Proving the Impossible: Provable Route Avoidance using Alibi Routing

Victoria Lai, Dave Levin
University of Maryland
CMSC499A

May 18, 2013

Abstract

We introduce route avoidance using alibi routing, in which a source can provably avoid a particular area when sending traffic to a destination. Our approach combines overlay routing and latency estimation using network coordinates to establish an "alibi" — a packet could not possibly have gone to the relay(s), the area to be avoided, and the destination within the considered timeframe. In our simulation over PlanetLab ping trace data, we show that geographic coordinates map well to 2-D network coordinates and allow us to reason about who can avoid whom and at what costs. We find that geography, particularly the continent and distribution of nodes, is a primary determinant of the possibility and costs of provable avoidance. We also discuss potential extensions to alibi routing, including secure network coordinates and "scaffolding" for constructing paths with multiple relays.

Introduction

In this work, we study a new networking primitive that we refer to as *route avoidance*: Send to a destination while *provably* avoiding certain parts of the network. Route avoidance has a broad set of applications, such as avoiding regions that censor traffic they transit [3] or performing what-if analyses of network failures (forward as if all of, say, Texas had experienced major power outages). There are two key properties we seek to obtain: easy deployment and provable avoidance. The purpose of this study is to understand if they can be achieved in today's Internet.

Ideally, route avoidance would be a low-level network primitive, but to facilitate deployment and adoption, we will investigate whether route avoidance can be provided using *overlay routing*. Overlay routing is a class of peer-to-peer protocols in which nodes form a virtual network on top of the physical one and influence the physical route by specifying intermediate destinations [2]. To select the intermediate destinations, we need to know or estimate the relative placement of nodes and costs of routing to them. Network coordinate systems such as Vivaldi [7] assign coordinates to each node such that the distance between two nodes' coordinates estimates their communication latency.

Previous avoidance routing work by Kline and Reiher implements avoidance at the autonomous system (AS) level and requires participation from ASes, but we would like the system to be decentralized and easy to deploy. Moreover, they do not provide proof that the avoidance was successful. To prove that the route successfully avoided an area, we can provide an "alibi" that the packet had to have gone another way. Network coordinates estimating latencies combined with overlay routing allow us to prove that a packet's route could not have gone through a particular area.

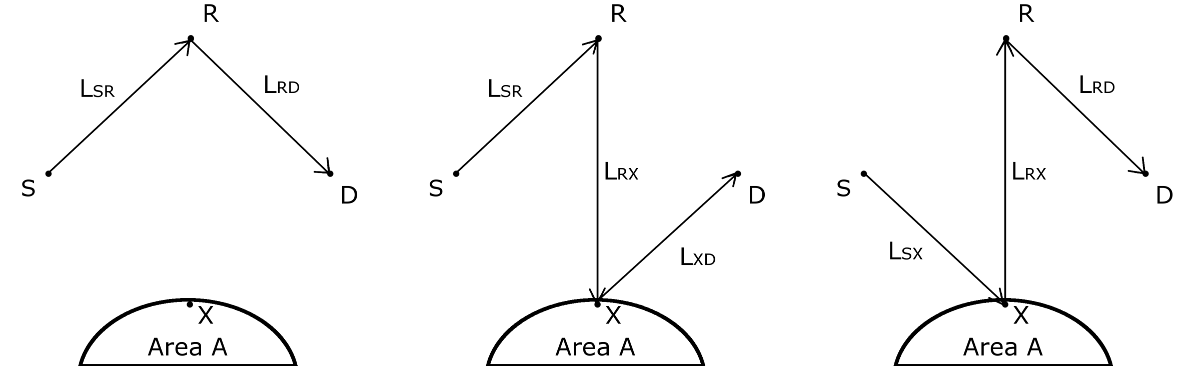
The organization of this paper is as follows: We begin with an overview of the system, introducing a scenario using one relay and discussing security considerations. Next we describe the data and simulation methods. We then present results and analysis from our simulation, including the mapping from geographic to network coordinates, comparison of countries' ability to reach destinations while avoiding a particular country, the costs of avoidance, and scenarios of high uncertainty. Finally we conclude with a discussion of future work and other considerations, such as relay selection, multiple relays and avoidees, and alternative coordinate systems.

System Overview

Alibi Routing

Suppose that a source S wishes to send a packet to destination D and be able to check that the packet did not travel through area A. Network coordinates [7, 9], along with a system that helps peers navigate through them [13], will allow S to select a relay R through which to forward packets. The question is does there exist a relay R such that forwarding through R can provide some proof that the packet avoided A?

Let $L_{P_1 P_2 \cdots P_n}$ denote the latency of the path through points P_1 through P_n (i.e. the sum of one-way latencies from P_1 to P_2, P_2 to P_3, $\cdots$, and P_{n-1} to P_n). As shown in Figure 1, we can compare latency estimates of the shortest path going through R but not A (L_{SRD}) to



(a) The packet does not go through X. (b) Route through X is detected using (1a). (c) Route through X is detected using (1b).

Figure 1: Avoidance scenario illustrating potential paths and latency estimates.

those of the shortest possible paths going through R and a point X in A (L_{SRXD}, L_{SXRD}).

Suppose S has sent a packet to D via R with round-trip time (RTT) r such that $\frac{r}{2} \approx L_{SRD}$.

$$\begin{aligned} L_{SRD} &\ll L_{SRXD} && \text{(1a)} \\ &\ll L_{SXRD} && \text{(1b)} \end{aligned}$$

If (1a) and (1b) hold, then we conclude that the packet could not have gone to X. The packet's alibi is that it went through R and reached D in a timeframe such that it could not have had time to go to X. Thus, a good relay is one such that L_{SRD} can serve as an alibi to determine that the packet did not go through A.

Security

We consider an attack model consisting of malicious nodes that may collude to provide honest nodes with false coordinates. The popular Vivaldi network coordinate system [7] is insecure to malicious nodes that advertise false coordinates or inflate RTT estimates to influence honest nodes' coordinates. A particularly insidious attack on network coordinate systems is the frog-boiling attack [6], in which malicious nodes falsely report their latency measurements in small increments; these go undetected because coordinate updates must allow for small network fluctuations. Newer latency estimation systems that protect against frog-boiling and other attacks include Treeple [4], KoNKS [5], and Newton [15]. Treeple forgoes the Euclidean coordinate system in favor of a graph-based approach to latency estimation, but it requires centralized, trusted vantage points. KoNKS and Newton remain fully decentralized and still use the Euclidean coordinate system, so they would be appropriate for our system.

We define a "good" relay as one that successfully leads to avoidance and a "bad" relay as one that leads to traversing the avoidance area A. Using the closest-node and basic-targeted attack variations of the frog-boiling attack [6], an attacker could make a bad relay appear good by obtaining a coordinate that is close to the source and destination and far from the avoidance area. With the network partition variation of frog-boiling [6], the attacker artificially partitions the network into two clusters of nodes. The attacker could put all bad relays and the destination in the same cluster as the source and the optimal relays and avoidance area in the other cluster, making bad relays appear good and vice versa. In these situations, the alibi routing approach would select the bad relays that make traversing the avoidance area more likely.

At a high level, an attacker can proceed in two ways — make all good relays appear bad so that avoidance seems impossible (I) or make bad relays appear good (II). In approach I, the source S believes that it, the destination D, or all possible relays R are in A, so there is no good relay that could help in avoiding A. If the source believes that avoidance is impossible, it will not use overlay routing to avoid the area and will not be able to determine whether its traffic went through A. In approach II, the source has a false location for S, D, R, or A that leads it to pick a bad relay. Traffic goes through A but the source thinks that it successfully avoided A. If S, D, or R are actually in A, the source thinks avoidance is possible when it is not. Otherwise if they are not in A, false coordinates can lead to selecting a relay that requires traversing A. In both approaches I and II, the attacker can either convince honest nodes of false coordinates for S, D, or R or for the avoidance area.

We proceed with simulation using the basic Vivaldi coordinate system, but a possible approach to making our system more secure is for the source and a set of peers (the verification set) to contact at least one node M in A. These probes provide RTT estimates, which may not be correct. Based on the results from the source and verification set, the source determines whether the initial network coordinates for M are reasonable (similar to [5, 16, 17]). The source can repeat this for multiple nodes in A, and only proceeds with the alibi routing protocol if its confidence in A's network coordinates are beyond a source-specific threshold T. It may also be possible for the source to use maximum likelihood estimation, the collected RTT estimates, and verifiers' coordinates to find a better approximation for M and thus A's network coordinates. Experimental investigation of the system under various attack models would help determine the optimal value of T, the size and selection method of the verification set, and the number of nodes and characteristics of M in A. Such a parameterization is an area of future work.

Data and Methodology

We used Harvard's Vivaldi simulator (http://www.eecs.harvard.edu/~syrah/nc/), which takes as input a set of latency measurements and outputs each node's network coordinates. For simulation, we needed a dataset with latency estimates between nodes and location information about the node (at the finest level, a latitude-longitude pair and at the coarsest, a country). We used median latencies from Harvard's 72-hour PlanetLab ping trace, which represents RTTs among 226 PlanetLab nodes distributed globally. The Vivaldi coordinates for Harvard's 4-hour subset of the trace and for the median latency set did not differ significantly, so we can use a single representative snapshot of the network without loss of generality. To determine the location of each node, we used MaxMind's GeoLite Country database (http://dev.maxmind.com/geoip/legacy/geolite), which maps IP address blocks to a country. MaxMind mapped IP addresses in the PlanetLab ping trace to 22 "countries," which we can also group into five "continents".[1]

Our settings for the Vivaldi simulator were set to two dimensions without height and 56,500 rounds (250 times the number of nodes). We discuss the consideration of higher dimensions or a different coordinate system in the Discussion section. To investigate how well countries are grouped in the 2-D network coordinate space, we ran classifiers from the Weka toolkit to classify nodes in the network coordinate system by country. The Naive Bayes and J48 decision tree classifiers were able to classify nodes by country, each having about 18% misclassified instances. There was good precision and recall for the countries with more nodes (e.g. U.S., China) and poorer precision/recall for countries with fewer nodes. This demonstrated that network coordinates permit thinking of avoidance at a country level of granularity.

Given our results and the clear clustering by country, using Maxmind works well for looking up country from IP address. At least for this PlanetLab set of 226 nodes, the 2-D network coordinates do lead to clusters by countries and are classifiable. Ideally, we would analyze a dataset with more nodes that are more geographically diverse and non-PlanetLab data that is more "organic" (since Ledlie et al. found that PlanetLab data are not representative of real networks).

Accounting for Uncertainty

Network coordinates based on latency measures are an approximation and could be off by some margin of error, especially since networks fluctuate, latencies can vary, and the triangle inequality does not always hold in Internet routing [12]. As a result, we need to tolerate some uncertainty in the network coordinates and build this into our checks for avoidance. We take a conservative approach and only choose relays we are certain allow us to avoid a given region. The certainty margin ε defines the $\ll$ expressed in our inequalities proposed in Alibi Routing. We can define ε in terms of raw latency (in milliseconds) or relative latency (a percentage). We use the latter for the purposes of our simulation; the $\varepsilon = 10\%$ certainty margin means that $a \ll b$ if and only if b exceeds a by more than 10%: $\frac{b-a}{a} > 10\%$. In situations of high uncertainty, a higher margin should be used because it means we have a higher threshold for certainty that we avoided the region.

iPlane Dataset

We began with the University of Washington iPlane datasets [14] but found that their estimates did not result in a representative distribution of nodes in the network coordinate space. The datasets include latency estimates between points of presence (PoPs), mappings from IP addresses to PoPs, and location information about some IP addresses as either latitude-longitude or a city and country. Initial analysis of the results in the network coordinate system showed high error values, clear violations of basic properties, and poor conditions for avoidance. In particular, we discovered the following oddities with the iPlane latency data that lead us to avoid using them in our study:

1. For 12 out of 47 countries, the median distance for an intranational link exceeds that of international links. Since the median latency estimates for intranational links were reasonably less than those of international links, we expect a similar relationship for distances within the network coordinate system. It should not be the case, for instance, that links within China take much longer than China's typical link to other countries.

[1] MaxMind's database mapped one of the PlanetLab nodes to "Europe" rather than a specific country. We treat Israel as part of the Middle East instead of Asia, because its latency behavior more closely resembles that of Europe than Asia in our results.

2. For 113 out of 695 pairs of countries, the latency estimate appears to violate physical laws. We used the Wolfram Alpha API to look up the time it would take the speed of light in fiber to travel the great circle distance between the centers of two countries. Comparing this time to the average latency for each pair of countries roughly checks whether the latency estimate is possible. For example, a link from the U.S. to Saudi Arabia with iPlane's latency estimate of 1ms is unreasonable given that the speed of light would take 56ms to travel the great circle distance.
3. There is very little separation/clustering of countries within the network coordinate system, as shown by plots color-coded by country and poor classifier and clustering results. Most countries were centered around the origin.

We suspect the latencies and resulting network do not reflect the actual network topology, based on the physical violations and the lack of obvious clustering that the PlanetLab results have. Regardless of which country the nodes belong to, the PlanetLab trace shows distinct clusters in the coordinate space that are not present with the iPlane data. For these reasons, we used the Harvard data only.

Results

How do geographic coordinates map to network coordinates? Our simulation over the PlanetLab trace gave us a better sense of the mapping between geographic and network coordinates and the kind of areas that can be avoided. In particular, we wanted to see if contiguous areas such as a country in one system would still be continuous in the other. We found that geographic coordinates map fairly well to 2-D network coordinates without height. Contiguous areas such as countries end up clustered in the network coordinate system, as shown by a plot of network coordinates color-coded by country (Figure 2). There is some overlap of countries that are geographically close, due to international links with latencies shorter than intranational links. As evident in Figure 2, limited data can make it difficult to clearly separate "borders" of neighboring countries, such as U.S. and Canada, in the network coordinate space. At a coarser level, however, it is easy to separate larger regions such as continents — Asia, North America, and Europe visibly map to distinct, separated clusters.

There is a strong correlation between median latency between two given countries and the corresponding network coordinate distance, based on plots of latency against distance (Figure 3). This verifies that the

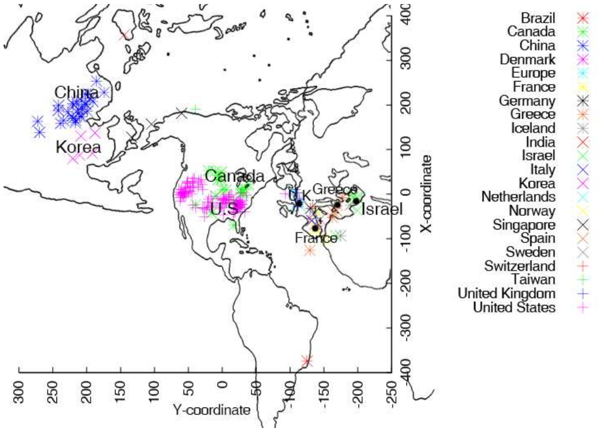


Figure 2: Plot of 2-D network coordinates, color-coded by country, overlaid over transformed[2] geographic regions.

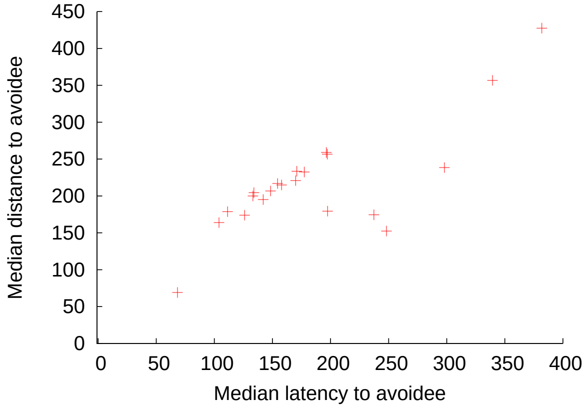


Figure 3: Plot of median latency against median distance for United States to the other countries.

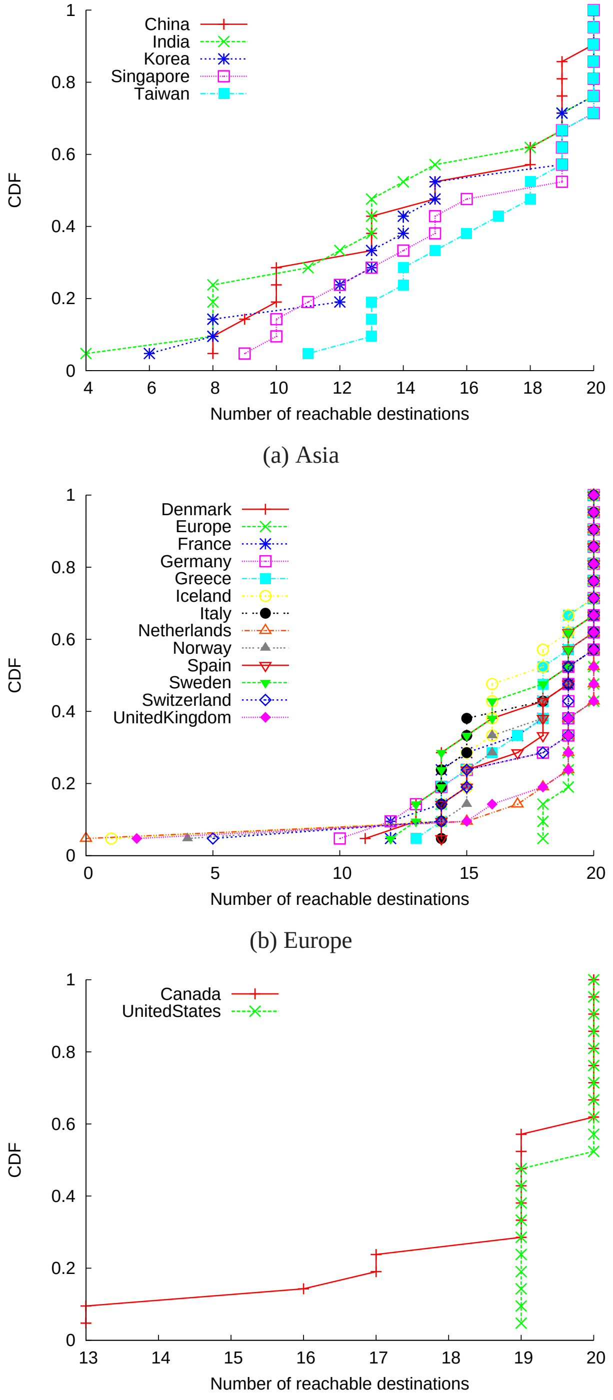
[2] The three geographic regions were individually rotated, scaled, and skewed to roughly match countries' locations in the network coordinate space.

transformation to network coordinates preserves the relationships among countries and that we can take advantage of the correlations among geographic distance, network latency, and network coordinate distance.

What kind of areas can be avoided, and under what circumstances? After mapping geographic areas to network coordinates, the next question is what kind of areas can be avoided and when. The areas that can be provably avoided depend on the source, destination, and available relays, particularly their geographic locations and latencies to other countries. Our plots focus on "reach": the number of reachable destinations that a source can successfully reach given an avoidee (rather than the possible avoidees given a destination). A source country's reach seems closely tied to the continent. For each source country, we plot the CDF of reach values over all of the avoidees (Figure 4). The shape of the curve appeared distinct across continents but fairly similar for countries from the same continent.

Though the shape of the curve is consistent for a continent, some countries have more reach than others (the curve is farther to the right), perhaps due to development. For example, within Asia, Taiwan and Singapore have better reach than China and India (Figure 4a). PlanetLab's data and its geographic distribution of nodes may bias the results.[3] U.S. had the highest reach, followed by Europe and then Asia. Overall, there are many opportunities for provable avoidance, with geography significantly influencing who can avoid whom.

In the plots of number of reachable destinations against distance to avoidee (Figure 5), we generally see that reach depends primarily on geography rather than distance to the avoidee when avoiding a particular country. Our initial hypothesis was that the plots would indicate a positive linear relationship between distance to avoidee and reach, where being farther from the avoidee increases ability to reach destinations successfully. Instead, we see the results depend on the continents of the source and avoidee rather than distance. For instance, the plots for other Asian countries resemble that of China, with generally high reach for Asia and the Americas but variable reach for Europe. When sending from Brazil or Asia and avoiding a European country, the reach varies with the avoidee even when distance to the avoidee is roughly the same. Our initial results when using two relays instead of one (see Multiple Relays and Avoidees) were similar; Brazil and China had slightly increased reach for the European countries but it was still determined by continent.



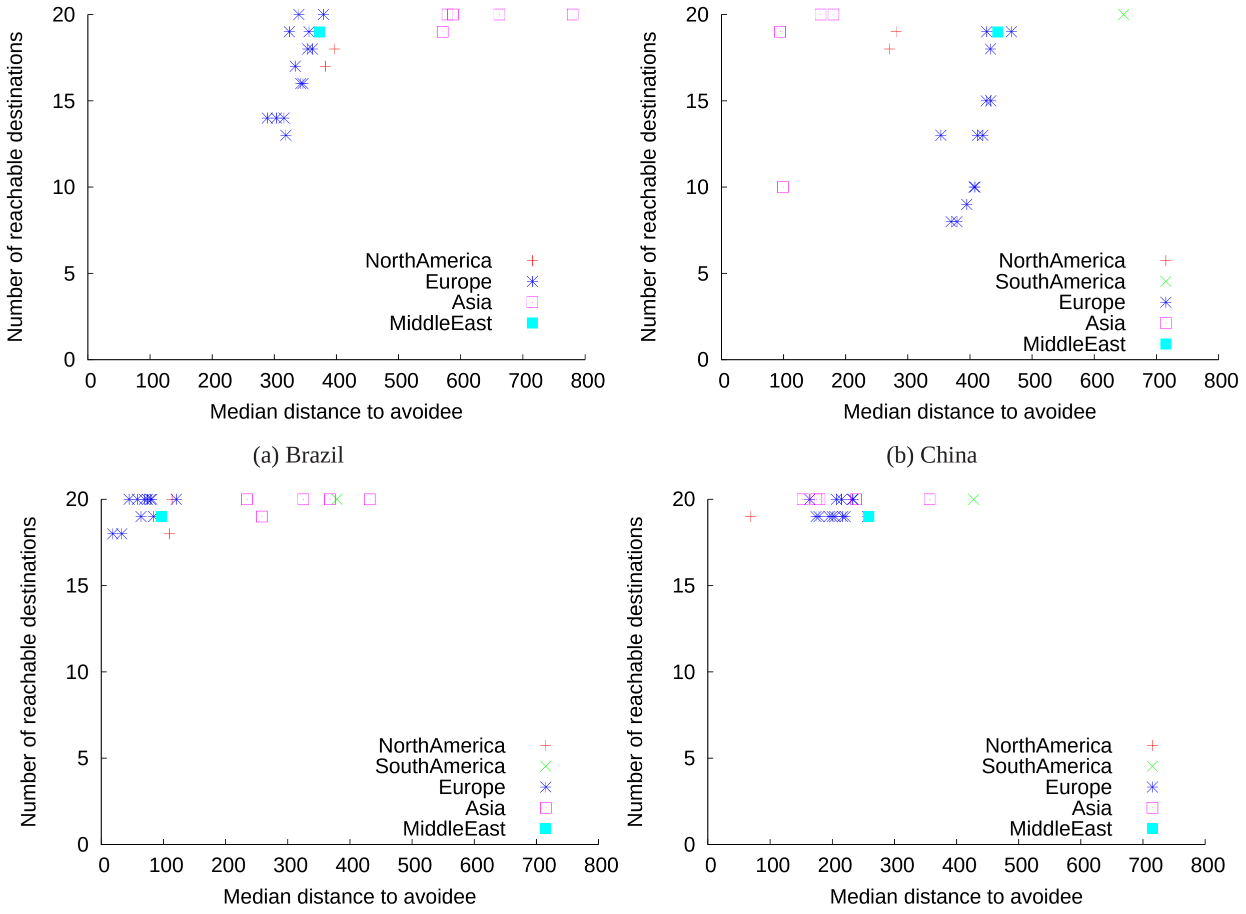
(a) Asia

(b) Europe

(c) North America

Figure 4: For the 10% certainty margin, the CDF of number of reachable destinations over different avoidees. Series are grouped by the continent of the source country.

[3] More than half of the countries in the dataset are in Europe, so reach is biased toward European destinations.



(a) Brazil

(b) China

(c) Europe

(d) United States

Figure 5: For a particular source country, the median distance against the number of reachable destinations for different avoidees at the 10% certainty margin.

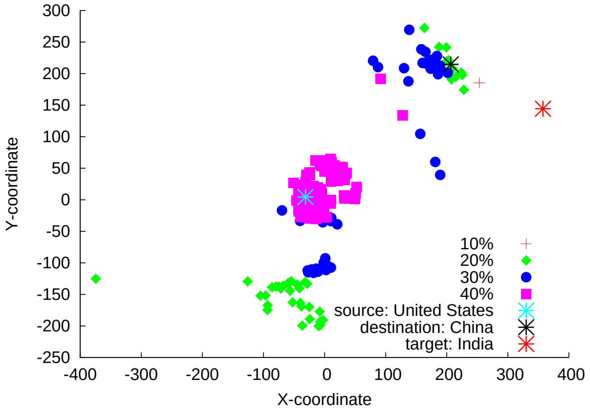


Figure 6: Plot of all nodes as potential relays when sending from United States to China and avoiding India. Relays are color-coded by the allowable uncertainty (ε values).

The distribution and placement of nodes throughout the network determines who has the best reach and who can be easily avoided. For instance, with so many countries in Europe, European countries have good reach while Brazil and Asian countries are less able to find an effective relay for many destinations. Having the avoidee far from the path SRD makes successful avoidance more likely, especially in situations of high uncertainty (as shown by relays color-coded by allowable uncertainty in Figure 6).

Does avoidance require accurate coordinates? Even when allowing for high uncertainty, there are still many instances of successful avoidance. As uncertainty increases, the separation by continent becomes more evident, as shown in Figure 7 (i.e. countries from the same continent exhibit less variation in reach, converging toward the same curve). As mentioned previously, having the avoidee far from the path SRD improves chances for avoidance when there may be high uncertainty. ε serves as a confidence measure based on $min(L_{SRAD}, L_{SARD}) - L_{SRD}$ (how much longer a path including the avoidee takes). This implicitly specifies the error in the network coordinates that we tolerate.

How much delay does avoidance cost? To analyze costs, we can look at the CDF plots of median additional cost. We first drop the (source, destination, avoidee) triples where the direct path without using a relay may suffice (see Multiple Relays and Avoidees section). In fact, for 72.8% of the $22 * 21 * 20 = 9240$ triples, the direct path is enough to prove avoidance and relays are not necessary. With the remaining scenarios that require and can provably avoid using a single relay, we find the minimum additional cost for each (source, destination, avoidee) triple. We then aggregate over the destinations and reduce to (source, avoidee) pairs by taking the median cost over all of the destinations.[4] Finally we plot the CDF of this median additional cost for the particular source country over all avoidees. For scaling purposes, we plot with a log scale. For many of the source countries, we see in Figure 8 that costs are low (less than 10ms) for a majority of the avoidees.

We hypothesized that as uncertainty increases, the cost of avoidance would increase since relays have to be farther from the avoidee. Instead, we find that though reach decreases, the costs for many countries still remain low to get to destinations that are still reachable. We again see that the curve is closely related to which continent the country is in. The countries with higher costs for some avoidees are generally non-European, due to the geographic distribution of our dataset (more European countries). In general, the factors that affect cost seem to be geographic location (e.g. continent, country), the country's reach, and relative placement of relays. This indicates that soliciting participation from widely-positioned relays can significantly improve performance.

Discussion

Through simulation and initial analysis, we investigated the possibility of route avoidance and a potential method of proving successful avoidance. The next step would be to use these observations in designing and building an implementation. Specific applications where route avoidance would be useful to implement as a plugin include the Chrome web browser and Tor overlay routing [8], which supports anonymous communication.

Some remaining open questions are better informed by a system implementation and evaluation, such as the following:

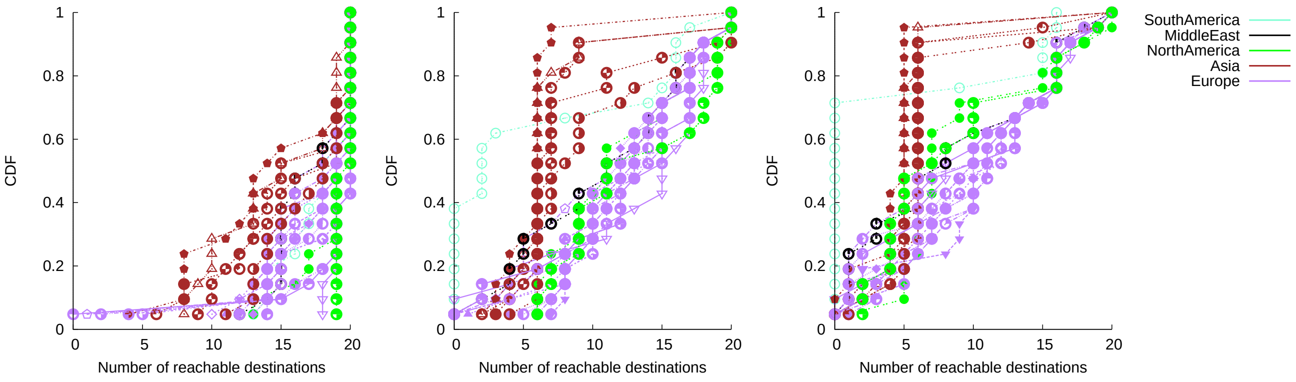
- How often do network coordinates change and by how much, such that new relays need to be selected?
- What incentives exist for participation? If a larger and well-distributed network allows for more avoidance scenarios, what aspects of the system's design can encourage more nodes to join the overlay network?
- How well does the system perform under various attack models?

In designing the system, we would want a more principled approach to measuring and reasoning about error in latency measurements and network coordinates.

Relay Selection

Our simulation revealed who can avoid whom and evaluated potential relays based on global knowledge, but

[4] We use an additional cost value of -1 when there are no destinations for the particular source and avoidee that require or can be provably reached with a single relay.



(a) $\varepsilon = 10\%$ (b) $\varepsilon = 30\%$ (c) $\varepsilon = 50\%$

Figure 7: The CDF of number of reachable destinations over different avoidees for all source countries, shown with different values of ε. Series are colored by the continent of the source country.

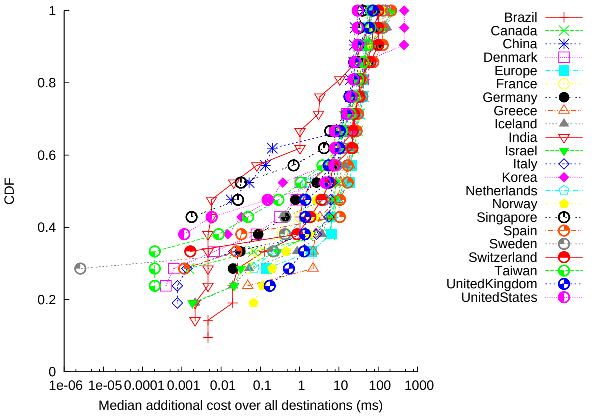


Figure 8: CDF of log-scaled median additional cost to destinations over all avoidees for $\varepsilon = 10\%$.

our system implementation has to be able to choose the "best" relay to use based on local knowledge. Sherpa [13] can help with relay selection by finding the node that minimizes a specified cost function. Intuitively, the cost function has to state which nodes can serve as relays given our ε and of these, which ones are better or worse. Once we know a relay can lead to successful avoidance, we want to minimize the additional cost in terms of latency — how much longer is the path through the relay than the direct path? This reasoning produces the following cost function for some potential relay R:

$$cost(R) = \begin{cases} \infty & \text{if (1a) or (1b) does not hold,} \\ L_{SRD} & \text{otherwise.} \end{cases}$$

In other words, ε determines the radius for some circle around A where the cost is ∞. Any nodes in this circle are too close to A to serve as a good relay. For the nodes outside of this circle, the cost is simply the latency of the path SRD. (We do not need to subtract L_{SD} since it is constant.)

Multiple Relays and Avoidees

For many (source, destination, avoidee) triples in our dataset, there does not exist a provable avoidance path using only one relay. Our initial results from investigating two-relay paths revealed generally increased reach but also increased additional costs. To expand our check for successful avoidance to multiple relays, we simply extend the inequalities in (1). For example, with two relays R_1 and R_2, we compare the "safe" path latency $L_{SR_1R_2D}$ in each of

$$L_{SR_1R_2D} \ll L_{SAR_1R_2D} \tag{2a}$$
$$\ll L_{SR_1AR_2D} \tag{2b}$$
$$\ll L_{SR_1R_2AD}. \tag{2c}$$

For R_1 and R_2 to generate an alibi, any path including R_1 and R_2 that also goes through A must have much greater latency than the path that does not go through A. A more general definition is that for N relays, we have to check $N + 1$ inequalities, since A could be inserted between any two nodes in the safe path. Note that this also holds for the 0-relay case, where the direct path is enough to prove a packet did not go through A. With 0 relays, we have one inequality comparing SD to SAD. If $SD \ll SAD$, we can prove avoidance and do not incur any additional latency costs.

Sherpa [13] only supports looking for the single best relay, so in order to support multiple relays while maintaining a decentralized, local-knowledge approach, we introduce "scaffolding." Based on local knowledge, if a source cannot prove avoidance using a single relay, then it greedily chooses a relay with the highest chance of avoiding the destination using additional relays. Treating this relay as a virtual source, this process continues recursively until we reach the second-to-last relay, which is able to prove avoidance using the single-relay approach discussed earlier. Using the scaffolding technique, the source aims to choose as its next relay an existing neighbor who is able to successfully avoid. To illustrate, assume that some neighbor R_1 is able to reach D avoiding A using one relay R_2. (R_1 uses the approach defined earlier to select R_2.) By the alibi routing approach, this means the following two inequalities hold:

$$\begin{aligned} L_{R_1 R_2 D} &\ll L_{R_1 A R_2 D} \\ &\ll L_{R_1 R_2 A D} \end{aligned}$$

We can add S to the beginning of each path because L_{SR_1} is a constant latency. This gives us (2b) and (2c):[5]

$$\begin{aligned} L_{SR_1 R_2 D} &\ll L_{SR_1 A R_2 D} \\ &\ll L_{SR_1 R_2 A D} \end{aligned}$$

To prove successful avoidance using relays R_1 and R_2, we only have to additionally show $(2a)$. Since $L_{R_1 R_2 D}$ is constant, we simplify $(2a)$ to

$$L_{SR_1} \ll L_{SAR_1}. \qquad (3)$$

Thus, if S cannot avoid A using one relay, it simply has to select as its next relay a node R_1 such that R_1 can avoid A in one relay and (3) holds. Under these conditions, S can avoid A using two relays. More generally, if $L_{SR_1} \ll L_{SAR_1}$ and R_1 can avoid A using N relays, S can avoid A using $N+1$ relays (the first of which is R_1).

There may also be practical applications where the source wishes to provably avoid multiple avoidees at once. We could integrate obstacle avoidance algorithms from artificial intelligence, which find the ideal point(s) (relays) for navigating around the obstacles (avoidees). This would likely require multiple relays, so the complete solution would involve some path-finding AI method as well as multiple relay scaffolding.

Alternative Coordinate Systems

While we focused on mapping to a 2-D Euclidean space, we can also consider adding height and/or more dimensions or switching to a non-Euclidean (e.g. spherical) space. The original Vivaldi work [7] found spherical coordinates to be less effective, positing that Internet paths do not wrap around the Earth. However, Agarwal and Lorch found that spherical coordinates with height worked well in Vivaldi when coordinates were initialized using geographic coordinates.

Based on Figure 2 and the latency data, we observed that paths between Asia and Europe may indeed travel through North America. For many European countries, the median "direct" latency from China to the country roughly equals the latency of the "indirect" path from China to the United States to the destination. In fact, for all European countries except the United Kingdom and the "Europe" classification, the indirect latency through the United States is within 3.3% of the direct latency. As an example, the median latency from China to Denmark is 446ms, from China to the United States is 290ms, and from the United States to Denmark is 154ms. Comparing $290 + 154 = 444$ms for the indirect path to the measured 446ms for the direct path, we find it within reason that packets from China to Denmark may travel through North America. These observations support (1) Internet paths do not wrap around the earth and (2) the 2-D mapping in Figure 2, where North America lies between Asia and Europe, is representative of the Internet. Adding more dimensions may make it easier to separate neighboring countries, but we also do not want to overspecify the model when we have limited data.

Acknowledgments

We would like to thank Cristian Lumezanu, Matt Lentz, Vassilios Lekakis, and Brandi Adams for their insightful conversations.

References

[1] Sharad Agarwal and Jacob R. Lorch. Matchmaking for online games and other latency-sensitive P2P systems. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 315–326, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-594-9. doi: 10.1145/1592568.1592605. URL http://doi.acm.org/10.1145/1592568.1592605.

[2] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, SOSP '01, pages 131–145, New York, NY, USA, 2001. ACM. ISBN 1-58113-389-8. doi: 10.1145/502034.502048. URL http://doi.acm.org/10.1145/502034.502048.

[3] Anonymous. The collateral damage of Internet censorship by DNS injection. *SIGCOMM Comput. Commun. Rev.*, 42(3):21–27, June 2012. ISSN 0146-4833. doi: 10.1145/2317307.2317311. URL http://doi.acm.org/10.1145/2317307.2317311.

[4] Eric Chan-Tin and Nicholas Hopper. Accurate and provably secure latency estimation with Treeple. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011*.

[5] This requires $\ll$ to be defined in terms of absolute rather than relative latencies, unlike our simulation.

The Internet Society, 2011. doi: http://www.isoc.org/isoc/conferences/ndss/11/pdf/7_1.pdf.

[5] Eric Chan-Tin and Nicholas Hopper. KoNKS: konsensus-style network koordinate system. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 61–62, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1648-4. doi: 10.1145/2414456.2414491. URL http://doi.acm.org/10.1145/2414456.2414491.

[6] Eric Chan-Tin, Victor Heorhiadi, Nicholas Hopper, and Yongdae Kim. The frog-boiling attack: Limitations of secure network coordinate systems. *ACM Trans. Inf. Syst. Secur.*, 14 (3):27:1–27:23, November 2011. ISSN 1094-9224. doi: 10.1145/2043621.2043627. URL http://doi.acm.org/10.1145/2043621.2043627.

[7] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: a decentralized network coordinate system. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, pages 15–26, New York, NY, USA, 2004. ACM. ISBN 1-58113-862-8. doi: 10.1145/1015467.1015471. URL http://doi.acm.org/10.1145/1015467.1015471.

[8] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: the second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1251375.1251396.

[9] B. Donnet, B. Gueye, and M.A. Kaafar. A survey on network coordinates systems, design, and security. *Communications Surveys Tutorials, IEEE*, 12 (4):488 –503, quarter 2010. ISSN 1553-877X. doi: 10.1109/SURV.2010.032810.00007.

[10] Erik Kline and Peter Reiher. Securing data through avoidance routing. In *Proceedings of the 2009 workshop on New security paradigms workshop*, NSPW '09, pages 115–124, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-845-2. doi: 10.1145/1719030.1719046. URL http://doi.acm.org/10.1145/1719030.1719046.

[11] Jonathan Ledlie, Paul Gardner, and Margo Seltzer. Network coordinates in the wild. In *Proceedings of the 4th USENIX conference on Networked systems design & implementation*, NSDI'07, pages 22–22, Berkeley, CA, USA, 2007. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1973430.1973452.

[12] Cristian Lumezanu, Randy Baden, Neil Spring, and Bobby Bhattacharjee. Triangle inequality variations in the internet. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 177–183, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-771-4. doi: 10.1145/1644893.1644914. URL http://doi.acm.org/10.1145/1644893.1644914.

[13] Cristian Lumezanu, Dave Levin, Bo Han, Neil Spring, and Bobby Bhattacharjee. Don't love thy nearest neighbor. In *Proceedings of the 9th international conference on Peer-to-peer systems*, IPTPS'10, pages 5–5, Berkeley, CA, USA, 2010. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1863145.1863150.

[14] Harsha Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: an information plane for distributed services. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*, OSDI '06, pages 26–26, Berkeley, CA, USA, 2006. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1267308.1267334.

[15] Jeffrey Seibert, Sheila Becker, Cristina Nita-Rotaru, and Radu State. Securing virtual coordinates by enforcing physical laws. In *Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems*, ICDCS '12, pages 315–324, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4685-8. doi: 10.1109/ICDCS.2012.22. URL http://dx.doi.org/10.1109/ICDCS.2012.22.

[16] Micah Sherr, Matt Blaze, and Boon Thau Loo. Veracity: practical secure network coordinates via vote-based agreements. In *Proceedings of the 2009 conference on USENIX Annual technical conference*, USENIX'09, pages 13–13, Berkeley, CA, USA, 2009. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1855807.1855820.

[17] Shining Wu, Yang Chen, Xiaoming Fu, and Jun Li. Ncshield: Securing decentralized, matrix factorization-based network coordinate systems. In *Quality of Service (IWQoS), 2012 IEEE 20th International Workshop on*, pages 1–9, 2012. doi: 10.1109/IWQoS.2012.6245983.