Alibi Routing

Dave Levin[*] Youndo Lee[*] Luke Valenta[†] Zhihao Li[*] Victoria Lai[*]
Cristian Lumezanu[‡] Neil Spring[*] Bobby Bhattacharjee[*]

[*] University of Maryland [†] University of Pennsylvania [‡] NEC Labs

ABSTRACT

There are several mechanisms by which users can gain insight into where their packets have gone, but no mechanisms allow users undeniable proof that their packets did *not* traverse certain parts of the world while on their way to or from another host. This paper introduces the problem of finding "proofs of avoidance": evidence that the paths taken by a packet and its response avoided a user-specified set of "forbidden" geographic regions. Proving that something did *not* happen is often intractable, but we demonstrate a low-overhead proof structure built around the idea of what we call "alibis": relays with particular timing constraints that, when upheld, would make it impossible to traverse both the relay and the forbidden regions.

We present *Alibi Routing*, a peer-to-peer overlay routing system for finding alibis securely and efficiently. One of the primary distinguishing characteristics of Alibi Routing is that it does not require knowledge of—or modifications to—the Internet's routing hardware or policies. Rather, Alibi Routing is able to derive its proofs of avoidance from user-provided GPS coordinates and speed of light propagation delays. Using a PlanetLab deployment and larger-scale simulations, we evaluate Alibi Routing to demonstrate that many source-destination pairs can avoid countries of their choosing with little latency inflation. We also identify when Alibi Routing does not work: it has difficulty avoiding regions that users are very close to (or, of course, inside of).

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

SIGCOMM '15, August 17–21, 2015, London, United Kingdom

DOI: `http://dx.doi.org/10.1145/2785956.2787509`

Keywords

Alibi Routing; Provable route avoidance; Censorship avoidance; Peer-to-peer; Overlay routing

1. INTRODUCTION

Users have little control over where in the world their packets travel en route to their destinations. Some mechanisms exist to provide insight into where packets traveled, such as the record-route IP option, overlay routing systems (§7), or to a lesser extent source-routing. While these approaches expose a subset of the path the user's packets took, they do not allow a user to determine or provably influence where their packets do *not* go.

This paper introduces a new primitive we call *provable avoidance routing*. With provable avoidance routing, a user specifies arbitrary geographic regions—such as countries or UN voting blocs—to be avoided while communicating with a destination. If successful, the primitive returns *proof* that the user's packets did not traverse the forbidden regions. If it is unsuccessful, it concludes only that the packets *may have* traversed them.

The goal of provable avoidance routing is *detection*, as opposed to *prevention*. In other words, alone, it is unable to ensure a user's packets *will not* traverse a region of the world—we do not require modifications to the underlying routing protocols or hardware, and so we are subject to all of today's uncertainties as to where packets will travel. Rather, what we are able to provide is assurance that the user's packets and their respective responses took paths that *did not* traverse regions of the world. Our proofs of avoidance are provided on a per-packet basis, and are *a posteriori*: only after sending the packet and getting a reply can we ascertain whether or not the round-trip communication avoided the forbidden region.

While outright prevention would be ideal, detection can be a powerful tool, as well. For example, consider one of the greatest threats to open communication on the Internet: censorship. Beyond just dropping [34] or logging [29] users' traffic, censorship can take many forms, including *injecting* packets with false information [4]. Recent results indicate that many users may be censored not by their (or their destination's) countries, but by regimes through which their packets transit; a group of anonymous researchers demonstrated that DNS queries that merely traverse China's borders are subject to the same injection of false responses as if the queries came from one of its own citizens [4]. Incomplete deployment of authenticated protocols such as DNSSEC requires users to take other approaches, typically consisting of hiding packets' contents via encryption [10, 8], forwarding through hidden proxies [18, 46, 17], or applying steganography [11]. We offer an orthogonal approach: rather than use-and-confuse a censoring regime, we show that it is possible to simply *avoid the censor altogether*.

As another example of the usefulness of provable avoidance routing, two parties could perform Diffie-Hellman key exchange and use the proofs of avoidance to ensure that there could not have been a man-in-the-middle from user-specified forbidden regions. Subsequent communication after the initial key exchange would thus provide confidentiality even if the default route traversed the forbidden regions.

This paper makes two main technical contributions. The first is a means of proving that a packet avoided a forbidden region (§3). Our proofs of avoidance are built around the idea of using what we call "alibis": relays that are sufficiently far away from the forbidden region such that traversing both relay *and* forbidden region would result in a noticeably high delay.

The second contribution we make is the design and implementation of *Alibi Routing*, a peer-to-peer overlay routing system for finding alibis safely and efficiently (§4). Alibi Routing is secure in that, when tasked with finding alibis for a forbidden region F, it too avoids F (§5). It is efficient in that it requires a small amount of state, and takes few hops, especially when the source and destination are both reasonably far from the regions they seek to avoid. Most importantly, Alibi Routing is immediately and incrementally deployable: it requires no public key infrastructure (PKI) or modifications to existing routing protocols or switching hardware; it does not require synchronized clocks; and it does not require access to any information about the underlying routing topology of the Internet. Rather, it derives its security and proofs of avoidance from "a clock and a map": local measurements of round-trip times and a rough knowledge of one's own (and one's attacker's) GPS coordinates.

Using an implementation and deployment on PlanetLab (as well as large-scale simulations), we show that many source-destination pairs can avoid countries of their choosing with reasonably low latency inflation (§6). We also identify the instances when Alibi Routing does not work: in general, the closer a source or its destination is to a forbidden region F, the fewer potential alibis there are.

Alibi Routing is not a panacea; for instance, it is impossible for users to avoid the countries they are in—the very problem traditional censorship-resistant systems address. Our goal is not to replace such systems, but to complement them; as we will show, Alibi Routing offers an orthogonal set of properties that combine well with prior systems (§7).

2. GOALS AND NON-GOALS

In this section, we describe the goals (and non-goals) of a provable route avoidance primitive. Suppose that source s has sent a query to d and received a response, and s wishes to verify neither query nor response traversed some region of the world F. Our ultimate goal is to be able to construct a *proof* that s can check to make sure that the packet and its response *could not* have possibly traversed F.

Ideally, this primitive should be easy to deploy and use. To this end, we avoid modifications to existing routing protocols like BGP [50], or to hardware in the Internet [2, 28]. Rather, we show that users themselves can provide this service with an overlay protocol.

Provable route avoidance does *not* seek to provide two otherwise desirable properties. First, it does not seek to guarantee that an adversary would never see a *copy* of the packet. Even if the user is able to prove that the adversary was not on the packet's path, it does not stop nodes on the path from copying and later delivering the packets to the adversary. Existing approaches to anonymity (§7) can complement alibi routing to make copies less useful to a censor.

Second, alibi routing seeks to allow users to prove that a packet must have avoided F after the fact, not to guarantee that a packet will not traverse F. Higher-layer protocols must choose how to react to the absence of a proof, that is, the observation that the communication may have traversed F. Some may require that all packets avoid a part of the network: such applications should treat packets that might have traversed F as failures and retransmit. Other higher-layer protocols may permit some fraction of packets through F, for instance if they are using alibi routing for non-adversarial reasons, e.g., for performance or path diversity.

3. PROOFS OF AVOIDANCE

Here, we demonstrate how to prove that a packet and its response did not traverse a region of the world. In general, proving that some event x did *not* happen is very difficult. Our proof structure seeks to demonstrate that x did not happen because it would have been *impossible*. It consists of finding a set of events A such that:

- It can be proved that events in A *did* happen.
- A and x are mutually exclusive.

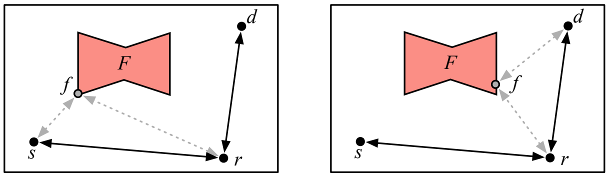
If these properties both hold, then x could not have happened: the events A serve as an *alibi* for x.

3.1 Mutually exclusive routing events

What then are the mutually exclusive events that would lead to provable route avoidance? In this setting, the event x that we wish to prove impossible is the event that a packet and its response from s to d transited forbidden region F. We need two pieces of evidence from A.

First, we must know a subset of the path that the packet took. To this end, a user forwards packets through a relay node r. r signs[1] the packet, and thus, if r can be trusted not to have shared his key, then this proves that the packet

[1] In fact, because we do not make use of digital signature's property of non-repudiation, a symmetric key MAC suffices.



(a) $R(s,r) + R(r,d) \ll \min_f \{R(s,f) + R(f,r)\} + R(r,d)$

(b) $R(s,r) + R(r,d) \ll R(s,r) + \min_f \{R(r,f) + R(f,d)\}$

Figure 1: s and d wish to communicate without their packets traversing geographic region F. Relay r can serve as an *alibi* if packets that would traverse any possible node f in F would induce noticeably higher RTTs, as captured in Eq. (1).

must have gone through r (we define our trust assumptions in Section 4).

Second, we must know that, for any possible path that includes s, r, and d, the packets could not have also gone through F. Of course, there are many ways the packet can traverse F: r could ignore the users' wishes and forward the packet through F, or F could lie on the path between s and r or r to d. In other words, proving a *subset* of the path a packet took is feasible, but how can we prove that no node from F was anywhere on the path?

The key idea is to choose a relay that is so distant from the forbidden region that transiting both would induce noticeably high delays. Figure 1 illustrates the idea. When s routes through r to get to d, it observes the round-trip times between itself and the relay, $R(s,r)$, and the relay and the destination, $R(r,d)$. It must ensure, for every packet, that this end-to-end latency is noticeably less than *lowest possible round-trip time* for any path that also traverses F. Concretely, in order to ensure that the packet did not traverse F, we must also demonstrate that *both* of the following inequalities hold:

$$\begin{aligned} R(s,r) + R(r,d) &\ll R(s,r) + \min_{f \in F} \{R(r,f) + R(f,d)\} \\ R(s,r) &\ll \min_{f \in F} \{R(s,f) + R(f,r)\} \end{aligned} \tag{1}$$

By "$x \ll y$", we mean "x is noticeably less than y," or more formally, that for some $\delta \geq 0$: $(1 + \delta) \cdot x < y$. Thus, the first inequality states that, if a packet goes through any f in the forbidden region on the path between the relay and the destination, then the increase in latency will be noticeable. The second inequality says the same for the path between the source and the relay. Note that s can compute both inequalities locally, without synchronized clocks: the min terms are estimates (based on the speed of light, §4), and though s cannot directly measure $R(r,d)$ in the first inequality, it can measure the end-to-end RTT, $R(s,r) + R(r,d)$, and its RTT to r, $R(s,r)$.

Suppose a relay has signed a packet (proving the packet traversed that relay), and it satisfies the timing constraints in Eq. (1) for any possible $f \in F$. These events are mutually exclusive to the packet traversing F, and thus we have our proof: the packet could not have possibly traversed F.

We call relays that yield such proof *alibis*. Note that, to be an alibi, it is necessary for a relay to be far from the forbidden region: if very close, then for any x there may be an f such that $R(r,x)$ is not noticeably different than $R(r,f) + R(f,x)$. However, simply being far away is *not sufficient* for a relay to be an alibi: if F were on the path from s to r, then no matter how far r is from F, Eq. (1) will not hold. As a result, locating alibis is non-trivial; in the next section, we describe one way to do it.

3.2 Practical considerations

How to obtain proof. Equation (1) asserts that no node f in the forbidden region could unnoticeably appear on the path. It would be unrealistic to identify and enumerate all *actual* hosts in a forbidden region—particularly when it is adversarial. We demonstrate in Section 4 how to use geographic distance to estimate the lowest *possible* round-trip time between two hosts.

When to obtain proof. Our proofs of avoidance are based on local measurements of round-trip times for packets. Latencies can vary over time—e.g., due to outages, route changes, or congestion [30]—and thus a relay that is a viable alibi at one point in time may not be one later, potentially even on a per-packet basis. As a result, our proofs of avoidance must be applied to *each* packet.

The factor of δ in the above equations helps insulate provable avoidance from latency fluctuations. δ represents a tradeoff between safety and efficiency; larger values of δ yield alibis who are so far away from a forbidden region that, if packets were to traverse both, there would be a very large increase in latency over a normal path through the alibi. Thus, with a large δ, one may be less likely to find a viable alibi, but that alibi is likely to work even in the face of variable round-trip times and congestion. Section 6 shows that we are successful in finding alibis for a range of δ values.

4. ALIBI ROUTING PROTOCOL

In this section, we describe Alibi Routing, a peer-to-peer overlay routing protocol for locating alibis. Once found, users forward their traffic through alibis, and apply the techniques from Section 3 to obtain proofs of avoidance. Alibi Routing is secure in that no routing messages are accepted unless they provably did not traverse the forbidden region the source node specified; we analyze its security properties in Section 5. Alibi Routing is efficient in that it finds relays quickly, without having to contact many intermediate hops; we evaluate its performance in Section 6.

4.1 Trust assumptions and attack model

Users query Alibi Routing by specifying (1) a destination with whom they wish to communicate, and (2) a geographic *forbidden region* F through which they want proof their packets do not traverse. For any peer who cannot be proved to be outside a user's specified forbidden region, we assume that it will act in a Byzantine faulty manner toward that user. The central assumption underlying Alibi Routing is that, *the user trusts all peers that are provably outside F to follow the protocol correctly*.[2].

The limitation is that it places the onus on users to determine where in the world their attackers are. As such, we expect Alibi Routing to be used mostly for avoiding large, very powerful adversaries [34, 37]. For example, Alibi Routing would be well-suited to avoid China's firewall, which appears to be run strictly within its borders [5, 4]. Attackers in our model can be routing-capable adversaries [37], i.e., we assume them to be capable of choosing how packets in their networks are routed, and of influencing routes to cause others' traffic to be routed through them. Fortunately, there has been significant work in identifying countries who launch such attacks, and ongoing efforts regularly identify new sources of misbehavior or malfeasance [33, 27, 7].

Even such powerful adversaries are faced with limitations, which Alibi Routing exploits. First, we make standard cryptographic assumptions; any scheme wherein an attacker cannot forge a MAC from a non-colluding peer suffices. Second, we make use of the fact that, while an attacker can lie about having greater latency to a victim, it cannot lie about having *lower* latency than it really has. This observation is commonly used in secure network coordinate systems [38, 43]. Finally, we apply the fact that information cannot travel faster than the speed of light, and that in fact most transmission media (especially fiber optic cables) peak at approximately $\frac{2}{3}c$. As we will demonstrate, these standard apparent impossibilities are sufficient for allowing many source-destination pairs to provably avoid various countries.

4.2 Query components

When a source node s wishes to find alibis, it constructs and forwards a query message, $\langle s, d, F, T\rangle$. Most of these are defined above: s and d are the source's and destination's IP addresses and ports, and F is the forbidden regions, represented by one or more ordered sets of $(\mathsf{lat}, \mathsf{lon})$ pairs. The forbidden regions are included in the query so that intermediate hops can determine which next-hop neighbors are *safe* to forward to.

The final component, T, is a set of what we call "target regions," which represent locations where alibis *might* reside. Target regions are included in queries to help guide routing towards parts of the network that make the most *progress* towards an alibi. Here, we describe how forbidden and target regions are represented; we then describe how peers forward them when routing.

4.2.1 User-specified forbidden regions

A forbidden region consists of a set of (possibly disjoint) polygons specified over a set of geographic $(\mathsf{lat}, \mathsf{lon})$ coordinates. A user wishing to avoid a particular country, for instance, can specify the country's borders. These are readily available in high precision online [13], but even an approximate circumscribing polygon can be calculated with a reasonably accurate map.

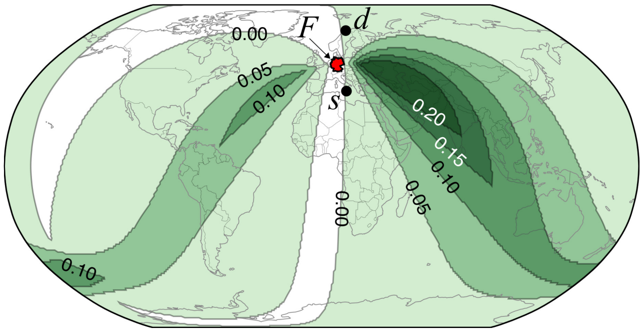


Figure 2: Example target regions, with end-hosts in Italy and Norway who seek to avoid Germany. The contours represent different values of δ in Eq. (2).

Each user can specify his or her own forbidden regions; Alibi Routing is agnostic to what these regions represent. This has the benefit that it supports a wide range of policies—users may choose to avoid cities where data logging facilities are expected to reside [29], an entire country, a UN voting bloc, and so on. Moreover, users can specify these policies without having to understand the underlying network topology: they only need to know where in the world those they do not trust reside.

4.2.2 Computed target regions

The final component of an Alibi Routing query is a set of *target regions*: geographic regions wherein alibis *may* exist. When a peer processes a query, its task is to choose next-hop neighbors who get the query closer to a target region (we describe this process in Section 4.4.2). As a result, the crucial property of a target region is that it include as many alibis as possible.

Similar to forbidden regions, target regions are represented by polygons of GPS coordinates. A node at GPS coordinate g is included in the target region if it satisfies the alibi conditions from Section 3. That is, if $D(\cdot,\cdot)$ represents the great-circle distance between two points, then g is in the target region if and only if:

$$\begin{aligned}(1+\delta)\cdot D(s,g) &< \min_{f\in F}\left\{D(s,f)+D(f,g)\right\}, \text{ and}\\ (1+\delta)\cdot D(g,d) &< \min_{f\in F}\left\{D(g,f)+D(f,d)\right\} \qquad (2)\end{aligned}$$

for some suitably large constant δ: this is the same "noticeably-larger-than" relationship as captured in Eq. (1)[3].

For an arbitrary set of forbidden regions, we do not know of a closed form solution to represent all GPS points in T. Instead, we segment the globe into a grid of points (in our implementation, we take $(\mathsf{lat}, \mathsf{lon})$ points at 2° intervals). For each such point g, we use Snell's law [12] to determine the r.h.s. of Eq. (2). If three contiguous grid points are in the target region and form a triangle, we add the entire triangle to T and take their union, forming a smaller set of polygons.

[2] This particular assumption need apply only to our protocol for finding alibis, and not to the proofs of avoidance (§3).

[3] The formulations of these two equations are slightly different; this is because s can accurately *estimate* the distance between relay and destination, $D(g, d)$, required in Eq. (2), but can directly *measure* only the full end-to-end RTT, $R(s, r) + R(r, d)$, required by Eq. (1).

Figure 2 illustrates the target region for a peer in Italy wishing to communicate with a peer in Norway while avoiding Germany. Note that larger values of δ result in smaller target regions; when $\delta = 0$, nearly the entire world has the potential of hosting an alibi, but when $\delta = 0.2$, alibis can only possibly be located in the Middle East, extending eastward to India.

The grid interval spacing represents a trade-off between efficiency and accuracy. With larger grids, the computation is faster and the target region's polygons can be represented with fewer points, thereby decreasing query size. However, larger grids can be inaccurate: they may miss viable relays, and, if left unchecked, could include portions of the forbidden region. To alleviate this second concern, we include each triangle in T only if it does not intersect the forbidden region. We have found grid intervals of 2° to be safe and efficient for all single-country forbidden regions we tested.

Target regions may (and often do) also include peers who are *not* viable relays. Consider for example a peer who has a satellite link with extremely high latencies: such a peer may never satisfy the alibi conditions, regardless of the path his or her packets take. In other words, a benevolent relay whose packets never traverse the forbidden region might never be viable simply because it has poor connectivity. With respect to safety, this is not a concern: target regions are used only to guide queries toward potential alibis, and Alibi Routing peers check the alibi conditions (Eq. (1)) to verify a relay's actions for each packet. We evaluate how likely a node is to be an alibi given that it is inside a target region in Section 6.

4.3 Neighbor maintenance

Every peer in the system maintains a constant-sized set of neighbors (32 in our implementation). Our primary requirement is that these neighbors are diverse in terms of both latency and geography, so as to increase the likelihood that peers can route queries towards a given target region (and away from the corresponding forbidden region).

To maintain as diverse a set of neighbors as possible, each peer p maintains two sets of peers: (1) a set of m *known-active* peers, whom p has heard from recently, and (2) a set of n *neighbors*, which p uses when processing queries, The known-active set is larger (4× in our implementation), and is used for populating the neighbor set as follows:

Latency diversity. Peers regularly obtain round-trip time (RTT) measurements to peers in their known-active set: they actively ping peers when they first meet (and periodically thereafter), and record the RTTs from routing messages. When a peer obtains a new RTT measurement, it updates its known-active set of peers, and decides whether or not its neighbor set should be updated. To maintain a diverse set of neighbors, the invariant we would like to maintain is that, at any point in time, a peer's relative differences in latencies to its neighbors are maximized. We approximate this with the following simple heuristic:

Periodically, each peer p determines his most redundant neighbor in terms of RTT. Suppose r_i represents neighbor i's RTT to p, and that $r_i \leq r_{i+1}$ for all i. Neighbor i's "redundancy" is captured by the inverse of its relative difference to its neighboring values: $\frac{r_i}{r_{i+1} - r_{i-1}}$ (for notational convenience, let $r_{-1} = -\infty$ and $r_{n+1} = \infty$). Peer p then removes the neighbor i with the greatest redundancy, and adds a random peer from its known-active set[4].

Geographic diversity. A geographically diverse set of neighbors is also important in Alibi Routing; it increases the likelihood that each relay has a neighbor outside of a given forbidden region. In Alibi Routing, when peers exchange entries from their known-active set with one another, they share not only a list of peers they know, but also those peers' GPS coordinates (including their own). Sharing geographic information is important for processing queries (§4.4), so we leverage them for achieving diverse neighbors, as well.

To achieve geographic diversity, we apply a similar heuristic as with latencies: Peer p computes the bearing θ_i between its GPS coordinate and q_i's, that is, $\theta_i = \tan^{-1}\left(\frac{p.\mathsf{lat} - q_i.\mathsf{lat}}{p.\mathsf{lon} - q_i.\mathsf{lon}}\right)$. p sorts these bearings (w.l.o.g., suppose $\theta_1 \leq \cdots \leq \theta_M$), and removes the "most redundant" bearing from the list. θ_i is considered the most redundant if it has the smallest average difference with its predecessor θ_{i-1} and successor θ_{i+1}.

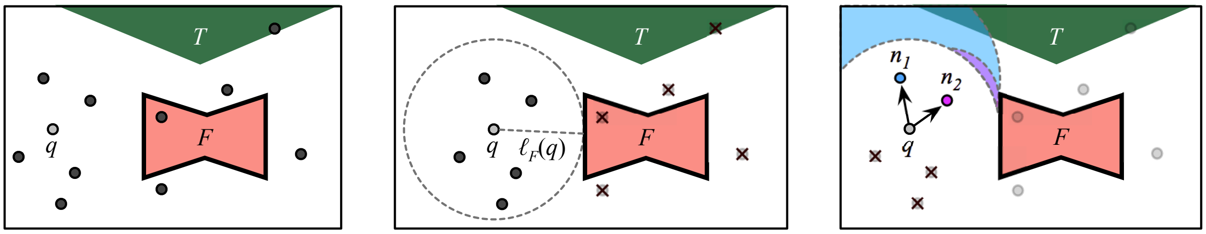
These measures of diversity guide an Alibi Routing peer's decisions as to what neighbors to add or drop. Adding new neighbors is done as follows:

Joining. To join, n first contacts a peer p it knows, and obtains p's known-active set (this contains but is not limited to p's neighbor set). n then pings these nodes with a random nonce, asks them for their GPS coordinates, adds them to his known-active set, and uses them to construct his own neighbor set, as described above. Note that the neighbor set of a new node is likely to be different from the node that bootstrapped it (unless they are extremely close to one another). This process is trivial to bootstrap—any peer can initiate its own instance of a Alibi Routing overlay—and permits incremental deployment.

Establishing neighbors. When a peer p decides to add peer q as a neighbor, p first pings q with a random nonce, and records the RTT. All pings in Alibi Routing have such an unpredictable nonce; without it, q could under-report his RTT by constructing and sending a response before receiving p's ping[5]. The peers then exchange their GPS coordinates—precise locations would be a violation of the users' privacy, but fortunately, as Figure 2 indicates, even relatively coarse-grained GPS coordinates (city- or often even country-level) often suffice. Finally, the peers establish a shared symmetric key, which they use to compute and verify MACs on the packets they forward for one another. This same process applies when establishing a connection between a source node s and an alibi peer a: the MACs provide the proof that a indeed forwarded the packet, as described in Section 3.

[4] We considered alternative schemes such as Meridian's expanding rings [44], but, in our setting, we found their difference to be statistically insignificant in evaluation.

[5] Such an attack, which seeks to under-represent one's RTT, is similar in spirit to the TCP OptAck attack [39].



(a) q's neighbors (b) Safety condition (c) Progress condition

Figure 3: Choosing a set of next-hop neighbors when processing a query, with target region T and avoidance region F.

Properties. Note that our neighbor maintenance protocol does *not* explicitly invoke user trust assumptions, nor does it make use of forbidden regions. Peers can lie about the data they share—they can arbitrarily inflate latencies and completely forge GPS coordinates. However, as we will see, Alibi Routing uses the fact that attackers cannot lie about having *lower* latencies in order to filter out false information from potentially forbidden peers. We demonstrate in Section 5 that these mechanisms ensure that Alibi Routing is safe regardless of any peer's neighbor set.

4.4 Query processing

When a peer obtains a query message, its task is to determine the next-hop neighbor who has the greatest chance of locating an alibi. Because of its adversarial setting, we can think of query processing as having to satisfy two conditions: safety and progress. First, the next-hop neighbor must be *safe*, i.e., the neighbor must not reside in a forbidden region nor can communication with the neighbor traverse a forbidden region. Second, each hop must make as much *progress* as possible towards a target region. We next describe how a peer checks both of these conditions.

4.4.1 Safety: Determining trustworthy neighbors

Suppose peer q is processing a query $\langle s, d, F, T\rangle$. How does q determine with certainty that one of its neighbors, n, is not in F?

The insight is that the latency between two nodes cannot be lower than the speed of light (c) would permit. So long as peer q knows its own latitude and longitude, it can compute $d = \mathsf{ShortestDistance}(q, F)$: the great-circle distance between itself and the closest point in F. It can then use this distance to estimate what the *minimum possible RTT* is between itself and any node in the forbidden region. We denote this minimum RTT by $\ell_F(q)$. Concretely, if q's lowest measured RTT to n is $L(q, n)$, then q can be certain that n is not in the forbidden region so long as

$$L(q, n) < \ell_F(q). \tag{3}$$

A peer that satisfies this condition is certainly not in the forbidden region. However, if peer n does not satisfy the condition, it does not necessarily mean that n is in F; only that q cannot determine with certainty that n is *not* in F.

Estimating ℓ_F: the minimum possible RTT. This approach depends heavily on being able to estimate $\ell_F(q)$, the minimum possible RTT between peer q and any possible host in forbidden region F. As information cannot travel faster than the speed of light, the safest approach is to let $\ell_F(q) = \frac{2}{c} \cdot \mathsf{ShortestDistance}(q, F)$ (the factor of two captures the fact that ℓ_F represents a round-trip time).

However, assuming that all information can travel this fast may be overly conservative—real latencies are typically not so close to the speed of light, so we may be able to identify more peers as being trustworthy if we can choose a larger (but still safe) value. Agarwal and Lorch [1] observed from latency measurements on over 3.5 million gaming consoles that there is an approximately linear relationship between RTTs and the great-circle distance between two hosts on Earth. The least-squares fit they propose represents approximately a five-fold increase over the speed of light. However, their fit overestimates latencies for many node pairs. Overestimating latencies could violate correctness in our setting: it would correspond to a peer believing that one of its neighbors is not in a forbidden region when it really is.

What, then, is a mapping from distance to RTT that is *safe* (i.e., close to $2/c$) but not overly *conservative* (i.e., not much closer to $2/c$ than what current links achieve)? There are many factors that influence how close Internet communication comes to the speed of light. Most fiber-optic links achieve approximately 2/3 the speed of light, while coaxial cables typically obtain between 66–82%, depending on the type of insulator [6]. Additionally, there are serialization delays, buffering at routers, and so on [42], but lacking a concrete understanding of how much "friction" such effects add to various regions of the Internet, we choose to err on the side of safety and assume these delays are zero.

We take the conservative approach of assuming fiber-optic links operating at $2/3$ the speed of light, and zero delays otherwise. Thus, we estimate the lowest possible RTT between q and F as follows:

$$\ell_F(q) := \frac{3}{c} \cdot \mathsf{ShortestDistance}(q, F) \tag{4}$$

We observe that all points in Agarwal and Lorch's data fall above this line. We have also verified this property with all data that we have collected in our experiments on PlanetLab.

A particularly useful property of Eqs. (3) and (4) is that they require only information that is readily accessible to the peer seeking to check the condition. Note that, to compute them, q needs to know: (1) its own geographic coordinates (we assume each user knows where he or she is, at least to some reasonably fine granularity), (2) the forbidden region (which is provided in the query), and (3) its observed RTT to n. Moreover, we argue in Section 5 that an adversarial neighbor n cannot meaningfully manipulate q's computation as to whether n meets the safety condition.

4.4.2 Progress: Efficiently finding alibis

An intermediate node q may have more than one neighbor who satisfies a query's safety condition; in the extreme, if the forbidden region were very small, all of q's neighbors might be considered safe. Forwarding to all safe neighbors would be inefficient (and could, in the extreme, lead to flooding). Instead, an intermediate Alibi Routing node returns the neighbors who are both safe and who make the most progress towards finding alibis.

Consider node q with safe neighbors $S = \{n_1, \ldots, n_{|S|}\}$. There are two broad cases to consider: (1) If q is not in a target region, it must get the query closer to one. (2) If q is in a target region, it should forward the query to a relay that reduces the latency inflation between source and destination.

Getting to a target region. If neither q nor its safe neighbors are in a target region, q must pick the neighbors who have the greatest chance of ultimately forwarding the query to someone who is. It may seem tempting to choose the safe neighbor who is the closest to some t_i, that is, the $n_i \in S$ who minimizes $\ell_T(n_i)$. This would ensure that the query makes progress at each hop, but it can also cause queries to fail. For example, neighbor n_2 in Figure 3(c) is closest to the target region, but because it is also close to the forbidden region, it is likely to have few safe neighbors to forward to.

The key insight is that neighbors with larger distances to the forbidden region will have larger values of ℓ_F, and therefore will have more neighbors who satisfy the safety condition. We balance the two goals—get closer to T while staying far from F—by having q choose neighbor n_i who minimizes $\ell_T(n_i) - \ell_F(n_i)$. In the example in Figure 3(c), the progress that q's neighbors make are represented by the shaded circles: n_i's circle is centered at n_i and has radius $\ell_F(n_i)$. A neighbor's progress is captured by how close this circle is to T. Note that n_1 maximizes progress, and is therefore the most likely to have a neighbor who is not only close to T, but possibly inside it. q would therefore choose n_1 as the next hop in this example.

4.5 Avoiding local minima

Alibi Routing is in some ways similar to prior "geographic routing" protocols, in that it uses a heuristic to measure progress towards a goal (we describe prior work in this area in Section 7). A classic problem facing greedy heuristics is that they are likely to run into *local minima*: when processing a query, peer $p \notin T$ may have the lowest value of $\ell_T - \ell_F$ out of all of its neighbors. What this means in Alibi Routing is that it may be necessary to sometimes forward queries *away* from a target region in order to find a peer who knows neighbors who can ultimately get the query to the target.

We make use of two key mechanisms to achieve this. First, each peer p maintains data about its two-hop neighbors, $\mathcal{N}^2(p)$. To do so, p periodically requests each of his one-hop neighbors $q \in \mathcal{N}^1(p)$ to send him $\mathcal{N}^1(q)$. When forwarding a query, p chooses the neighbor $q^\star \in \mathcal{N}^1(p)$ who has neighbor $n^\star$ who minimizes $\ell_T(n) - \ell_F(n)$. This helps avoid some local minima, but there may still be peers who appear to make better progress than any of their two-hop neighbors.

To this end, the second metric we make use of is *query forking*. Each peer randomly splits its neighbors into two disjoint sets, S_1 and S_2. When processing a query, the peer forwards it to peer $p_i \in S_i$ who minimizes the progress condition, $\ell_T - \ell_F$, for *both* $i = 1$ and 2. This has the possibility of introducing loops, and thus each peer maintains short-lived state of what queries they have seen; if they receive a query they have already seen, they simply drop it. Query forking also has the possibility of flooding a larger portion of the network than necessary, if left unconstrained. To address this, each query contains a TTL, and we use a simple *expanding ring* search: a source node initializes TTL to 2, and increments it until it succeeds, up to a maximum supported TTL (in our implementation, we have observed that a TTL above 7 does not yield enough marginal return to merit how many additional nodes it contacts).

Alibi Routing is unique with respect to many other overlay routing systems in that it bases its routing decisions only on a vague notion of where the destination might be (target regions can constitute a huge portion of the world). As we will demonstrate in Section 6, however, Alibi Routing is still able to achieve high success rates without contacting many nodes. Before we evaluate empirically, however, we analyze Alibi Routing's security properties.

5. SECURITY ANALYSIS

We analyze Alibi Routing's security with respect to attacks on its safety and progress, and discuss attacks that Alibi Routing does not seek to solve.

Attacks on safety. We begin our security analysis by considering attacks that seek to undetectably divert traffic through a forbidden region. First, we show that one cannot trick a trusted peer into thinking that an unsafe peer is safe:

PROPERTY 1. *If $q \in F$, then there does not exist a trustworthy peer p for which q satisfies the safety condition.*

Conceptually, this is true because q would have to fabricate a lower latency than is physically possible to appear convincingly safe to p. To see this, recall that $\ell_F(p)$ is the minimum possible latency between p and any point in F. Because $q \in F$, we have $L(p, q) \geq \ell_F(p)$, that is, there exists some $\Delta \geq 0$ such that $L(p, q) - \Delta = \ell_F(p)$. Suppose that q were indeed able to satisfy the safety condition; then $L(p, q) \ll \ell_F(p)$, that is, there would exist some $\delta > 0$ such that $L(p, q) + \delta = \ell_F(p)$. These together would imply that $\delta = -\Delta$, and thus that $\Delta < 0$, a contradiction.

We next show that the Alibi Routing protocol is not susceptible to packet manipulation by nodes within a forbidden region. Moreover, this security property ensures that any packet from an attacker in a forbidden region will be ignored altogether.

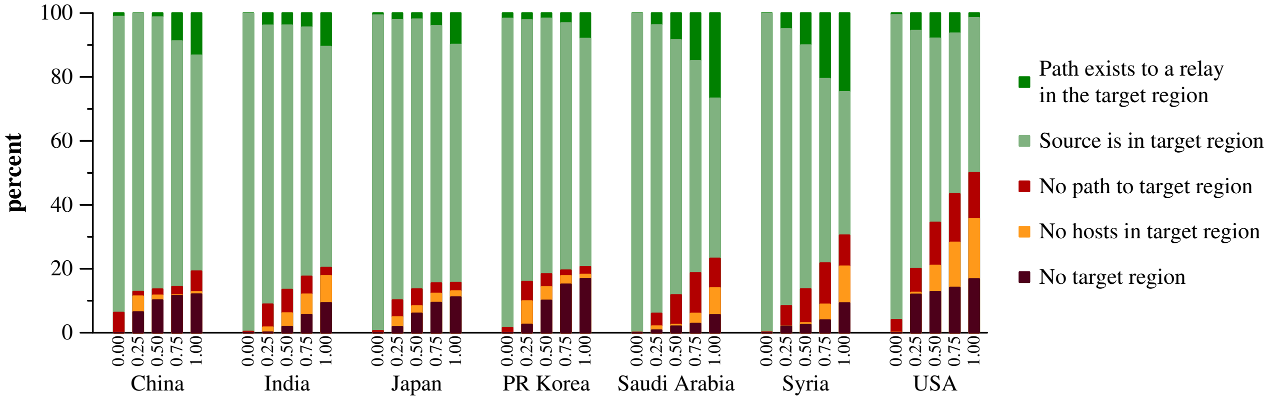


Figure 4: Feasibility of Alibi routing for different forbidden regions. (Simulated deployment on PlanetLab.)

PROPERTY 2. *All trustworthy peers ignore all packets that traverse F.*

To see this, suppose peer p is trustworthy. Suppose further that p forwarded a packet through q, but that in the process of doing so, the packet traversed F and yet p did not ignore it. Because a trustworthy peer p ignores any packet that violates Eq. (1), the supposition that p did not ignore it would mean that $R(p,q) \ll \min_{f \in F}(R(p,f) + R(f,q))$. The only way that this inequality could hold for a packet that traversed F is if q and F are colluding, and thus that $q \in F$. However, Property 1 says that, if p is trustworthy and $q \in F$, then p would not believe q to be safe, and therefore would not have forwarded a packet through q in the first place. We thus have our contradiction: a packet that traverses F must have been ignored by any trustworthy peer.

Note that establishing Properties 1 and 2 required no assumptions of the peers' neighbor sets. This leads to an interesting corollary that Alibi Routing is safe regardless of any peer's neighbor set.

Attacks on progress. An adversary could launch an eclipse attack [41] by attempting to populate a victim's neighbor set with all attackers. Note that such an attack would require an attacker to be very close to the victim. Although, as we have shown, the attacker cannot violate the victim's safety, it may be able to impact progress. Recall that next-hop peers are ranked by their progress condition: the neighbor n who minimizes $\ell_T(n) - \ell_F(n)$ makes the most progress. Peers who are most susceptible to an eclipse attack are those who are closest to $F(n)$ and thus have small values of $\ell_F(n)$ Thus, the more likely a peer n is susceptible to an eclipse attack by attackers in F, the greater the value of $\ell_T(n) - \ell_F(n)$ in general, and thus the less likely n will be chosen as a next hop in a query. Constraining progress therefore requires proximity to otherwise viable relays.

Non-attacks.

We close this section by describing some attacks on users that we do not believe need to be solved by a provable route avoidance system, as they can be solved by combining Alibi Routing with a more traditional system.

Laundering attack traffic. In any overlay routing system, relays could be used for reflecting attack traffic: s could send attack traffic to d via a relay r to make it appear that r is the one attacking. We do not believe there is any fundamental difference between such an attack in Alibi Routing and other systems, and so traditional approaches apply (e.g., white-listing sources or destinations, as in Tor [10], and rate-limiting how much a peer contributes to the system, as in BitTorrent [9]).

Sending copies of data to attackers. Any host or router on the path from s through an alibi to d could send copies of packets to the forbidden region. This does not violate Alibi Routing's goals: to establish an unadulterated path of communication between s and d. Keeping communication private is, of course, an important issue: to this end, s and d ought to employ end-to-end encryption. Further, if s and d desire sender and/or receiver anonymity, they should apply anonymous systems such as Tor [10] or $\mathcal{P}5$ [40]. Alibi Routing can be composed with such systems to provide defense in depth, for instance by using the alibi condition when constructing Tor circuits (§7).

6. EVALUATION

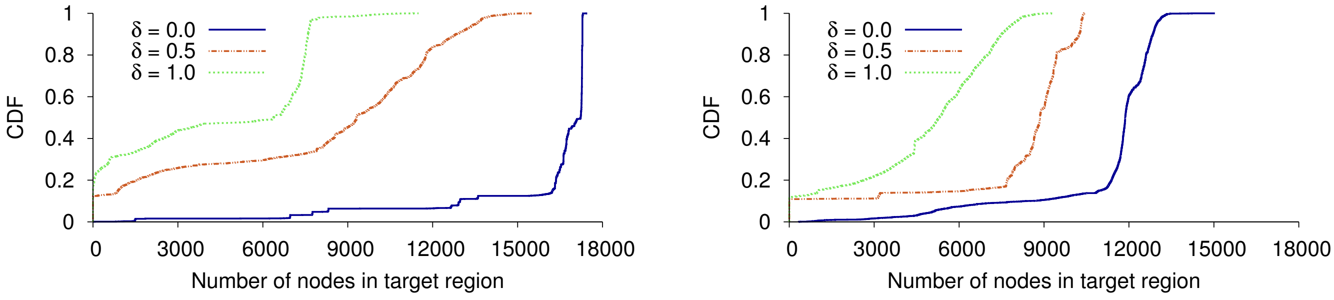
We present an evaluation of Alibi Routing using both simulations and an implementation deployed on PlanetLab. Our data and code are publicly available.

6.1 Who can be avoided?

The first question we seek to evaluate is: for what source, destination, forbidden region triples could alibis exist?

To answer this, we compute target regions using the method described in Section 4.2.2 with both data collected from PlanetLab and using a simulated deployment of 20,000 nodes. For forbidden regions, we used several countries identified in the 2012 Internet Enemies Report [34]—China, Syria, North Korea, and Saudi Arabia—as well as the three other countries with the most number of Internet users as of the time of this writing—USA, India, and Japan.

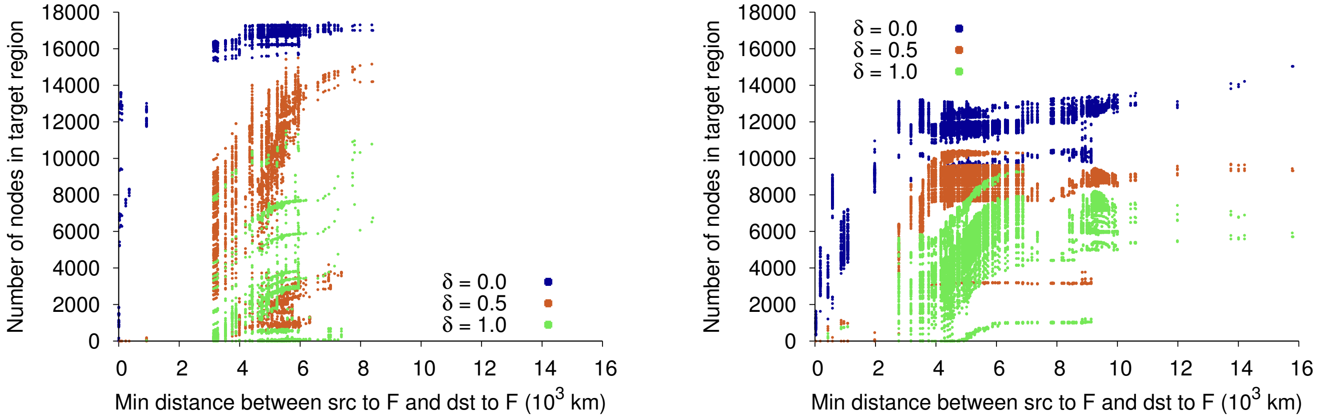
Who can avoid whom? Using latency data we gathered from PlanetLab, we simulated Alibi Routing with different



(a) USA is the forbidden region.

(b) China is the forbidden region.

Figure 5: CDF of the number of simulated nodes in the target region. (Simulated deployment of 20,000 nodes.)



(a) USA is the forbidden region.

(b) China is the forbidden region.

Figure 6: Effect of source/destination distance on the number of nodes in the target region. The x-axis is the minimum of the distance between (source, destination) to the forbidden region. (Simulated deployment of 20,000 nodes.)

inequality values (δ from Eq. (1)). Figure 4 shows the fraction of source-destination pairs of PlanetLab nodes for which (from bottom to top):

(1) *There is no target region whatsoever.* This happens infrequently, and with greater values of δ; it reflects the instances in which the source and/or destinations are simply too close to the forbidden region to obtain proof (we do not include in our evaluation instances where s or d are *inside* F).

(2) *There are no hosts within the target region.* This would improve with a more geographically diverse deployment, yet even with our 425-node PlanetLab dataset, we find these numbers to be encouraging.

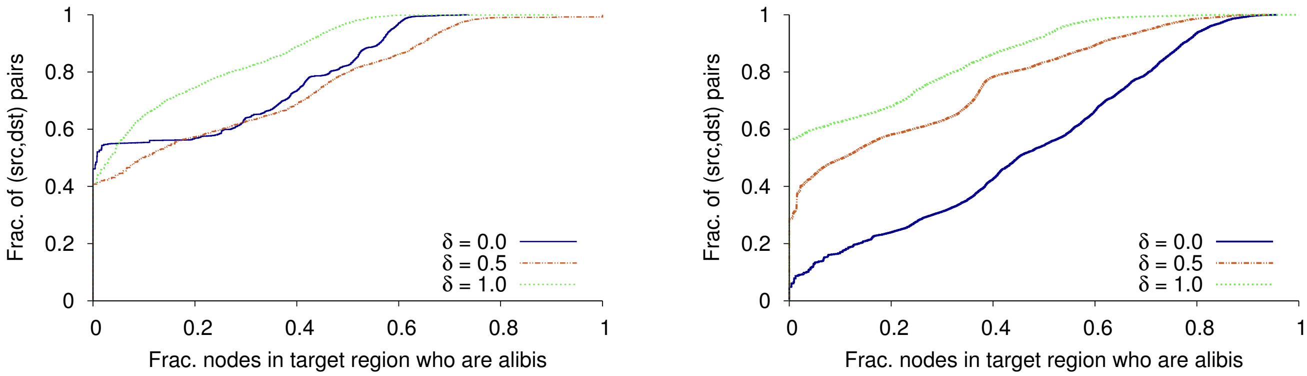
(3) *There is no safe path to the target region.* This, too would improve with a more geographically diverse deployment.

(4) *The source-destination pair does not need a relay* to be able to communicate while provably avoiding the forbidden region (i.e., the source is in the target region). This is rather common in our dataset, and corresponds to instances wherein both source and destination are far from the forbidden region (for example, two hosts on the same subnet obtaining sub-millisecond latencies would not need to use a third party relay to ensure they are avoiding someone thousands of miles away).

(5) *An alibi relay is necessary and Alibi Routing succeeds in finding one.* Finally, we see the fraction of hosts who need an alibi, and for whom Alibi Routing would be able to deliver one. This value generally increases with larger values of δ: when avoiding China, for instance, Alibi Routing would be able to locate alibis for roughly 80% of source-destination pairs when $\delta = 1.0$, but far fewer when δ is as low as 0.5. This is actually reflective of the fact that the system does not *need* alibis when δ is very low (and would have instead fallen into category (4) above). These results are also correlated with location: note that Saudi Arabia, and Syria show similar trends and are geographically proximal.

In the vast majority of cases, target regions are non-null, and in fact, Alibi Routing *can* be successful. Figure 4 only measures the case when a safe path to a node in the target region can be found: our later experiments measure the fraction of time such paths find relays that satisfy the alibi condition (Eq. (1)).

How well populated are target regions? The above results demonstrate how often an alibi is needed and, when so, how often a peer can find at least one. We next investigate *how many* alibis are available in a much larger (simulated) deployment of 20,000 nodes. We chose the location of these



(a) USA is the forbidden region.

(b) China is the forbidden region.

Figure 7: Fraction of possible alibis that pass the alibi condition. (Simulated deployment of 20,000 nodes.)

nodes by subsampling the MaxMind node list [26] which provides a representative sample of global Internet deployment. For each forbidden region, and for each pair of nodes in our data set, we computed the target region (if any), and counted the number of simulated nodes contained in the target region.

Figures 5(a) and 5(b) show the CDF of number of nodes that lie in the target regions with the USA and China as forbidden regions. Each plot contains data for three different inequality factors (δ). As expected, as δ increases, constraints on the target region are stronger, which leads to smaller target regions and fewer possible relays. Figure 5(b) shows that approximately 10% of pairs are not able to provably avoid China when the inequality factor is set to 1.0; this number rises to 22% for the USA.

Intuitively, it will be more difficult to find valid alibis when the source or destination node is close to the forbidden region. Figure 6 quantifies this. Each point in this figure represents a source-destination pair, with x-value equal to the minimum distance of either the source or the destination to the forbidden region. The horizontal gaps in these plots is due to oceans: in a global deployment, distances between nodes are not uniformly distributed. The y-value corresponds to the number of possible relay nodes available to the source-destination pair for the given forbidden region.

For both the USA and China set as forbidden, Figure 6 shows the expected strong correlation between the minimum distance and the number of possible relays. In most cases, when relays cannot be found, either the source or destination is close to the forbidden region. Similarly, as the inequality factor increases, the target regions are constrained, resulting in fewer possible relays: the vertical striations in the graph captures this phenomenon.

6.2 How predictive are target regions?

Recall that target regions indicate the geographic areas wherein alibi nodes *may* exist; that is, no peers outside a target region could possibly be an alibi, but not all peers within a target region are guaranteed to have low enough latencies to be a viable alibi. We next evaluate how predictive target regions are by measuring the likelihood that a given peer within a target region can forward packets quickly enough to satisfy the alibi conditions (Eq. (1)). To assess what fraction of nodes in the target region are viable alibis, we conducted the following experiment: we periodically, once every 10 minutes, computed all-pairs pings between each pair of nodes on PlanetLab for 24 hours. During each run, each host sent five pings to all other hosts, and we recorded this data. Next, for different forbidden regions, we computed target regions, and classified whether a PlanetLab host was a possible alibi, i.e., whether or not it was in the target region.

Figure 7(a) plots, over all (source, destination, possible alibi) triples, the cumulative distribution of the fraction of times a possible alibi passes the alibi condition, i.e., the possible alibi is a viable relay. The plot shows that about half of the time, PlanetLab nodes in the target region are not able to pass the alibi condition. This is explained partly by the routing centrality of the US [19]: disproportionately many routes pass through the US, thus even if a peer is in the target region, there may not be a safe path to it. Further, delays within PlanetLab and queuing delays on the Internet ensure that only relays that are very favorably placed can pass the alibi condition. The picture is different when China is considered the forbidden region (Figure 7(b)). For low inequality values, only in about 5% of the cases can a valid alibi not be found. We acknowledge that this real-world measurement result is biased by the placement of PlanetLab nodes.

Over one day of pings, the RTTs we measured did not change enough to cause nodes to oscillate between being valid alibis and not. If a node was ever a valid alibi, it remained so (with high probability) for all our measurements. The same is true for nodes that were never a valid alibi.

We next turn to evaluating how close to these ideals our specific Alibi Routing protocol performs.

6.3 Alibi Routing success and performance

We next measure how successful Alibi Routing is at finding alibis "in the wild" by running our implementation on 370 PlanetLab hosts, and through a simulation over tens of thousands of nodes. Our findings indicate that Alibi Routing succeeds the vast majority of the time, and moreover, finds alibis quickly.

δ	Number of nodes 10,000	 20,000
0	99.5	100.0
0.5	84.12	93.60
1.0	84.12	93.28

Table 1: Protocol success rate (simulation).

δ	Forbidden Region USA	 China
0	100.0	97.19
0.5	99.56	100.0
1.0	100.0	97.30

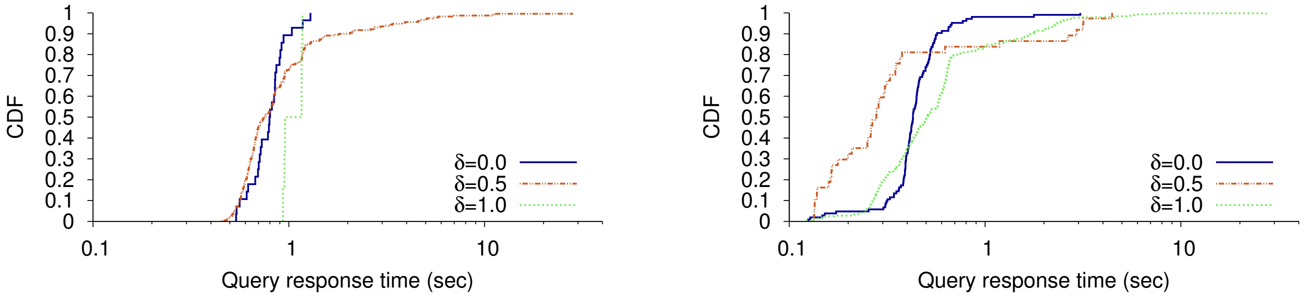
Table 2: Protocol success rate (PlanetLab).

δ	Number of nodes 10,000	 20,000
0.0	7.11	4.68
0.5	44.40	37.14
1.0	38.76	35.58

Table 3: Number of nodes contacted (simulation).

δ	Forbidden Region USA	 China
0	1.03	1.00
0.5	1.00	1.30
1.0	1.66	1.00

Table 4: Number of nodes contacted (PlanetLab).



(a) USA is the forbidden region.

(b) China is the forbidden region.

Figure 8: Time taken to find a relay for all PlanetLab source-destination pairs. (Implementation on PlanetLab.)

Table 1 shows Alibi Routing's success rate for a simulated dataset with 10,000 and 20,000 nodes subsampled from the MaxMind node set [26]. In these results, we capped the maximum TTL to 7, and with this setting, for larger values of δ, the protocol is successful about 84% of the time for the 10,000 node deployment. Running the protocol with much larger TTLs would increase the success rate, as it would increase the chances of finding a path to a target region. However, larger TTLs impose an exponentially higher cost (in terms of messages and nodes contacted) for these queries. For our PlanetLab deployment (Table 2), our implementation (also with max TTL 7) has near 100% success rate regardless of the inequality factor.

Tables 3 and 4 show the protocol overhead in terms of average number of nodes contacted, both for simulations and the PlanetLab deployment. The average overhead is extremely low: on average, on PlanetLab, most searches terminate in two hops. In our simulations, even with 20,000 nodes, the average search cost is less than 40 nodes. The peak search cost is incurred in the 10,000 node case with inequality factor set to 0.5. This, too, is because a very low δ makes it easy to find eligible relays, and a large δ constrains the target region such that the search cannot proceed very far before all eligible nodes are exhausted.

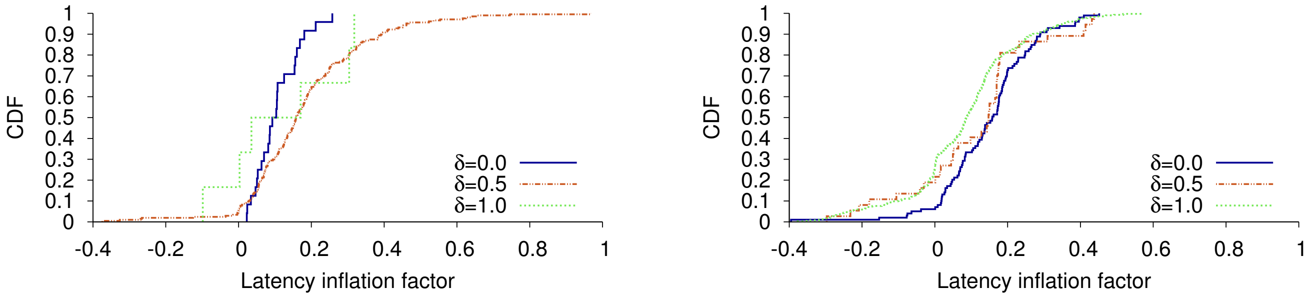
Figures 8(a) and 8(b) show the time taken to find relays, as measured by the `gettimeofday()` call at the source for the cases when a relay is found. The plots show data for all source-destination pairs when they are trying to avoid USA or China. There are relatively few feasible pairs with a non-null target region for $\delta = 1.0$ when trying to avoid the USA, and the corresponding CDF does not have many data points. Note that the plots do *not* include the cases when a relay cannot be found: in these cases, our code waits for a maximum of 40 seconds before timing out on ongoing searches. The plots show that for successful queries, our implementation finds relays relatively quickly, the vast majority being found in less than one second. Interestingly, the time it takes to find alibis for $\delta = 0.5$ is usually higher than the time taken for $\delta = 1.0$. This is because the target regions are smaller for $\delta = 1.0$, causing more queries to fail (which is not captured in this plot but is evident in Tables 1 and 2).

Figures 9(a) and 9(b) show how Alibi Routing affects end-to-end latency. Again, the plots show the data for all pairs when they are trying to avoid either the USA or China. For many pairs, Alibi Routing *improves* latency [36, 25]. But for the vast majority, it increases latency by less than 50%. This is a surprisingly positive result, given both the geographic area covered by the USA and China, and their routing centrality. Finally, we note that when relays can be found, latency inflation is relatively insular to the inequality factor.

7. RELATED WORK

Alibi Routing is broadly related to a wide range of work towards influencing what paths users' packets take, inferring what actions were taken within a distributed system, and hiding packet contents from untrusted third parties. We discuss related work here, and observe that Alibi Routing constitutes a unique set of goals that are largely orthogonal and complementary to prior systems.

Avoidance without proof. Recently, there has been a wide array of research into systems that avoid parts of the network via explicit support from in-network routers [21, 23, 24, 49]. For example, LIFEGUARD [21] identifies routing failures and routes around them by sending BGP messages that "poison" a failure-prone area. Also, Kline and Reiher [23] propose a scheme for avoidance routing that involves explicit participation from BGP routers; the idea is to issue queries for destinations that also include requests for certain security properties, such as an AS's location.



(a) USA is the forbidden region.

(b) China is the forbidden region.

Figure 9: Latency inflation for using Alibi Routing over the direct path. (Implementation on PlanetLab.)

However, *none of these prior systems offer* ***proof*** *of avoidance*, and are designed to operate in non-adversarial settings. Each of these prior systems trusts, to some extent, the ASes in the network to follow the protocol; for instance, to avoid an AS, LIFEGUARD relies on that AS to announce poor routes, which a routing-capable adversary [37] could easily avoid. What distinguishes our approach from these is the goal of *proving* that an area of the network was avoided, without explicit involvement of those whom users seek to avoid. Additionally, because Alibi Routing is a peer-to-peer system, it can be deployed without broad adoption by ISPs.

Policy routing. Many prior systems allow hosts to express some degree of path preference [14, 35, 47, 48], including the original IP RFC [31]. More recent systems have introduced the ability of enforcing compliance to users' desired routing policies [28, 22]; these systems in particular obtain proof that a packet traversed a sequence of locations in the network, even in an adversarial setting. However, none of these systems offer proofs of avoidance; they can ensure that a packet visits an ordered set of entities in the network, but do not detect the intermediate hops: what Kim et al. refer to as the path detour attack [22]. In this sense, Alibi Routing is orthogonal and potentially complementary; we use a much more rudimentary form of path enforcement (the relay simply signs the packet)—policy enforcement mechanisms could improve this, particularly if Alibi Routing were to be expanded to support multi-relay paths.

Obfuscating packet contents. Anonymity and censorship-resistant systems often hide packets' contents via obfuscating proxies [8, 10, 40] or steganography [11]. Whereas Alibi Routing seeks to avoid censors altogether, these approaches seek an orthogonal set of properties: to minimize what a censor can learn when it sees a user's traffic.

Several recent approaches broadly referred to as *decoy routing* [18, 46, 17], attempt to make it appear to a censor that the user is communicating with some destination d whom the censor allows. In reality, a "decoy router" on the path from the user to d intercepts these requests, and serves as a proxy to the user's true destination. As these systems also "use and confuse" forbidden regions, they too seek goals orthogonal to Alibi Routing's.

A natural question to ask is: why do we need Alibi Routing if we have systems that preserve anonymity and encrypt end-to-end? In practice, *no one approach works all the time*: anonymity systems must typically make their proxies well-known, and thus a resource-rich adversary may be able to identify and block access to them; decoy routing systems make strong trust assumptions that are not compatible with routing-capable adversaries [37]; and of course Alibi Routing itself cannot make any avoidance guarantees for users within a censored regime.

We believe, however, that Alibi Routing complements the others well. For example, consider Tor [10], in which a source chooses a set of relay nodes and constructs a virtual circuit through them. These relays are typically chosen at random, but doing so can result in two consecutive relays whose traffic transits a censoring regime. Alibi Routing can be applied to Tor's relay selection to compute which circuits provably avoid known censors. Alibi Routing also composes well with decoy routing: one could for instance use decoy routing to get out of a censored regime, and Alibi Routing to avoid others further down the path. These are but brief design sketches, but they demonstrate Alibi Routing's potential for composing with prior work.

Influencing underlay paths with overlay routes. Overlay routing protocols, such as RON [3] and SOSR [15], demonstrate that overlay paths can be used to influence and improve upon the underlay paths that packets take. A natural approach would be to use overlay routing to find a path that avoids a censor—much like how SOSR uses one-hop relays to avoid slow parts of the network. This is in essence the approach we take, but the primary challenges are in *proving* when a relay avoids a part of the network, and *finding* relays who can achieve such proof. To meet this goal, our overlay routing techniques exploit the connection between geographic distance and latencies.

Other systems incorporate latency measurements into overlay routing [25, 36, 44]. Meridian [44] is an unstructured overlay system that iteratively uses latency measurements to perform a set of queries, such as nearest-neighbor search and leader election. It may be possible to achieve some form of avoidance routing with Meridian, for instance as a sort of "nearest-but-not-too-near" neighbor search. However, Meridian is not designed to operate in an adversarial setting, and, unfortunately, it is straightforward for an attacker to appear to be "not too near" by fabricating higher

RTTs. We believe that the mechanisms we present in this paper can be generally applied in securing these systems.

Geographic routing. Geographic routing protocols incorporate the location of nodes in their routing decisions [20, 32, 23], as does Alibi Routing. The typical approach to geographic routing is to apply a greedy heuristic which attempts to move as close to the target as possible at each hop. This is similar to Alibi Routing's use of target regions, but we are not aware of any geographic routing protocol that achieves avoidance in an adversarial setting. In such a setting, applying a greedy heuristic can, in the worst case, traverse the forbidden region; even when the heuristic is safe, it may end up forwarding it to a node who is just outside the border of the forbidden country. Alibi Routing shows that progress and safety must be balanced to achieve high query success rate; we believe these lessons can apply to other geographic routing protocols, as well.

Accountability and provenance. A tempting way to prove what routers *did not* do is to exhaustively prove what actions they *did* take. There is a wealth of prior work on holding participants in a network accountable for the actions they *did* take [2, 16, 50]. For instance, PeerReview [16] assigns "witnesses" to each participant, which monitor all incoming and outgoing messages, and emulate the protocol to ensure that the participant behaved correctly. More recent findings apply counterfactual reasoning to observations about the network to achieve "negative provenance:" attributing a set of (in)actions led to the absence of an expected event [45].

Unfortunately, it is not clear how to apply these approaches to solve the problem of provable route avoidance; it would appear to require witnesses to be able to verify latency measurements (which does not seem possible in general settings, but would be an interesting area of future work).

Alibi Routing's light-weight proof structure can be performed using only local observations. We believe our "proof by alibi" can be combined with both positive and negative provenance systems to yield a broader set of inferences.

8. CONCLUSION

This paper introduces a primitive, provable avoidance routing, that, when given a destination and region to avoid, provides "proof" after the fact that a packet and its response did not traverse the forbidden region. We rely on the insight that a packet could provide an "alibi"—a place and time where it was—to prove that it must have avoided the forbidden region in transit from source to destination.

To demonstrate the feasibility of implementing this primitive, we have developed and evaluated an overlay routing protocol, Alibi Routing. Alibi Routing assumes that nodes outside the forbidden region are trustworthy in reporting their geographic locations and in vouching for neighbors that are too nearby to be in the forbidden region. It leverages this assumption to direct relay discovery queries toward a target region in which alibis might reside.

Our empirical results show that Alibi Routing is effective at finding alibis for a range of forbidden countries. However, Alibi Routing is not a panacea; primarily, it is unable to assist hosts who are very close to (or, of course, inside of) the regions they seek to avoid. But because its properties are largely orthogonal to prior work, we believe that Alibi Routing will compose well with them to strengthen their security guarantees. Moreover, we believe the techniques we have introduced can be applied to myriad other domains.

There are several possible extensions to Alibi Routing. Allowing routes through more than one relay could potentially improve Alibi Routing's success rates; discovering multi-relay paths and generating proof that they collectively avoid a forbidden region is an interesting area of future work.

Our implementation and data are publicly available at:

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

Acknowledgments

We thank the anonymous reviewers and our shepherd, Xiaowei Yang, for their helpful comments. We also thank Brendan Rowan for his help in developing an earlier version of the Alibi Routing protocol, and Dave Mount for his helpful guidance on computing target regions. Finally, we thank Brandi Adams for coming up with the name "Alibi Routing." This research was supported in part by NSF grant IIS-0964541, ONR grant N00173-13-1-G001, and an Amazon Web Services in Education grant.

9. REFERENCES

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

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

[3] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *ACM Symposium on Operating Systems Principles (SOSP)*, 2001.

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

[5] Anonymous. Towards a comprehensive picture of the Great Firewall's DNS censorship. *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.

[6] ARRL. *The ARRL Handbook for Radio Communications*. The ARRL, 89th edition, 2012.

[7] S. Aryan, H. Aryan, and J. A. Halderman. Internet censorship in Iran: A first look. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2013.

[8] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66, July 2000.

[9] B. Cohen. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, 2003.

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

[11] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing censorship and surveillance. In *USENIX Security*, 2002.

[12] R. Feynman, R. Leighton, and M. Sands. *The Feynman Lectures on Physics Vol. 1.* (Chapter 26). Addison-Wesley, 1963.

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

[14] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica. Pathlet Routing. In *ACM SIGCOMM*, 2009.

[15] K. Gummadi, H. Madhyastha, S. D. Gribble, H. M. Levy, and D. J. Wetherall. Improving the reliability of Internet paths with one-hop source routing. In *Symposium on Operating Systems Design and Implementation (OSDI)*, 2004.

[16] A. Haeberlen, P. Kuznetsov, and P. Druschel. PeerReview: Practical accountability for distributed systems. In *ACM Symposium on Operating Systems Principles (SOSP)*, 2007.

[17] A. Houmansadr, G. T. K. Nguyen, M. Caesar, and N. Borisov. Cirripede: Circumvention infrastructure using router redirection with plausible deniability. In *ACM Conference on Computer and Communications Security (CCS)*, 2011.

[18] 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.

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

[20] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[21] E. Katz-Bassett, C. Scott, D. R. Choffnes, Ítalo Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. LIFEGUARD: Practical repair of persistent route failures. In *ACM SIGCOMM*, 2012.

[22] T. H.-J. Kim, C. Basescu, L. Jia, S. B. Lee, Y.-C. Hu, and A. Perrig. Lightweight Source Authentication and Path Validation. In *ACM SIGCOMM*, 2014.

[23] E. Kline and P. Reiher. Securing data through avoidance routing. In *New Security Paradigms Workshop (NSPW)*, 2009.

[24] D. Levin, A. Bender, C. Lumezanu, N. Spring, and B. Bhattacharjee. Boycotting and extorting nodes in an internetwork. In *Joint Workshop on the Economics of Networked Systems and Incentive-Based Computing (NetEcon+IBC)*, 2007.

[25] C. Lumezanu, R. Baden, D. Levin, B. Bhattacharjee, and N. Spring. Symbiotic relationships in Internet routing overlays. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[26] MaxMind, Inc. `http://dev.maxmind.com/geoip/legacy/geolite/`.

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

[28] 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.

[29] J. A. Obar and A. Clement. Internet surveillance and boomerang routing: A call for Canadian network sovereignty. Online: `http://ssrn.com/abstract=2311792`, 2013.

[30] V. Paxson. End-to-End Routing Behavior in the Internet. In *ACM SIGCOMM*, 1996.

[31] J. Postel. Internet Protocol. IETF RFC-791, Sept. 1981.

[32] J. Preethi and R. Sumathi. An energy efficient on-demand routing by avoiding voids in wireless sensor network. In *Internation Conference on Information Systems Design and Intelligent Applications (INDIA)*, 2012.

[33] C. Reis, S. D. Gribble, T. Kohno, and N. C. Weaver. Detecting in-flight page changes with web tripwires. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[34] Reporters Without Borders. Enemies of the internet 2013, report. `http://surveillance.rsf.org/en/wp-content/uploads/sites/2/2013/03/enemies-of-the-internet_2013.pdf`, Mar. 2013.

[35] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching. IETF RFC-3031, Jan. 2001.

[36] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: Informed Internet routing and transport. *IEEE Micro*, 19(1):50–59, 1999.

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

[38] J. Seibert, S. Becker, C. Nita-Rotaru, and R. State. Securing virtual coordinates by enforcing physical laws. In *International Conference on Distributed Computing Systems (ICDCS)*, 2012.

[39] R. Sherwood, B. Bhattacharjee, and R. Braud. Misbehaving TCP receivers can cause Internet-wide congestion collapse. In *ACM Conference on Computer and Communications Security (CCS)*, 2005.

[40] R. Sherwood, B. Bhattacharjee, and A. Srinivasan. $\mathcal{P}5$: A protocol for scalable anonymous communication. *Journal of Computer Security*, 13(6):839–876, 2005.

[41] A. Singh, M. Castro, P. Druschel, and A. Rowstron. Defending against eclipse attacks on overlay networks. In *Proc. of ACM SIGOPS European Workshop*, 2004.

[42] G. Varghese. *Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices*. Morgan Kaufmann, 2004.

[43] G. Wang and T. E. Ng. Distributed algorithms for stable and secure network coordinates. In *ACM Internet Measurement Conference (IMC)*, 2008.

[44] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A lightweight network location service without virtual coordinates. In *ACM SIGCOMM*, 2005.

[45] Y. Wu, M. Zhao, A. Haeberlen, W. Zhou, and B. T. Loo. Diagnosing Missing Events in Distributed Systems with Negative Provenance. In *ACM SIGCOMM*, 2014.

[46] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the network infrastructure. In *IFIP International Information Security and Privacy Conference (SEC)*, 2011.

[47] X. Yang, D. Clark, and A. W. Berger. NIRA: A New Inter-Domain Routing Architecture. *IEEE/ACM Transactions on Networking (ToN)*, 15(4):775–788, 2007.

[48] X. Yang and D. Wetherall. Source Selectable Path Diversity via Routing Deflections. In *ACM SIGCOMM*, 2006.

[49] X. Zhang, H.-C. Hsaio, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. SCION: Scalability, Control and Isolation On Next-Generation Networks. In *IEEE Symposium on Security and Privacy*, 2011.

[50] M. Zhao, W. Zhou, A. J. T. Gurney, A. Haeberlen, M. Sherr, and B. T. Loo. Private and verifiable interdomain routing decisions. In *ACM SIGCOMM*, 2012.