levin et al. [24] refer to a relay a who provides such a proof an *alibi peer*.

These proofs of avoidance must be obtained for *each* round-trip of communication. The factor of δ acts as an additional buffer against variable latencies. The larger δ is, the fewer potential alibis there will be, but they will be able to provide proofs of avoidance even when packets suffer an uncharacteristically high delay, for instance due to congestion. DeTor makes use of δ in the same manner.

One technical detail in Alibi Routing's proof that we will make use of is the process of computing $\min_{f\in F}[R(s,f)+R(f,a)]$ and $\min_{f\in F}[R(a,f)+R(f,t)]$. After all, how can one compute the shortest RTT through a host in an untrusted region of the world? The insight is that, if we know the geographic locations of the hosts in question, then we can obtain a lower bound on the round-trip time between them. In particular, if $D(x,y)$ denotes the great-circle distance between hosts at locations x and y, then we have the following bound:

$$\min_{f\in F}\left[R(s,f)+R(f,a)\right]+R(a,t) \;\geq\; \frac{3}{c}\cdot\left(\min_{f\in F}\left[D(s,f)+D(f,a)\right]+D(a,t)\right) \quad (2)$$

where c denotes the speed of light. In general, information cannot travel faster than the speed of light; in practice, information tends to travel no faster than two-thirds the speed of light. Coupled with a $2\times$ factor to capture the RTT, this gives us the $\frac{3}{c}$ value in Eq. (2) as a way to convert the great-circle distance between two hosts to a

minimum RTT on the Internet. Provided with the set of geographic coordinates defining the border of F, one can compute the geographic coordinate $f \in F$ that provides this minimum distance. Critically, computing this does not require any participation from F (e.g., sending responses to pings)—it only depends on knowing the geographic coordinates of those trusted to forward the packets: s, t, and a.

As mentioned above, Alibi Routing assumes that node s trusts all nodes that are provably outside of its specified forbidden region F. To determine if a node n is definitively outside of F, s directly measures the RTT to n by asking it to echo a random nonce. Recall from §2 that attackers cannot lie about having lower latencies; thus, if this measured RTT is smaller than the theoretical minimum RTT between s and F, then n cannot be in F. Alibi Routing applies these trust inferences transitively. We adopt this assumption in our DeTor, as well.

3.2 Remaining Challenges

Applying Alibi Routing to Tor is not immediately straightforward. First, Alibi Routing is defined only with respect to a single alibi relay, whereas Tor circuits consist of three or more relays. Even if we were to extend the proofs from Eqs. (1) and (2), it is not obvious how well this would work in practice. As we will demonstrate, we are able to extend Alibi Routing's approach to Tor, and that it is surprisingly effective at finding "alibi circuits."

Second, the notion of a fixed forbidden region does not directly apply to the problem of deanonymization attacks like those described in §2.3. Recall that these attacks arise when an adversary is on both (a) the path between source and entry node and (b) the path between exit node and destination. Avoiding a region F altogether (as with Alibi Routing) ensures that F could not have launched such an attack, but it is overly restrictive to do so. Note that it is not necessary to avoid a given region altogether—it suffices to ensure that the region is not on the path *twice*, at both the entry and exit legs of the circuit. This relaxation allows users to protect themselves against deanonymization attacks launched by their home countries, whereas it would be impossible to avoid one's own country altogether.

Moreover, using a static forbidden region would require users to anticipate all of those who could have launched an attack. Ideally, a solution would be more adaptive, by permitting avoidance of the form "wherever packets might have gone between source and entry, avoid those places between exit and destination."

We demonstrate an adaptive "never-twice" technique that provably avoids regions that could launch deanonymization attacks, and we demonstrate that it is highly successful on the Tor network.

4 Provable Avoidance in Tor

In this section, we introduce how to construct proofs that a round-trip of communication (a packet and its response) over a Tor circuit has avoided geographic regions of the world. These proofs have the benefit of being easy to obtain (they largely consist of taking end-to-end round-trip time measurements), easy to deploy (they do not require modifications to Internet routing or buy-in from ISPs), and resilient to manipulation.

4.1 Never-Once Avoidance

The goal of never-once avoidance is to obtain proof that at no point during a round-trip of communication could a packet or its response have traversed a user-specified forbidden region F. Like with Alibi Routing, our proof consists of two parts:

First, we obtain proof that the packets *did* go through selected Tor routers. Whereas Alibi Routing traverses only a single relay, we traverse a circuit of at least three hops. Fortunately, Tor already includes end-to-end integrity checks in all of its relay cells [8], which successfully validate so long as the packets followed the circuit and were unaltered by those outside or inside the circuit. This serves as proof that the packets visited each hop, and, thanks to onion routing, that they visited each hop in order.

Second, we obtain proof that it could not *also* have gone through the forbidden region. To this end, we measure the end-to-end round-trip time R_{e2e} through the entire circuit, and we compute the shortest possible time necessary to go through each circuit *and* the forbidden region:

$$R_{\mathsf{min}} \;=\; \frac{3}{c} \cdot \min \begin{cases} D_{\mathsf{min}}(s,F,e,m,x,t) \\ D_{\mathsf{min}}(s,e,F,m,x,t) \\ D_{\mathsf{min}}(s,e,m,F,x,t) \\ D_{\mathsf{min}}(s,e,m,x,F,t) \end{cases} \tag{3}$$

Here, $D_{\mathsf{min}}(x_1,\ldots,x_n)$ denotes the shortest possible great-circle distance to traverse nodes $x_1 \to \cdots \to x_n$ in order. We abuse notation to also account for regions—for example, $D_{\mathsf{min}}(s,F,e) = \min_{f\in F}[D(s,f)+D(f,e)]$. Note that Eq. (3) is in essence a generalization of Alibi Routing's single-relay proof (Eq. (2)), and can be easily extended to support longer circuits.

Equation (3) captures the shortest possible distance to go through each hop in the circuit (in order) as well as through F. It also applies the observation that information tends to travel no faster than two-thirds the speed of light on the Internet. For example, in Figure 1, the top circuit has its shortest detour through F between the middle and exit nodes; the bottom circuit's shortest trajectory

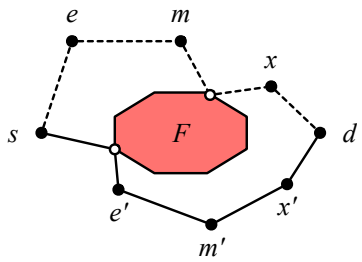


Figure 1: **Never-once:** To prove that a Tor circuit has avoided region F, we compute the shortest possible distance to traverse the circuit as well as a point in F. This figure has two example circuits, showing that the shortest distance might traverse F at different legs of the circuit.

that includes F does so between source and entry.

Last, we compare this theoretical minimum RTT including F (R_{min}) with the end-to-end measured RTT (R_{e2e}), and ensure that

$$(1+\delta)\cdot R_{\mathsf{e2e}} < R_{\mathsf{min}}. \tag{4}$$

For round-trip communications that pass Tor's integrity checks and satisfy Eq. (4), we obtain our proof that the packets could not have possibly traversed the forbidden region. For those that do *not* satisfy the equation, much like Alibi Routing, we are unable to distinguish whether the packets traversed the forbidden region or, e.g., were simply delayed on a congested link. We discuss such application-level considerations in §5.3.

4.2 Never-Twice Avoidance

The goal of never-twice is to ensure that a potential adversary is not able to see and manipulate *both* the traffic between source and entry node *and* the traffic between exit node and destination. Adversaries who are on both the entry and exit legs of a circuit are able to launch deanonymization attacks [28, 14, 17, 15]. However, an adversary on no more than one of those legs cannot.

As with never-once, Tor's onion routing ensures that the packet and its response indeed traveled through the Tor circuit in order, and we measure the end-to-end round-trip time R_{e2e}. However, as described in §3.2, our step for establishing mutual exclusion requires a significantly different approach.

The attacker seeks to be on the path both between s and e (the entry leg) and between x and t (the exit leg). All other parts of the circuit (entry to middle and middle to exit) do not help the adversary in this particular attack.

Never-twice avoidance of a single host We begin by constructing a proof that a circuit could not have traversed any single host on both the entry and exit legs. Ultimately, we seek to show that there is no point p for which $\frac{3}{c}\cdot D_{\mathsf{min}}(s,p,e,m,x,p,t) \le R_{\mathsf{e2e}}$.

Iterating through all points on the Earth would be computationally infeasible; although we do not have a closed-form solution, we present an efficient empirical check.

Note that the best-case scenario for the attacker is that all traffic on the (e,m) and (m,x) legs of the circuit take the least amount of time possible: a total of $R_m = \frac{3}{c}\cdot D(e,m,x)$. This leaves a total remaining end-to-end latency of $R_{\mathsf{e2e}} - R_m$. This is the total time the packets have to traverse (s,e) and (x,t); the larger this value, the greater the chance an attacker can be on the path of both of these legs (in the extreme, were this difference on the order of seconds, there would be enough time to theoretically traverse any point on the planet on both legs).

A useful way to visualize this problem is as two ellipses. Recall that an ellipse with focal points a and b and radius r represents all points p such that $d(a,p) + d(p,b) \le r$. Larger values of r result in ellipses with greater area, while larger values of $d(a,b)/r$ result in more elongated ellipses (in the extreme, an ellipse with $d(a,b) = 0$ is a circle).

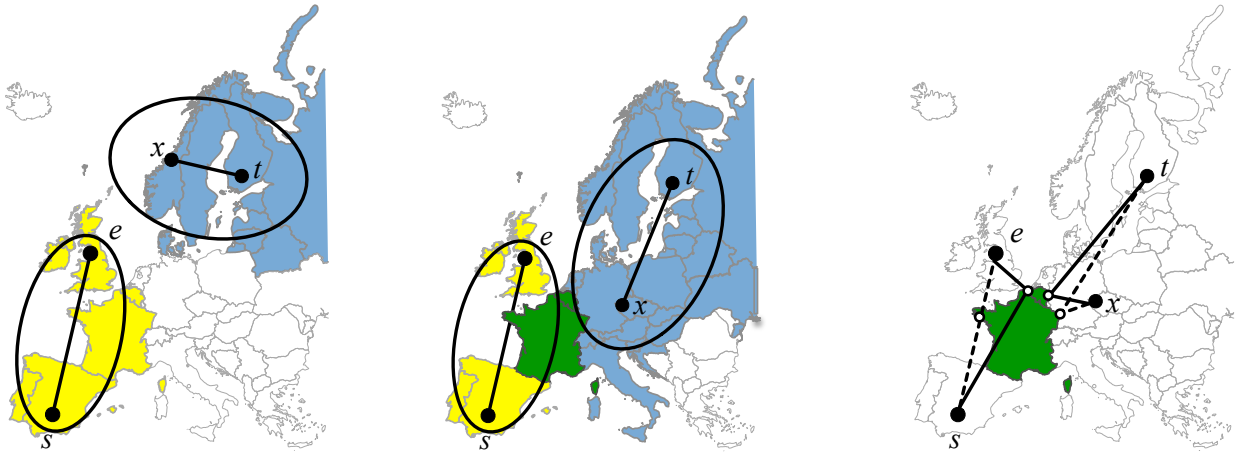
Thus, we can view this problem as two ellipses—one with focal points s and e and radius r_e and the other with focal points x and t and radius r_x, such that $r_e + r_x = \frac{c}{3}\cdot(R_{\mathsf{e2e}} - R_m)$.

If these two ellipses intersect, then there could exist a host through which the traffic on both the entry and exit leg could have traversed.

Never-twice avoidance of a country The above technique for avoiding double traversal of a single host does not preclude a powerful attacker such as a nation-state from deploying multiple vantage points within their borders. For example, as in Figure 2b, consider an ellipse around the entry leg that traverses southwest Europe, and an exit leg that traverses central and eastern Europe—even though the two ellipses never intersect one another, they share two common nation states: France and Belgium. We next explore how to avoid double traversal of countries.

This process begins by identifying the set of countries that either leg could go through were all of the extra latency spent on either leg individually. This corresponds to two ellipses: one with focal points s and e and radius $\frac{c}{3}\cdot(R_{\mathsf{e2e}} - R_m) - D(x,t)$, and the other with focal points x and t and radius $\frac{c}{3}\cdot(R_{\mathsf{e2e}} - R_m) - D(s,e)$. We intersect these ellipses with a database of countries' borders to obtain the sets of countries that could have been traversed on the entry leg (C_e) and on the exit leg (C_x).

If $C_e \cap C_x = \emptyset$, as in Figure 2a, then it is not possible for the same country to have been traversed twice, and



(a) When C_e (yellow) and C_x (darker blue) do not intersect, double-traversal of any country is impossible.

(b) When C_e and C_x *do* intersect (dark green), we must compute the shortest distances for both legs to go through each country (right). The dashed lines show the shortest distances through France, and the solid lines through Belgium.

Figure 2: **Never-twice**: To prove that a Tor circuit did not traverse any given country at the beginning and end of the circuit, we compute the set of countries C_e that could have been on the path of the entry leg and the countries C_x that could have been on the exit leg. This figure shows two example circuits with different exit nodes.

we have our proof of never-twice avoidance.

However, if the intersection is non-empty, as in Figure 2b, then we need to perform additional checks. For each country $F \in C_e \cap C_x$, we ensure that the minimum RTT to go through the entry leg and F plus the minimum RTT to go through the exit leg and F is larger than the end-to-end RTT would allow:

$$\forall F \in C_e \cap C_x : \quad (1+\delta) \cdot R_{\text{e2e}} \quad < \\ \frac{3}{c} \cdot (D_{\min}(s,F,e) + D(e,m,x) + D_{\min}(x,F,t)) \tag{5}$$

The subtle yet important difference between Eq. (5) and the previous equations is that the right hand side need not minimize distance with respect to a single $f \in F$. Rather, there could be two distinct points in F: one on the entry leg's path and another on the exit leg's. This puts the attacker at a greater advantage; consider the above example wherein the entry leg was geographically isolated to western France and the exit leg was isolated to eastern France. When a single $f \in F$ required to be present on both legs, the packets would be required to traverse an extra distance of roughly twice the width of France. But with separate points in F, it could impose arbitrarily low additional delay.

What Eq. (5) does share in common with the other equations is that it can be computed purely locally, using only the knowledge of the circuit relays' locations and a database of countries' borders, which are readily available [16].

Never-twice avoidance of colluding countries Finally, we consider how to avoid deanonymization attacks from a group of countries who might coordinate their efforts. For example, the Five Eyes is an alliance of five countries (Australia, Canada, New Zealand, the United Kingdom, and the United States) who have agreed to share intelligence. Were such a group of countries to collude, then traversing one of them on the entry leg and another on the exit leg could result in a successful deanonymization attack.

Our above method for avoiding double-traversal of a country extends naturally to colluding nation-states. One can simply use a modified database of country borders to flag all those in an alliance as a single "country." That this would result in a set of disjoint geographic polygons is of no concern to our algorithm; in fact, many countries are already made up of disjoint regions (for instance islands off of a country's coast).

5 DeTor Design

The previous section demonstrates how to prove, for a given circuit, whether a single round-trip of communication provably avoided a region (once or twice). Unfortunately, not all circuits can provide such proofs, even if they were to minimize latencies between all hosts. Trivially, any circuit with a Tor router in some region F cannot be used to avoid F. Subtler issues can also arise, such as when two consecutive hops on a circuit are in direct line-of-sight of a forbidden region.

In this section, we describe how DeTor identifies

which circuits could *possibly* provide a proof by alibi, and how we choose from among them to maximize both anonymity and likelihood of success.

5.1 Identifying Potential DeTor Circuits

Alibi Routing identifies potential alibi peers through a sophisticated overlay routing protocol in which peers assist one another in finding alibis. This is necessary in Alibi Routing because no one peer knows all other peers. Fortunately, Tor's design includes downloading a list of all Tor routers, so we can search for alibi circuits without requiring any explicit assistance from Tor hosts, and thus without requiring any modifications to Tor clients or the Tor protocol in general.

A DeTor peer first downloads the list of all Tor routers. This includes many pieces of information about each router, including its name, IP address, port, public key, and typically also which country it is in. We make the simplifying assumption that we can also obtain each Tor router's GPS coordinates. We envision two ways this information could be made available: First, we can use publicly available IP geolocation databases that map IP addresses to locations [25, 31]. Second, we could augment the Tor protocol to allow routers to include their locations (perhaps within some privacy-preserving range) in the public list of Tor routers.[2] For never-once, as with Alibi routing, we trust the nodes to be honest so long as they can be proved to be outside the forbidden region (§3); for such nodes, we trust the GPS coordinates they self-report.

If we have the latitude and longitude of each Tor router as well as the source and destination, then we can determine if a circuit has the potential to offer proof of avoidance by replacing R_{e2e} in Eqs. (4) and (5) with the shortest possible RTT (by two-thirds the speed of light) through the circuit. This is in essence testing whether, in the best case scenario, it would be possible to obtain a proof of avoidance. DeTor considers all circuits that meet these criteria as *potential DeTor circuits*.

Alternatively, if precise GPS coordinates are not available, we can assume that we do not have exact GPS locations, but only which country each router is in (as Tor currently reports). In this case, we redefine $D_{\mathsf{min}}(x_1,\ldots,x_n)$ to be the shortest sum distance from any point in x_1's country to any point in x_2's country to any point in x_3's country, and so on.

Armed with a set of potential DeTor circuits, we next address the question: which among them should we choose?

[2] We require that Tor routers not move significant distances between the time that a client obtains their GPS coordinates through the time the client uses those routers.

5.2 Choosing Circuits

There are two key considerations in choosing from among potential DeTor circuits:

First, the circuit should have a high likelihood of actually providing proofs of avoidance. Satisfying the above alibi conditions are necessary but not sufficient to truly offer proof of avoidance. If any host on the circuit has very high latencies (e.g., because their last-mile link is a satellite or cellular link [32]), then we will never be able to definitively prove with RTT measurements alone where their packets could not have gone.

It is difficult to determine whether there are such high-latency links without directly measuring them. However, as multiple prior studies have shown, there is a strong correlation between distance and RTT [6, 1], with very long distances typically resulting in significantly larger departures from the minimum speed-of-light propagation time. Therefore, as a first approximation, we seek to choose very distal legs less often than shorter legs. We must be cautious here: using very short legs, while likely to offer successful proofs of avoidance, runs the risk of choosing Tor routers within the same administrative domain, violating the goal of having three (or more) distinct routers on the circuit. To address this, DeTor can optionally take a parameter Δ representing a desired minimum distance between any two routers on the circuit. Note that this naturally captures Tor's policy of never choosing two hosts on the same subnet.

This brings us to our second consideration: the chosen circuit should be chosen randomly, minimizing the difference in probabilities of choosing one node (or administrative domain) over another. Tor's circuit selection provides greater weight to nodes with greater bandwidth; we incorporate this with our desire for higher likelihood of success (lower latencies). After filtering the circuits that can never provide us with an alibi, as well as filtering the circuits based on minimum distance Δ, we then choose from all remaining circuits with probability weighted in favor of higher bandwidth and lower latency.

5.3 Constructing and Using Circuits

DeTor makes use of Tor's transport plug-ins to guide circuit construction without requiring any modifications to the Tor client. In particular, DeTor uses the Stem [36] Tor controller for constructing circuits and attaching TCP connections to them.

Much like Alibi Routing, DeTor must check for proofs of avoidance for every round-trip of communication. Half of this is provided by Tor's checks that the packets followed the circuit and were not altered; additionally, a DeTor client measures the end-to-end RTT for each round-trip of communication and checks this against

Eq. (4) for never-once avoidance and/or Eq. (5) for never-twice. A natural question is: what should DeTor do when a round-trip of communication does not provide proof, for instance because the end-to-end RTT is too high? For never-once avoidance, we believe that this is an application-specific concern. Some applications may wish not to accept any packet that might have traversed a forbidden region, and so they may drop these packets. Other applications may accept some rounds of communication without proof, particularly if the data in them had end-to-end verification or if it were not sensitive. This is an interesting area of future work.

However, when DeTor is used for never-twice avoidance, it is critical that not too many packets be sent if there is the possibility that they doubly traversed an adversary. After a round-trip of communication fails to obtain proof, it may be useful for the source node to try to trick the adversary by inserting a random number of packets that terminate at the middle node. Such defenses are another interesting area of future work; in the remainder of this paper, we focus primarily on how often we are able to obtain proof of avoidance, and the quality of the circuits that provide such proof.

6 Evaluation

In this section, we present the evaluation of DeTor in terms of both never-once and never-twice avoidance.

Our evaluation is driven by several fundamental questions: who can avoid whom, does provable avoidance harm anonymity, what is the performance of the circuits that DeTor provides, and what are the primary indicators of DeTor's success (or failure)?

6.1 Experimental Setup

We evaluate DeTor in simulation, using a Tor latency dataset collected with the Ting measurement tool [6]. As a brief overview, Ting performs active measurements of the Tor network and, through a novel sequence of circuits, is able to directly measure the RTT between *any* two active Tor relays. This measured RTT between two Tor relays contains the forwarding delays, which includes Tor's crypto operations. As a part of this work, Cangialosi et al. [6] released a dataset comprising RTTs between all pairs of a set of 50 Tor relays spread throughout the world.[3] Also included in this dataset is the GPS location of all 50 nodes obtained from a publicly available IP geolocation database [31] (as measured at the time of their study). Although we used this static database for our evaluation, the DeTor design assumes that a client can obtain Tor relays' GPS coordinates, through one of several means discussed in §5.1.

[3]For seven pairs of nodes, the RTT was reported as `'Error'`. For these, we assume an RTT of 10 seconds; this is surely greater than their real RTT, and so it strictly puts our results at a disadvantage.

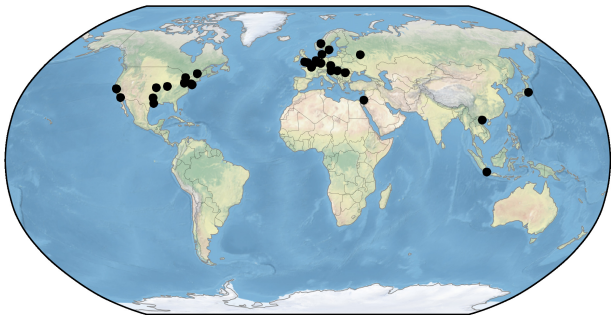


Figure 3: Locations of the 50 nodes used in our evaluation: a subset of real Tor nodes, as provided in the Ting [6] dataset.

Figure 3 shows the position of the Tor routers we use in our study. Note that, like real Tor deployments, it is skewed towards North America and Europe.

We simulate DeTor by using Ting data as a stand-in for both `ping` (when establishing the set of trusted Tor relays; see §3) and for end-to-end RTTs of the circuit. Recall that we only use these RTT measurements when determining if a chosen circuit successfully provides provable avoidance. Conversely, when we compute whether a circuit could *possibly* offer avoidance, we rely only on distances (computed using great-circle distance over the relays' GPS coordinates) and the two-thirds speed of light propagation of data. For the purpose of the simulation, the source and destination are also Tor nodes from the selected Tor nodes set. For never-once, we construct candidate circuits by selecting all possible permutations of three nodes from this set. For never-twice, we construct candidate circuits in the same way, but due to the additional computation needed, we only evaluate a random sample of the candidate circuits (1000 circuits per source-destination pair).

For never-once avoidance, we attempt to avoid several countries identified as having performed censorship [33]: China, India, PR Korea, Russia, Saudi Arabia, and Syria. Also, to see how well DeTor avoids countries with high "routing centrality" [22], we also attempt to avoid some countries that are on many paths: Japan and the US.

6.2 Never-Once

6.2.1 Who can avoid whom?

We begin by evaluating how successfully DeTor can find circuits to provably avoid various regions of the world. Figure 4 shows DeTor's overall success rate for each different forbidden country and δ values ranging from 0 to 1. Each stacked histogram represents the frac-

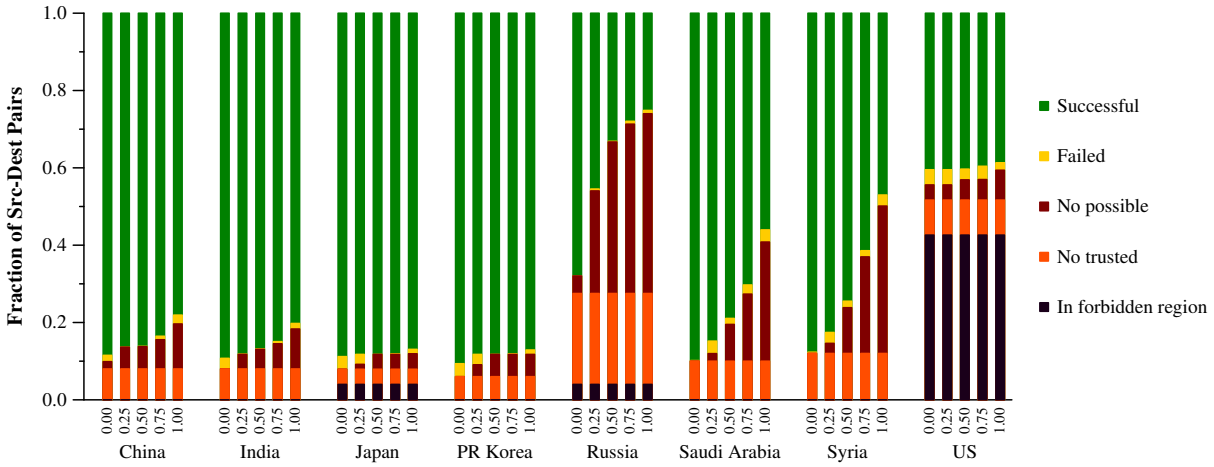


Figure 4: DeTor's success at never-once avoidance, and reasons for failure, across multiple choices of forbidden regions and δ. Overall, DeTor is successful at avoiding all countries, even those prevalent on many paths, like the US.

tion of all source-destination pairs who (from bottom to top): (1) terminate in the forbidden region and therefore cannot possibly achieve avoidance, (2) do not have any trusted nodes, typically because they are too close to the forbidden region to ensure that anyone they are communicating with is not in it, (3) have trusted nodes but no circuits that could possibly provide provable avoidance, (4) have circuits that could theoretically avoid the forbidden region, but none that do with real RTTs, and (5) successfully avoid the forbidden region over at least one DeTor circuit.

The key takeaway from this figure is that DeTor is generally successful at finding at least one DeTor circuit for all countries and all values of δ. We note two exceptions to this: Russia can only be avoided by approximately 35% of all source-destination pairs when $\delta = 0.5$. We believe this is due to the fact that Russia is close to the large cluster of European nodes in our dataset.

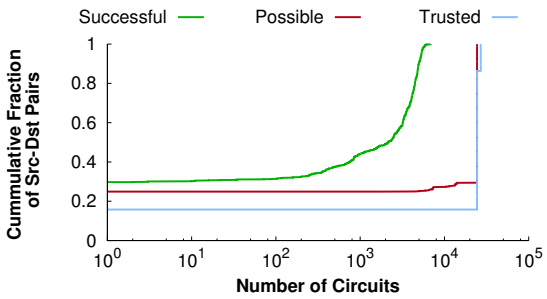
The US is another example of somewhat lower success rate; this is due, again, to our dataset comprising many nodes from the US, and thus 45% of all pairs in our dataset cannot possibly avoid the US. However, of the remaining source-destination pairs who do not already terminate in the US, 75% of them can successfully, provably avoid the US. We find this to be a highly encouraging result, particularly given that the US is on very many global routes on the Internet. We note that this is a higher avoidance rate than Alibi Routing was able to achieve; we posit that this is because DeTor uses longer circuits, thereby allowing it to maneuver around even nearby countries by first "stepping away" from them. Investigating the quality of longer DeTor circuits is an interesting area of future work.

We also observe from Figure 4 that larger values of δ lead to lower likelihoods of avoidance, as expected. This is particularly more pronounced with Russia, Syria, and Saudi Arabia; we believe that this, too, is because these countries are near the cluster of European nodes. Interestingly, this impact is least pronounced with the more routing-central adversaries we tested (Japan and the US). Some have proposed defense mechanisms that introduce packet forwarding delays in Tor [9, 5, 20]; these results lend insight into how these defenses would compose with DeTor. In particular, note that increasing δ in essence simulates greater end-to-end delays, which these defense mechanisms would introduce. Thus, with greater delay (intentional or not), DeTor experiences a lower likelihood of providing proofs of avoidance.

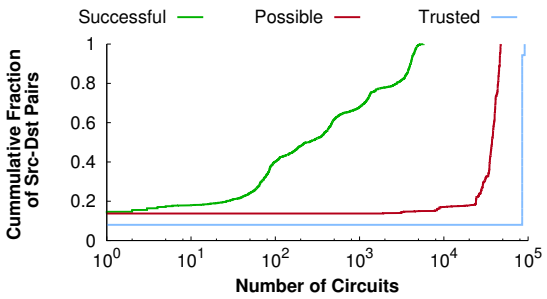
Number of DeTor circuits The above results show that we are successful at identifying *at least one* DeTor circuit for most source-destination pairs. We next look at *how many* DeTor circuits are available to each source-destination pair.

Figure 5 shows the distribution, across all source-destination pairs in our dataset, of the number of circuits that (1) offered successful never-once avoidance, (2) were estimated to be possible (but may not have achieved avoidance with real RTTs), and (3) were trusted, but not necessarily estimated to be possible. We look specifically at the number of circuits while attempting to avoid the US and China, with $\delta = 0.5$.

This result shows that approximately 30% of the source-destination pairs were only able to successfully use a single circuit while avoiding the US; 18% of pairs avoiding China had a single circuit. Fortunately, the majority had much more: avoiding the US, the median source-destination pair has over 1,000 successful circuits

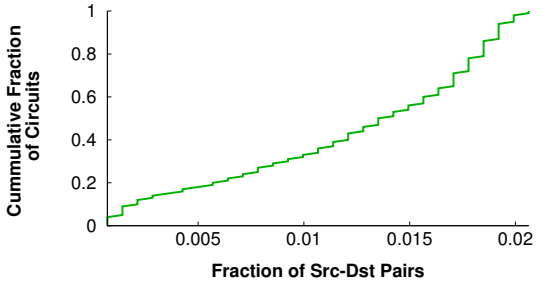


(a) US is the forbidden region.

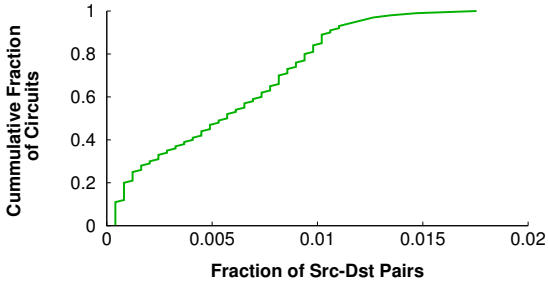


(b) China is the forbidden region

Figure 5: The distribution of the number of circuits that DeTor is able to find while avoiding (a) US and (b) China, with $\delta = 0.5$. Some source-destination pairs get only a single DeTor circuit, but the majority get 500 or more.



(a) US is the forbidden region.



(b) China is the forbidden region

Figure 6: The distribution of the fraction of source-destination pairs for which a given circuit successfully provides provable avoidance. ($\delta = 0.5$)

at its disposal; when avoiding China, this number is 500.

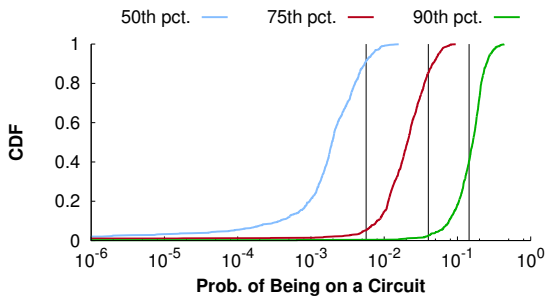
Even the most successful source-destination pairs tend to have far fewer successful circuits than "trusted" Tor circuits. These results allow us to infer how well Tor's current policies work. Recall that, in today's Tor, users can specify a set of countries from which they wish not to choose relays on their circuit. This is similar to the "trusted" line in the plots of Figure 5 (in fact, because we actually test the ping times to verify that it is not in the country, Tor's policy is even more permissive). This means that roughly 88% of the time (comparing the successful median to the trusted median), Tor's approach to avoidance would in fact not be able to deliver a proof of avoidance. It is in this sense that we say that Tor offers its users merely the illusion of control.

All together, these results demonstrate the *power* of DeTor—simply relying on random chance is highly unlikely to result in a circuit with provable avoidance.

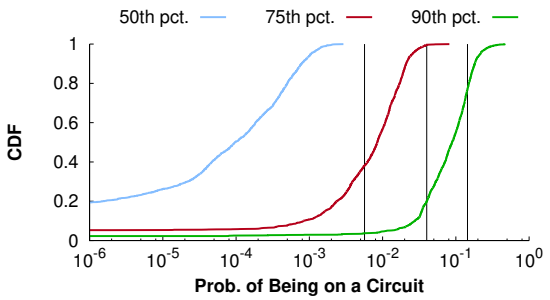
Given that there are source-destination pairs that have only a handful of DeTor circuits, we ask the converse: are there some circuits that offer avoidance for only a small set of source-destination pairs? If so, then this opens up potential attacks wherein knowing the circuit could uniquely identify the source-destination pair using that circuit. To evaluate this, we show in Figure 6 the distribution of the fraction of source-destination pairs for which a given circuit successfully provides proof of avoidance. The median circuit achieves provable avoidance to only 1.4% of source-destination pairs avoiding the US; 0.6% when avoiding China. These numbers are lower than desired (compare them to standard Tor routing for which nearly 100% of circuits are viable), but we believe they would be more reasonable in practice, for two reasons: First, the Ting dataset we use is not representative of the kind of node density that exists in the Tor network; in collecting that dataset, the experimenters explicitly avoided picking many hosts that were very close to one another, yet proximal peers are common in Tor. Second, in our simulations, we choose our source and destinations from the Tor nodes in our dataset; in practice, clients and destinations represent a far larger, more diverse set of hosts, and thus, we believe, would make it much more difficult to deanonymize.

6.2.2 Circuit diversity

Having many circuits is not enough to be useful in Tor; it should also be the case that there is diversity among the set of hosts on the DeTor circuits. Otherwise,

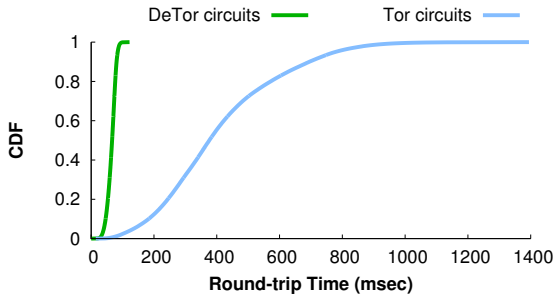


(a) US is the forbidden region.

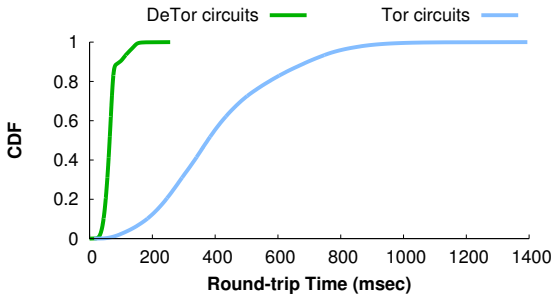


(b) China is the forbidden region

Figure 7: Distribution of the 50th, 75th, and 90th percentile probabilities of a node being selected to be on a circuit, taken across all DeTor circuits across source-destination pairs. Vertical lines denote these same percentiles across all Tor circuits. DeTor introduces only a slight skew, preferring some nodes more frequently than usual. ($\delta = 0.5$)



(a) US is the forbidden region.



(b) China is the forbidden region

Figure 8: The distribution of round-trip times for DeTor circuits ($\delta = 0.5$) and regular Tor circuits. Because avoidance becomes more difficult with higher-RTT circuits, DeTor's successful circuits tend to have lower RTTs.

popular Tor routers may become overloaded, and it becomes easier to predict which Tor routers will be on a circuit, thereby potentially opening up avenues for attack. We next turn to the question of whether the set of circuits that DeTor makes available disproportionately favor some Tor routers over others.

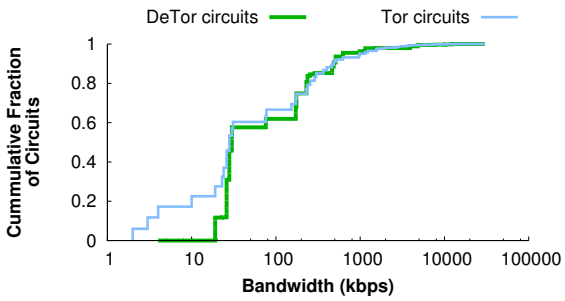
To measure how equitably DeTor chooses available Tor relays to be on its circuits, we first compute, for each source-destination pair, the probability distribution of each Tor relay appearing on a successful DeTor circuit. Figure 7 shows the distribution of the 50th, 75th, and 90th percentiles across all source-destination pairs. As a point of reference, the vertical lines represent these same percentiles for Tor's standard circuit selection (recall that Tor does not choose nodes uniformly at random, but instead weights them by their bandwidth).

We find that DeTor's median probability of being chosen to be in a circuit is less than normal, as evidenced by the 50th percentile curve being almost completely less than the 50th percentile spike. When avoiding the US, there is a slight skew towards more popular nodes, as evidenced by the 75th percentile also being less than normal. When avoiding China, on the other hand, DeTor's 90th percentile is typically less than Tor's, indicating that DeTor more equitably chooses nodes to be on its circuits.

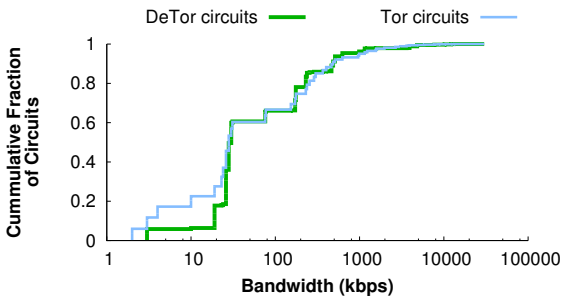
It is true that DeTor may result in load balancing issues, especially if Tor routers are not widely geographically dispersed – this is fundamental to DeTor: after all, if many users are avoiding the US, then all of this load would have to shift from the US to other routers. However, as shown in Figure 7, while DeTor does introduce some node selection bias, it is within the skew that Tor itself introduces.

6.2.3 Circuit performance

We investigate successful DeTor circuits by their latency and expected bandwidth. Figure 8 compares the distribution of end-to-end RTTs through successful DeTor circuits to the RTT distribution across *all* Tor circuits in our dataset. DeTor circuits have significantly lower RTTs—on the one hand, this is a nice improvement in performance. But another way to view these results is that DeTor precludes selection of many circuits, predominately those with longer RTTs. For some source-destination-forbidden region triples, this is a necessary byproduct of the fact that we are unlikely to be able

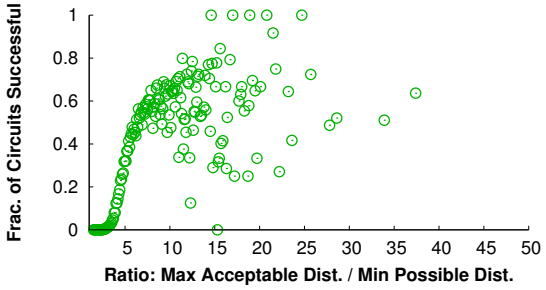


(a) US is the forbidden region.

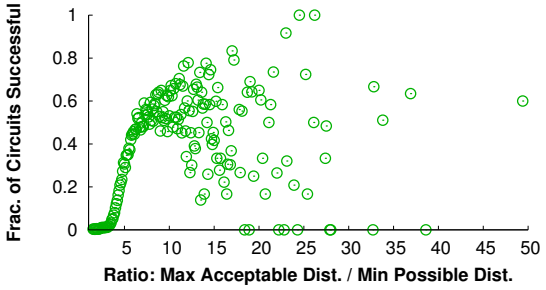


(b) China is the forbidden region

Figure 9: The distribution of minimum bandwidth for DeTor circuits ($\delta = 0.5$) and regular Tor circuits.



(a) US is the forbidden region.



(b) China is the forbidden region

Figure 10: Success rates for never-once circuits as a function of the ratio between the maximum acceptable distance (through the circuit but not through F) and the minimum distance (directly through the circuit). This shows a positive correlation, indicating that it is feasible to predict which circuits will be successful. ($\delta = 0.5$)

to get proofs of avoidance if we must traverse multiple trans-oceanic links. In these examples, China has access to some circuits with longer RTTs, because it is farther away from many of our simulated hosts than is the US.

Figure 9 compares bandwidths of DeTor and Tor circuits. For each circuit, we take the minimum bandwidth, as reported by Tor's consensus bandwidths. Here, we see largely similar distributions between DeTor and Tor, with Tor having more circuits with lower bandwidth. We suspect that those lower-bandwidth hosts that Tor makes use of may also have higher-latency links, therefore making them less likely to appear in DeTor circuits.

6.2.4 Which circuits are more likely to succeed?

As Figure 5 showed, it is not uncommon for there to be one to two orders of magnitude more circuits that meet the *theoretical* requirements for being a DeTor circuit than there are circuits who achieve avoidance in practice. In a deployed setting, a client would ideally be able to identify which circuits are more likely to work before actually going through the trouble of setting up the connection and attaching a transport stream to it.

As a predictor for a circuit's success for never-once avoidance, we take the ratio of the maximum acceptable distance (how far the packet could travel without traversing the circuit *and* the forbidden region) to the minimum possible distance (the direct great-circle distance through the circuit). Our insight is that, the larger this ratio is, the more "room for error" the circuit has, and the more resilient it is to links whose RTTs deviate from the two-thirds speed of light.

Figure 10 shows this ratio corresponds to the fraction of theoretically-possible circuits that achieve successful avoidance. As this ratio increases from 0 to 10, there is a clear positive correlation with success. However, with large ratio values, the relationship becomes less clear; this is largely due to the fact that large ratio values can be a result of very small denominators (the shortest physical distance).

These results lend encouragement that clients can largely determine *a priori* which circuits are likely to provide provable avoidance. Exploring more precise filters is an area of future work.

6.3 Never-Twice

6.3.1 How often does never-twice work?

Recall that, unlike never-once, there are no forbidden regions explicitly stated *a priori* with never-twice. Therefore, to evaluate how well never-twice works, we measure the number of source-destination pairs that yield a successful DeTor circuit.

Ruling out the source-destination pairs who are in the same country (as these can never avoid a double-transit), we find that *98.6% of source-destination pairs can find at least one never-twice DeTor circuit*. This is a very promising result, as it demonstrates that simple client-side RTT measurements may be enough to address a wide range of attacks. In the remainder of this section, we investigate the quality of the circuits that our never-twice avoidance scheme finds.

Turning once again to the number of circuits, Figure 11 compares the number of circuits that DeTor identified as *possibly* resulting in a proof of avoidance (as computed using Eq. (5), and those that were successful given real RTTs. Never-twice circuits tend to succeed with approximately 5× the number of circuits that never-once receives. This demonstrates how fundamentally different these problems are, and that our novel approach of computing "forbidden" countries on the fly (as opposed to some *a priori* selection of countries to avoid with never-once) results in greater success rates.

6.3.2 Circuit diversity

We turn again to the question of how diverse the circuits are; are some Tor relays relied upon more often than others when achieving never-twice avoidance?

Figure 12 shows the percentile distribution across all successful never-twice DeTor circuits. Compared with never-once (Fig. 7), never-twice circuits fall even more squarely within the distribution of normal Tor routers (the vertical spikes in the figures). In particular, the top 10% most commonly picked nodes appear roughly as often as Tor's top 10% (the median 90th percentile is almost exactly equal to Tor's 90th percentile). The median node is slightly less likely to be selected than in Tor, indicating only a small skew to more popular nodes.

6.3.3 Circuit performance

We investigate successful DeTor never-twice circuits, once more turning to latency and expected bandwidth. Figure 13 compares successful never-twice DeTor circuits' RTTs to those of Tor. Compared to never-once (Fig. 8), there are never-twice DeTor circuits with greater RTTs. We conclude from this that never-twice avoidance has the ability to draw from a more diverse set of links—particularly when the entry and exit legs are farther from one another.

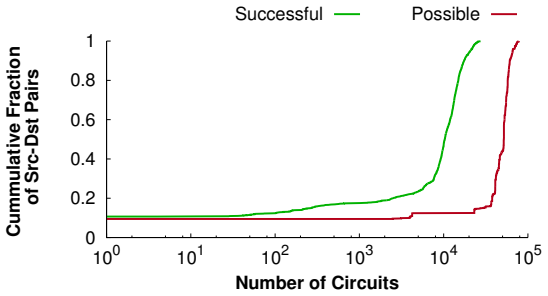


Figure 11: The distribution of the number of circuits that DeTor is able to find with never-twice ($\delta = 0.5$).

Figure 14 reinforces our finding that never-twice has access to a larger set of routers, as the distribution of successful never-twice bandwidths matches those of the full Tor network much more closely.

6.3.4 Which circuits are more likely to succeed?

We close by investigating what influences a never-twice DeTor circuit's success or failure. Figure 15 shows the fraction of possible never-twice circuits that were found to be successful, and plots them as a function of $D(e,m,x)/D(s,e,m,x,t)$. This ratio represents how much of the overall circuit's length is attributable to the middle: that is, everything but the entry and exit legs.

There are several interesting modes in this figure that are worth noting. When this ratio on the x-axis is very low, it means that almost the entire circuit is made up of the entry and exit legs, and therefore they are very likely to intersect—as expected, few circuits succeed at this point. DeTor succeeds more frequently as the middle legs take on a larger fraction of the circuit's distance, but then begins to fail as the length of the middle legs approaches the combined length of the entry and exit legs. This is because, in our dataset, when the middle legs are approximately as long as the entry and exit legs, this tends to correspond to circuits made out of the two clusters of nodes: one in North America and the other in Europe. Because these clusters are tightly packed, the probability of intersecting entry and exit legs increases. This probability of intersection decreases when the circuits no longer come from such tightly packed groups.

When the middle legs dominate the circuit's distance (the ratio in the figure approaches one), we again enter a particular regime in our dataset: These very high ratio values correspond to circuits with source and entry node both in North America (or in Europe), and with middle legs that traverse the Atlantic (and then return).

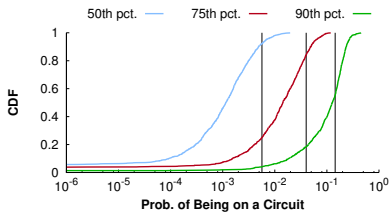


Figure 12: Nodes' probability of being on a successful never-twice DeTor circuit. ($\delta = 0.5$)

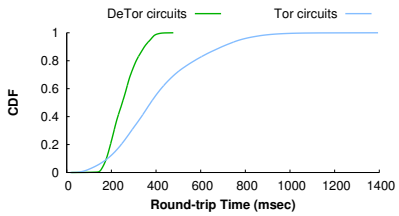


Figure 13: Round-trip times for never-twice DeTor circuits ($\delta = 0.5$) and regular Tor circuits.

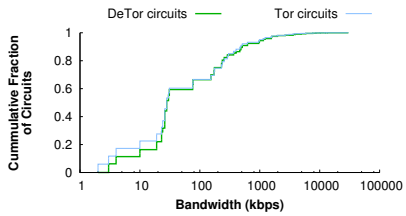


Figure 14: Bandwidth for never-twice DeTor circuits ($\delta = 0.5$) and regular Tor circuits.

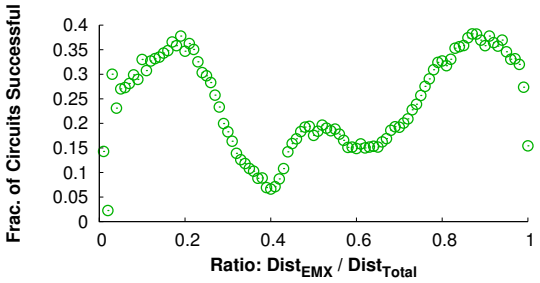


Figure 15: Success rate of never-twice DeTor circuits as a function of how long the (e,m,x) legs of the circuit are.

In other words, to accommodate such long middle legs, the source and entry node (and exit node and destination) are forced into the same cluster (either North America or Europe), which again increases the chances of intersection.

In sum, for never-twice, DeTor interestingly prefers circuits that have middle legs that are disproportionately large or small relative to the entry and exit legs. However, this may be dependent on the node locations from the Ting dataset we use, as the overall success rate of never-twice avoidance depends on the geographical diversity of where Tor routers are located.

7 Conclusion

In this paper, we have presented techniques that allow end-users to *provably* verify when packets over their Tor circuits have avoided traversing a geographic region once or twice. Our system, DeTor, builds upon prior work on provable avoidance routing [24], and extends it (1) to work over Tor's multiple hops, and (2) to achieve "never-twice" avoidance. Through extensive simulations using real Tor latency data [6], we have demonstrated that DeTor achieves provable avoidance for most source-destination pairs, even when avoiding large, routing-central countries like the United States.

Although the dataset we use in evaluating DeTor comes from live Tor measurements [6], the scale and geographic positions do not reflect the Tor network in its entirety; our results indicate that having more Tor routers would lead to more potential DeTor circuits and greater overall success rates. As with any such system, the best evaluation would be a longitudinal study with real users on the Tor network; this would be an interesting area of future work.

This paper is the first step towards bringing provable avoidance to Tor, but we believe that DeTor has the potential to be a powerful building block in future defenses against censorship and deanonymization of Tor.

Our code and data are publicly available at:

`https://detor.cs.umd.edu`

Acknowledgments

We thank Neil Spring, Matt Lentz, Bobby Bhattacharjee, the anonymous USENIX Security reviewers, and our shepherd, Prateek Mittal, for their helpful feedback. This work was supported in part by NSF grants CNS-1409249 and CNS-1564143.

References

[1] S. Agarwal and J. R. Lorch. Matchmaking for online games and other latency-sensitive P2P systems. In *ACM SIGCOMM*, 2009.

[2] M. Akhoondi, C. Yu, and H. V. Madhyastha. LASTor: A low-latency AS-aware Tor client. In *IEEE Symposium on Security and Privacy*, 2013.

[3] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable Internet Protocol (AIP). In *ACM SIGCOMM*, 2008.

[4] Anonymous. The collateral damage of Internet censorship by DNS injection. *ACM SIGCOMM Computer Communication Review (CCR)*, 42(3):21–27, 2012.

[5] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg. A systematic approach to developing and evaluating website fingerprinting defenses. In *ACM Conference on Computer and Communications Security (CCS)*, 2014.

[6] F. Cangialosi, D. Levin, and N. Spring. Ting: Measuring and exploiting latencies between all Tor nodes. In *ACM Internet Measurement Conference (IMC)*, 2015.

[7] R. Clayton, S. J. Murdoch, and R. N. Watson. Ignoring the great firewall of China. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2006.

[8] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, 2004.

[9] K. Dyer, S. Coull, T. Ristenpart, and T. Shrimpton. Peek-a-Boo, I still see you: Why efficient traffic analysis countermeasures fail. In *IEEE Symposium on Security and Privacy*, 2012.

[10] M. Edman and P. Syverson. AS-awareness in Tor path selection. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[11] A. Edmundson, R. Ensafi, N. Feamster, and J. Rexford. Characterizing and avoiding routing detours through surveillance states. `https://arxiv.org/pdf/1605.07685.pdf`, May 2016.

[12] A. Edmundson, R. Ensafi, N. Feamster, and J. Rexford. A first look into transnational routing detours. In *ACM SIGCOMM (Poster)*, 2016.

[13] R. Ensafi, P. Winter, A. Mueen, and J. R. Crandall. Analyzing the great firewall of China over space and time. In *Privacy Enhancing Technologies Symposium (PETS)*, 2015.

[14] N. S. Evans, R. Dingledine, and C. Grothoff. A practical congestion attack on Tor using long paths. In *USENIX Security Symposium*, 2009.

[15] Y. Gilad and A. Herzberg. Spying in the dark: TCP and Tor traffic analysis. In *Privacy Enhancing Technologies Symposium (PETS)*, 2012.

[16] Global Administrative Areas (GADM) Database. `http://www.gadm.org`.

[17] N. Hopper, E. Y. Vasserman, and E. Chan-Tin. How much anonymity does network latency leak? *ACM Transactions on Information and System Security (TISSEC)*, 13(2):13, 2010.

[18] A. Houmansadr, C. Brubaker, and V. Shmatikov. The parrot is dead: Observing unobservable network communications. In *IEEE Symposium on Security and Privacy*, 2013.

[19] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson. Users get routed: Traffic correlation on Tor by realistic adversaries. In *ACM Conference on Computer and Communications Security (CCS)*, 2013.

[20] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright. Toward an efficient website fingerprinting defense. In *European Symposium on Research in Computer Security (ESORICS)*, 2016.

[21] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer. Decoy routing: Toward unblockable Internet communication. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2011.

[22] J. Karlin, S. Forrest, and J. Rexford. Nation-state routing: Censorship, wiretapping, and BGP. `http://arxiv.org/pdf/0903.3218.pdf`, Mar. 2009.

[23] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas. Circuit fingerprinting attacks: Passive deanonymization of Tor hidden services. In *USENIX Annual Technical Conference*, 2015.

[24] D. Levin, Y. Lee, L. Valenta, Z. Li, V. Lai, C. Lumezanu, N. Spring, and B. Bhattacharjee. Alibi routing. In *ACM SIGCOMM*, 2015.

[25] MaxMind GeoIP2 Databases. `https://www.maxmind.com/en/geoip2-databases`.

[26] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg. SkypeMorph: Protocol obfuscation for Tor bridges. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[27] H. M. Moghaddam, T. Reidl, N. Borisov, and A. Singer. I want my voice to be heard: IP over voice-over-IP for unobservable censorship circumvention. In *Network and Distributed System Security Symposium (NDSS)*, 2013.

[28] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of Tor. In *USENIX Security Symposium*, 2005.

[29] S. J. Murdoch and P. Zieklński. Sampled traffic analysis by Internet-exchange-level adversaries. In *Workshop on Privacy Enhancing Technologies (PET)*, 2007.

[30] J. Naous, M. Walfish, A. Nicolosi, M. Miller, and A. Seehra. Verifying and enforcing network paths with ICING. In *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2011.

[31] Neustar IP Geolocation. `https://www.neustar.biz/services/ip-intelligence`.

[32] R. Padmanabhan, P. Owen, A. Schulman, and N. Spring. Timeouts: Beware surprisingly high delay. In *ACM Internet Measurement Conference (IMC)*, 2015.

[33] Reporters Without Borders. Enemies of the Internet 2013 Report. `https://surveillance.rsf.org/en/wp-content/uploads/sites/2/2013/03/enemies-of-the-internet_2013.pdf`, Mar. 2013.

[34] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper. Routing around decoys. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[35] R. Sherwood, A. Bender, and N. Spring. DisCarte: A disjunctive Internet cartographer. In *ACM SIGCOMM*, 2008.

[36] Stem Controller Library. `https://stem.torproject.org`.

[37] Tor Metrics. `https://metrics.torproject.org`.

[38] Q. Wang, X. Gong, G. T. Nguyen, A. Houmansadr, and N. Borisov. CensorSpoofer: Asymmetric communication using IP spoofing for censorship-resistant web browsing. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[39] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh. StegoTorus: A camouflage proxy for the Tor anonymity system. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[40] C. V. Wright, S. E. Coull, and F. Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *Network and Distributed System Security Symposium (NDSS)*, 2009.

[41] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the network infrastructure. In *USENIX Security Symposium*, 2011.