# Exploiting Internet Delay Space Properties for Selecting Distinct Network Locations

Bo Zhang and T. S. Eugene Ng
Department of Computer Science
Rice University

***Abstract*—Recent studies have discovered that the Internet delay space has many interesting properties such as triangle inequality violations (TIV), clustering structures and constrained growth. Understanding these properties has so far benefited the design of network models and network-performance-aware systems. In this paper, we consider an interesting, previously unexplored connection between Internet delay space properties and network locations. We show that this connection can be exploited to select nodes from distinct network locations for applications such as replica placement in overlay networks even when an adversary is trying to mis-guide the selection process.**

## I. Introduction

Recent studies [32] [14] [29] [17] have identified many interesting properties of the Internet delay space[1] , such as triangle inequality violations (TIV), clustering structures and constrained growth. With the increased understanding of Internet delay space properties, researchers have started applying them to solve some practical problems. For examples, [32] proposes a network delay model that takes the delay space properties into account, [29] improves the performance of two neighbor selection systems by making them TIV-aware, and [18] proposes a routing overlay that exploits TIV to select the best peerings. In this paper, we show that the Internet delay space properties can also be leveraged to select nodes from distinct network locations even when an adversary is trying to mis-guide the selection process, so that those applications needing nodes from distinct network locations can be benefited.

### A. The Need For Distinct Network Locations

In a decentralized distributed system such as peer-to-peer (P2P) systems, when a node needs to request service from a set of other nodes, it often prefers a set of nodes from distinct network locations because more network location diversity can generally increase the system's availability and make the system more resilient to locality specific network failures. Therefore, the ability to select multiple nodes from distinct network locations is an important primitive for many systems.

Some specific examples where the network location diversity is preferred are as follows: **Overlay routing:** in an overlay network (e.g., [26], [28], [24], [19], [4]), each node needs to use a number of other nodes as its overlay routing neighbors. Depending on specific overlay networks, the number of needed routing neighbors can range from tens to hundreds. Choosing neighbors in distinct network locations will increase route diversity and improve the overlay's robustness against network failures. **Proactive object replication:** The benefits of proactive object replication in structured overlays in reducing overlay lookup hops and latency have been exploited by Beehive [23]. When replicating an object on a set of overlay nodes, it is better that those nodes have diverse network locations for the sake of both better performance (e.g., minimize the average query latency) and better reliability (e.g., one replica getting disconnected will not affect the whole system). **Detecting Sybil identities from same network location:** P2P systems use logical identities to distinguish peers, so P2P systems are particularly vulnerable to Sybil attacks [10], where a malicious node assumes multiple identities, which are called Sybil identities. In fact, such Sybil attacks have already been observed in the real world in the Maze P2P file sharing system [15][31]. The implementers of the Maze system instrumented the Maze client so they can obtain and examine the complete user logs of the entire system. By analyzing these logs, they found that most colluding Sybil identities use *nearby machines* from the same network location. They hypothesize that creating Sybil identities in the same location makes it easier to control them and leverage the network proximity to maximize the throughput and the gain from collusion. Researchers [27] have also demonstrated that it is surprisingly easy to launch Sybil attacks from a single network location in the widely-used eMule system [1]. In their experiments, they created up to 64K distinct Sybil identities (i.e., 64K KAD IDs) on one physical machine, then they were able to spy on the whole system's traffic and launch DDoS attacks on any content. The ability to select identities from distinct network locations would reduce a system's susceptibility to Sybil attacks.

### B. Challenges of Selecting Distinct Network Locations

The first strawman solution for selecting nodes or identities (we use these two words interchangeably) from distinct network locations is to select nodes with distinct IP addresses (e.g., [11]). However, a malicious node may steal multiple IP addresses from the local network. What is worse, it may hijack a large number of IP addresses from diverse network

This research was sponsored by the NSF under CAREER Award CNS-0448546 and grant CNS-0721990. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF or the U.S. government.

[1]In this paper, "delay" means round-trip delay.

locations [13] [5] and give each fake identity its own unique IP address for communications.

The second strawman solution is to use the traceroute tool to check whether different identities share the same upstream router. If so, they are definitely from the same network location. However, a malicious node may reply to traceroute messages with different fake network hops for each identity to pretend that the identities originate from different network locations.

Another idea is to require each participating identity to provide its real physical location. This must involve some centralized authority (CA) who can verify each identity's physical location using private information such as a social security number, a driver's license number, etc. This approach exposes users' privacy and the required CA is not always available in a decentralized distributed systems.

Bazii et al. [6] propose to assign each node a *geographical certificate* based on network coordinates [20] [8]. The proposal is to use these geographical certificates to differentiate nodes from different network locations. However, the proposal assumes an idealized network where delays satisfy metric space properties such as triangle inequality and ignores the real properties of Internet delays. Therefore, in practice, the proposal has limited applicability.

### C. Exploiting Internet Delay Space Properties

In this paper, we propose a novel approach that uses Internet delays as "fingerprints" to identify distinct network locations. Distributed systems can pick nodes with different "fingerprints" to select nodes from distinct network locations. Even when an adversary is trying to mis-guide the selection process by manipulating network delays, we show that properties of the Internet delay space can be used to greatly limit a node's ability to fake its network locations.

The effect is that no matter how many network locations a malicious node tries to fake by manipulating delays, it can only appear to reside in a very small number of credible network locations. Our technique exploits two properties of the Internet delay space: (1) a network location cannot have small delays to all other network locations, and (2) if the delays from a network location to other network locations have been inflated heavily, the resulting delays will have unusual statistical properties.

To quantify the benefits of our technique, we conduct measurement-based experiments. In one experiment, when 6,000 legitimate peers are randomly scattered over the Internet, and the malicious node is placed at a random network location creating over 13 million identities on average by exhaustively manipulating delays in 1 millisecond increments, our technique can limit the percentage of fake identities accepted by a legitimate node to below 5%, i.e., if a legitimate node selects 100 peers among 6000 legitimate peers and 13 millions fake identities, no more than 5 fake identities are selected.

The rest of this paper is organized as follows. We establish the relevant delay space properties using Internet measurements in Section II. We present our technique and provide empirical justifications in Section III. The technique's effectiveness and characteristics are discussed in Section IV. We discuss several additional details in Section V. Section VI presents the related work, and we conclude the paper in Section VII.

## II. Properties of the Internet Delay Space

Our technique uses a set of trusted distributed landmarks (such as Planetlab [21] nodes) to measure their delays to an identity and then assign a "fingerprint" to the identity. Therefore, in order to study how the properties of the Internet delay space can be useful in selecting distinct network locations, we first collect Internet delay measurements using Planetlab. Our data collection methodology is presented in Section II-A. Two interesting properties of the Internet delay space that our technique relies on are introduced in Section II-B.

### A. Data Collection Methodology

As described above, our technique uses a set of trusted landmark nodes to measure other regular nodes. In our measurements, the two types of nodes are selected as follows:

**Landmark selection:** We use the Planetlab testbed (consisting of 826 machines in 406 sites) as the candidate landmarks. We select one machine from each Planetlab site and then keep the 100 machines with the lightest workload. We do not use those overloaded machines because the measurements performed from them may be skewed.

**Live IP addresses for simulating regular nodes:** In order to choose live IP addresses to simulate nodes on the Internet, we start with a list of 20,000 random IP addresses drawn from the prefixes announced in BGP as published by the Route Views project [25]. We probe each IP address to test whether it responds to ICMP Echo Request, and finally we get 7,000 live IP addresses that respond to our ICMP probes. We will use these 7,000 IP addresses to simulate regular nodes in this paper. Because the IP addresses are randomly chosen, they should be able to approximate the Internet delay space.

Note that in an actual implementation, using ICMP Echo request to measure delay is only one of many options. If a node does not respond to ICMP Echo request because it is turned off or it is behind a firewall, we can use its last-hop router to represent it. Transport-level ping or application-level ping may also be used to measure delays.

**Probing:** We let each selected landmark machine probe all the 7,000 live IP addresses to measure the round-trip time (RTT) between the landmarks and all IP addresses. Each IP is probed 5 times from each landmark and the minimum of the 5 delay samples is used. This produces a $100 \times 7{,}000$ delay matrix for our study. The landmarks also probe each other to measure the delays among themselves using the same probing methodology.

The following discussion about the Internet delay space properties and the empirical results presented in Section III and IV are based on this data set.

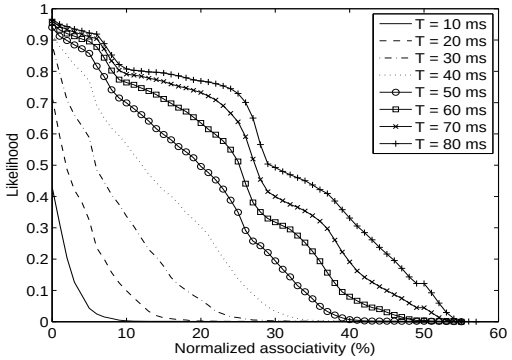


Fig. 1. Associativity of network nodes with different $T$.

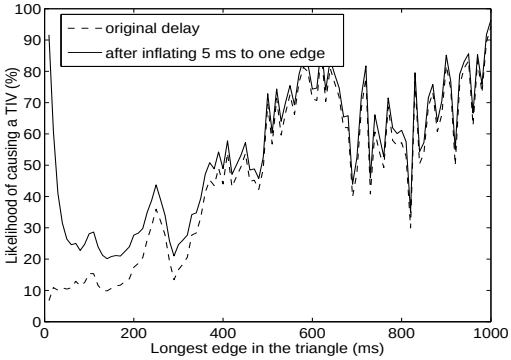


Fig. 2. Likelihood of causing a TIV in one triangle with respect to the longest edge in the triangle.

### *B. Two Properties of the Internet Delay Space*

In this section, we explain the two properties of the Internet delay space that our technique is based on. These properties represent a previously unexplored connection between Internet delay space properties and network locations.

**• Property 1: A network location cannot have small delays to all other network locations.**

This property is straightforward to see because the delay between any two network locations is ultimately lower bounded by the speed of light delay across the physical distance between the two network locations.

In order to quantify this property, we study how many landmarks one node can be close to in our data set. We define the *associativity* of one node $N$ given an *associativity threshold* $T$ as the number of landmarks that are within $T$ distance to $N$. The *normalized associativity* is just the associativity divided by the total number of landmarks. Figure 1 shows the normalized associativity with different associativity thresholds $T$. As can be seen, given a reasonable $T$, the likelihood that one node can be associated with a large fraction of landmarks is small. For example, given $T = 30$ ms, the likelihood that a random network node can be associated with more than 30% of the landmarks is nearly zero.

**• Property 2: If the delays from a network location to other network locations have been inflated heavily, the resulting delays will have unusual statistical properties.**

This property can be further interpreted in two folds:

*Property 2.1: Triangle inequality violations (TIV) widely exist but they happen far less frequently among nearby nodes. If a node inflates its delays to nearby nodes, it is very likely to cause unusual TIVs.*

Internet delays do not always obey the triangle inequality property because the Internet routing is not always optimal with respect to delays. Studies [32] [29] have shown that a small fraction of triangles in the Internet delay space violate triangle inequality and that long delays are more likely to cause a TIV. We have confirmed this property based on our Planetlab data set. The dotted line in Figure 2 shows the likelihood of one triangle causing a TIV with respect to the longest edge in that triangle.

The question then is, how does inflating delays change the TIV characteristics? The solid line in Figure 2 shows the likelihood that one triangle will cause a TIV if one of its edges is inflated by a small amount of delay, specifically 5 ms. As can be seen, triangles with small delays are highly sensitive to such small delay inflation, resulting in an unusually high likelihood of TIV. This result indicates that if we consider triangles with small delays, triangles with manipulated delays can be detected by inspecting their TIVs.

*Property 2.2: The Internet delay space forms multi-level clusters due to the heterogeneous physical distribution of nodes. Thus, the delays among a set of nodes conform to the clustering structure and cannot be random. If a node inflates delays to other nodes arbitrarily, those delays may not conform to the characteristics of delays found in a normal delay space.*

Internet hosts are not randomly distributed and thus the Internet delay space has a non-uniform structure. Studies such as [32] have shown that the continents (North America, Europe and Asia) with the largest concentration of IP subnetworks form recognizable clusters in the delay space. In addition to the global-scale clustering structure in the delay space, within each continent Internet hosts are concentrate in populated areas like big cities and form local clusters. This clustering property indicates that a node is highly unlikely to appear at an arbitrary location in the Internet delay space, i.e., it cannot have arbitrary delays to other nodes. To illustrate this property, in Figure 3, we use nodes on a 2-D plane to represent hosts in the Internet. Consider the top half of the figure. Assume node $X$ is originally located in a local cluster and the delays from $X$ to two landmarks $L_1$ and $L_2$ are represented by the dashed arrows. From the points of view of landmarks $L_1$ and $L_2$, node $X$ appears to have a legitimate location because two other nodes also reside in the same neighborhood. But if $X$ inflates its delays to landmarks $L_1$ and $L_2$ (the delays after inflation are represented by solid arrows in the bottom half of Figure 3), it will appear to have a new and unusual location $X'$ to landmarks $L_1$ and $L_2$, where no other legitimate nodes exist.

The Internet delay space is certainly not as simple as a 2-D plane. We need to quantify whether a set of delays is normal or unusual in the delay space. Given a set of landmarks $(L_1, L_2, ..., L_n)$, the *fingerprint* of a node $i$ is defined as the

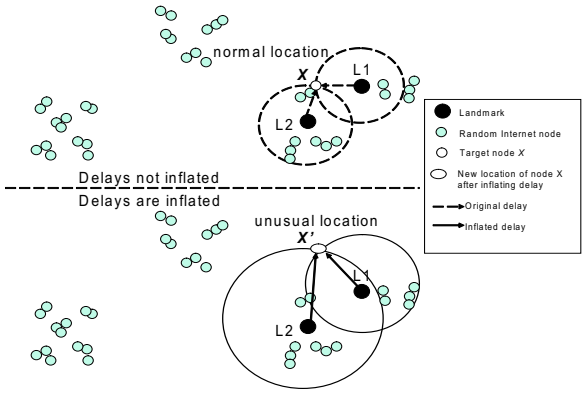


Fig. 3. Nodes cannot appear in an arbitrary location in the Internet delay space: if $X$ does not inflate delays, it should appear at a normal location clustered with other nodes; if it does inflate delays, it will appear to be located in an unusual location.

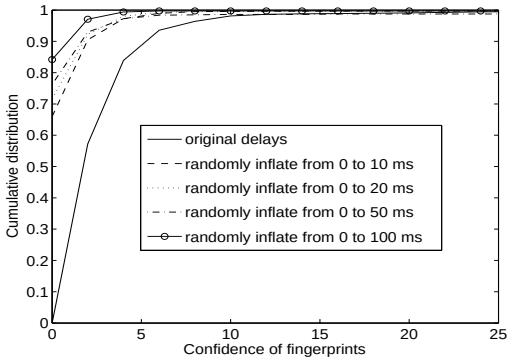


Fig. 4. If nodes inflate delays randomly, their confidence measure will be very likely to be lower than those of legitimate nodes.

delay vector composed of delays from a number of landmarks to node $i$: $(d_i^1, d_i^2, ..., d_i^m)$, where $m \leq n$ and $d_i^k$ is the delay between landmark $L_k$ and node $i$. Given any two fingerprints $(d_i^1, d_i^2, ..., d_i^m)$ and $(d_j^1, d_j^2, ..., d_j^m)$ for node $i$ and node $j$, the *distance between the two fingerprints* is defined as the Manhattan distance of the two fingerprints: $\sum_{k=1}^{m} |d_i^k - d_j^k|$. Furthermore, we assign a *confidence* value to each fingerprint by counting the number of fingerprints of legitimate nodes in our data set that are within certain *confidence threshold* $t$ to this fingerprint.

Given the above definition, we randomly select three landmarks and then generate a fingerprint for each node in our data set based on the selected three landmarks, then we can calculate the confidence values of all the 7,000 fingerprints. For comparison, we also calculate the confidence values of fake fingerprints. The fake fingerprints are generated by inflating the delays in the original 7,000 fingerprints by certain random values. Figure 4 shows the comparison result using $t = 3$ms. As can be seen, the confidence values of those fake fingerprints are much lower than the legitimate fingerprints. And the more heavily the delays are inflated, the lower are their confidence values. This indicates that when a node inflates delays, it will appear to have an unusual location where few other legitimate nodes exist.

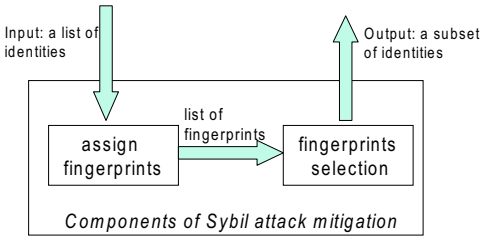


Fig. 5. Two components of our network location selection technique.

## III. Technique for Selecting Distinct Network Locations

In the previous section, we have presented two key properties of the Internet delay space and their sensitivity to artificial delay manipulation. In this section, we first state our assumptions about malicious nodes' capabilities on creating fake network locations and then present how delay space properties can help select distinct network locations.

### A. Assumptions About Malicious Nodes

We assume a malicious node has the following capabilities:

- It can possess an unlimited number of logical identities.
- It can hijack a large number of IP addresses and give each identity an unique IP address.
- It can respond with different fake network hops to pretend that the fake identities originate from different network locations, when a party attempts to traceroute to a fake identity.
- It can inflate the measured delay from a party to it arbitrarily by holding onto the probe message for an arbitrary amount of time. Note, however, that it is fundamentally impossible for a malicious node to reduce the measured delay.
- It knows everything about the employed network location selection strategy.

Our proposed technique simply leverages Internet delay space properties. It is effective even if a malicious node tries to game the system by inflating measured delays arbitrarily.

### B. Technique Overview

Our approach takes as input a list of identities, of which an arbitrary number could be originated from a malicious node, and then outputs a subset of carefully selected identities that are from as distinct locations as possible. Figure 5 illustrates the two key components in our approach: one component is used to assign a fingerprint to each identity and the other component is used for selecting a subset of identities based on their corresponding fingerprints. Note that once a fingerprint is obtained, it can be cached and reused later.

*1) Landmark Initialization:* When the system starts, a list of all landmarks and a list of random live IP addresses are input to each landmark. Please note that all the landmarks receive the same list of random live IP addresses and the list of live IP addresses is disjoint from IP addresses used by nodes in the distributed system. Each landmark then measures its delays to all the other landmarks and the provided random IP addresses. By probing other landmarks, a landmark will know which other landmarks are close to it. By measuring the delays

from itself to the list of random IP addresses, each landmark will get an empirical sample of the Internet delay space from its own point of view. We assume the random IP addresses do not behave maliciously and simply respond to ICMP pings. Landmarks may share their measured delays with each other if necessary. In our algorithm, one landmark will request the measured delays from a number of other closest landmarks. Each landmark may periodically restart the probings to update the measurements. We will explain how this information is used in our technique in the following.

*2) Assigning Fingerprints to Identities:* Assume $N$ landmarks $(L_1, L_2, ...., L_N)$ exist in the system. The following steps are used to generate a fingerprint for an identity $i$.

• **Step 1:** All landmarks will probe the identity $i$ to determine the closest landmark $L_k$ to $i$.

• **Step 2:** Landmark $L_k$ and its two closest landmarks $L_m$ and $L_n$ then generate a fingerprint $fp_i$ for identity $i$ in the format of $< (L_k, d_i^k), (L_m, d_i^m), (L_n, d_i^n) >$, where $d_i^k$ is the measured delay from landmark $L_k$ to identity $i$.

• **Step 3:** The confidence value $conf_i$ of the fingerprint $fp_i$ is computed by counting the number of legitimate fingerprints (corresponding to the random IP addresses provided to the landmarks in the initialization stage) that are within a certain confidence threshold $t$ to $fp_i$. The confidence calculation is the same as explained in Section II-B.

• **Step 4:** Since landmarks $L_k$, $L_m$ and $L_n$ know the delays among them and the delays from them to identity $i$, they can calculate whether $i$ causes a TIV together with the landmarks and record this using an indicator $tiv_i$, where $tiv_i = 1$ means that $i$ causes at least one TIV and $tiv_i = 0$ means that it does not cause any TIVs.

This component outputs the following information for identity $i$ to the identity-selection component: $<fp_i, tiv_i, conf_i>$.

*3) Selecting Identities Based on Their Fingerprints:* Given a set of fingerprints and their corresponding TIV measure and confidence measure, this component will select a subset of fingerprints using the following rules. The reasoning behind these rules are explained in Section III-C.

• **Rule 1:** If an identity causes TIVs, then we reject it.

• **Rule 2:** If the delay from an identity to its closest landmark is larger than a certain associativity threshold $T$, then we reject this identity because it is unacceptably far from its closest landmark. We now can classify the remaining fingerprints into different clusters based on their closest landmark. That is, fingerprints that have the same closest landmark will be classified into the same cluster. Then we can select fingerprints separately from each cluster.

• **Rule 3:** Within each cluster, we will first select fingerprints with the highest confidence measure because a fingerprint with a high confidence is considered to come from a realistic network location and is unlikely to have been manipulated. After selecting the fingerprint with the highest confidence, we will eliminate all other fingerprints from this cluster whose distances to the chosen fingerprint are smaller than the confidence threshold $t$. This is because we consider such similar fingerprints as originating from the same network location.

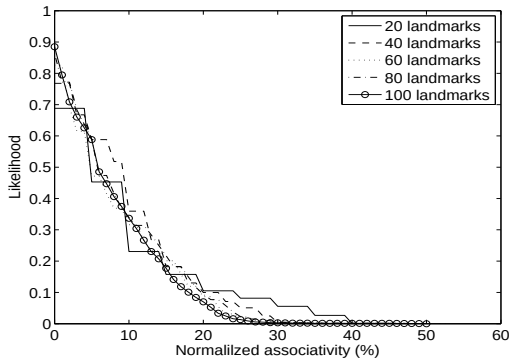


Fig. 6. Associativity of network nodes with different number of landmarks.

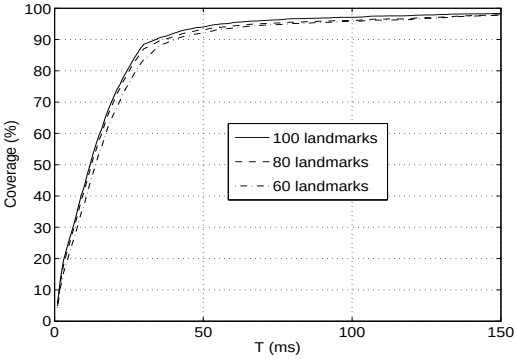


Fig. 7. Coverage rate of network nodes.

By using the above rules, we can select identities from the clusters in a round-robin fashion until all clusters have no more identities left. In summary, we use four techniques to select identities from distinct network locations even when an adversary is trying to mis-guide the selection process. 1) classify each identity to a local cluster, 2) favor identities not causing TIV, 3) favor identities with higher confidence, and 4) do not accept identities with similar fingerprint. The first three techniques are direct applications of the Internet delay space properties.

### *C. Exploiting Delay Space Properties for Network Locations Selection*

In this section, we explain in detail how the above techniques effectively exploit the properties of the delay space.

• **Exploiting Property 1:** Each identity is first associated with its closest landmark given a certain associativity threshold $T$. Identities associated with a particular landmark form a cluster.

A legitimate identity will simply be associated with its closest landmarks within delay $T$. A malicious node, however, will try to associate the identities it generates with as many landmarks as possible by manipulating the delays to make different identities appear closest to different landmarks.

However, property 1 guarantees that identities from a malicious node can only be present in a small fraction of clusters because the malicious node has limited associativity. Thus, if identities are chosen among clusters in a round-robin fashion,

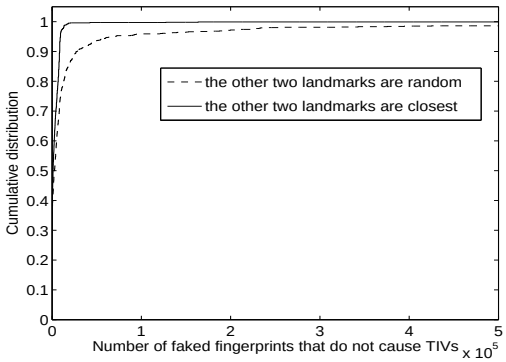


Fig. 8. Cumulative distribution of number of fake fingerprints that do not cause TIVs for all possible network locations in our data set.

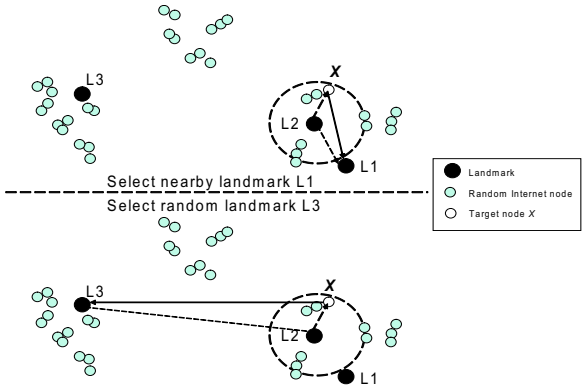


Fig. 9. Illustration of using nearby landmarks to assign fingerprint.

the fraction of identities from the same network location can be bounded by the associativity of the malicious node.

Figure 6 shows that when the number of landmarks changes, the normalized associativity of network nodes is quite stable (associativity threshold is 30ms). In other words, it is not very sensitive to the number of landmarks used. Moreover, the likelihood that a node can be associated with a large fraction of landmarks remain small. When 100 landmarks are used, nearly no node can be associated with more than 30% of the landmarks. Thus, a malicious node and all fake identities it creates cannot be associated with more than 30% of the landmarks.

Obviously, the associativity of a node varies with the associativity threshold $T$. Referring to Figure 1 again, it shows that as $T$ increases, a node (and thus all fake identities it may create) can associate with a larger fraction of landmarks. Thus, the goal is to choose a small enough $T$ such that a node cannot be associated with many landmarks, and at the same time most legitimate nodes can associate with at least one landmark.

Figure 7 shows how many nodes can be associated to at least one landmark with varying number of landmarks and associativity threshold $T$. As can be seen, given a reasonable threshold $T$, e.g., 30 ms, about 90% of nodes can be associated to a landmark. We can also see that a small fraction of nodes cannot be associated with any landmark even if we use a relatively large $T$. If these nodes simply have very large last hop delays (e.g. dial-up modem), we can use the delays to their last hop routers to perform the association instead (see discussion in Section V).

In summary, if we use a reasonably small $T$, e.g., 30 ms, the vast majority of legitimate nodes can be associated with their closest landmarks while a malicious node can only manage to associate its fake identities with a small fraction of landmarks.

• **Exploiting Property 2.1:** Property 2.1 states that TIVs happen less often among nearby nodes and if a node inflates delays to nearby nodes, it is likely to cause unusual TIVs. This property explains why we need to use a node $i$'s closest landmark $L_k$ and $L_k$'s two closest landmarks $L_m$ and $L_n$ to generate a fingerprint for node $i$. By using nearby landmarks to generate fingerprints we can reduce the number of legitimate nodes that are falsely rejected by applying Rule 1 and have a better chance to detect the manipulated delays by a malicious node. Figure 9 uses a simple example to demonstrate this property. In Figure 9, node $X$'s closest landmark is $L_2$. If we use a nearby landmark $L_1$ together with $L_2$ to generate a fingerprint for $X$, then $X$ cannot inflate its delay to $L_1$ too much because the other two edges $L_1L_2$ and $L_2X$ are already short. On the other hand, if the landmark $L_3$ is used to generate a fingerprint for node $i$, then because the edge $L_3L_2$ is relatively long, a malicious node can inflate the edge $L_3X$ a lot without causing a TIV.

Experiments show that if nearby landmarks are used to generate a fingerprint for each legitimate node, 16.2% of legitimate nodes will be falsely rejected because of them causing TIVs. In contrast, if random landmarks are used to generate fingerprints for legitimate nodes, 33.9% of legitimate nodes will be falsely rejected. Therefore using nearby landmarks can greatly reduce the negative impact on legitimate nodes. In addition, using nearby landmarks to generate fingerprints also helps to limit the total number of possible fake fingerprints. We generate fake fingerprints for each node in our data set in this way: given a node $i$ and three landmarks including its closest landmark, we generate all possible fingerprints by inflating the delay to its closest landmark in 1 ms increments, up to the associativity threshold $T$ and inflating its delays to other two landmarks in 1 ms increments until the inflated delays cause TIVs. Figure 8 compares the number of possible fake fingerprints by using nearby landmarks and using random landmarks. The result shows that if random landmarks are used, a malicious node can generate a lot more fake fingerprints without causing TIVs compared with using nearby landmarks.

• **Exploiting Property 2.2:** Property 2.2 states that when a node inflates delays heavily, it will appear to be at an unusual location. Our technique uses the confidence measure of a node's fingerprint to measure whether it resides at a realistic location. In order to compare the confidence values of legitimate fingerprints and fake fingerprints, we first calculate the confidence values for all legitimate fingerprints in our data. Then we calculate the confidence values of all possible fake fingerprints. Note that we use nearby landmarks to generate fingerprints. Figure 10 compares the confidence values of legitimate fingerprints and fake fingerprints. It shows that

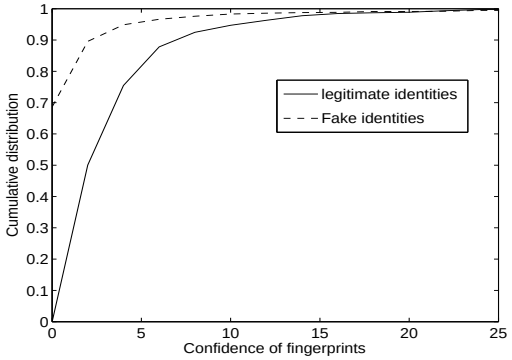


Fig. 10. Compare confidence of legitimate fingerprints and fake fingerprints.

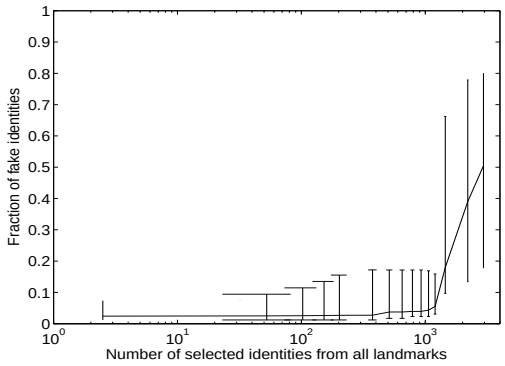


Fig. 11. Fraction of fake identities from all landmarks.

fake fingerprints have much lower confidence measure than legitimate fingerprints.

## IV. Evaluation

In this section, we simulate the scenario where a malicious node takes over a network location and tries to fake multiple network locations, which is observed by [15] [31] in Maze system and experimented in eMule by [27]. We evaluate the effectiveness of our technique for selecting nodes from distinct network locations when the malicious node is trying to fake all possible network locations by inflating delays to landmarks.

A *reference delay space* is needed by the landmarks for each experiment in this section. The reference delay space is composed of delays from all landmarks to a subset of the random IP addresses. Landmarks will use the reference delay space to calculate the confidence values for all identities. The remaining random IP addresses then can be used to simulate legitimate network locations in the system. Note that the IP addresses used in the reference delay space and the IP addresses used to represent legitimate network locations are disjoint. In this section, unless otherwise stated, the associativity threshold $T$ is 30 ms, the confidence threshold $t$ is 3 ms, the number of landmarks used is 100.

When one malicious node takes control of a network location, we assume it can fake multiple identities and then selectively inflate probing delays to make those fake identities associate with all the landmarks that are within $T$ delay to it instead of always associating with the true closest landmark. We name those landmarks that are within $T$ distance to the malicious node as *vulnerable landmarks* because they can be affected by the malicious node. The behavior of the malicious node is then: for each vulnerable landmark, the malicious node will generate as many fake identities as possible by inflating all possible delays in 1 ms increments to all corresponding landmarks to create fake fingerprints and then let those identities join the system. The number of fake fingerprints a malicious node can create varies according to the exact location of the malicious node, but on average it can create over 13 million identities associated with all possible vulnerable landmarks. We always let all legitimate identities join the system. Then the identity selection component is used to select a subset of identities out of all those legitimate identities and fake identities. We want to study how well we can select identities from truly distinct network locations using our technique.

### A. Basic Performance

In this section, we first fix the size of the reference delay space and the number of legitimate locations at both 3500 to show the basic performance of our technique. We let the malicious node take control of one network location in each experiment. The experiment is repeated for all possible network locations. The results presented here are accumulated over all possible network locations. If we select identities from all available landmarks in a round-robin fashion, Figure 11 shows the median fraction of accepted fake identities out of all accepted identities with 10% and 90% error bar. As can be seen, if the number of selected identities is below 1000, then the fraction of selected fake identities out of all selected identities in most cases is below 5% (i.e., less than 50 fake identities are selected). When we select more and more identities, legitimate identities will be exhausted sooner or later. When all the non-vulnerable landmarks are exhausted, the fraction of selected fake identities will increase sharply.

The above experiment has demonstrated the effectiveness of using the proposed technique to avoid fake identities form the same network location. The next natural question is how the performance of the technique will change with the increase of the number of legitimate locations. In our second experiment, we fix the size of the reference delay space at 1000 nodes, then we vary the number of legitimate locations from 1000 to 6000. Figure 12 shows the average fraction of selected fake identities. As can be seen, when we increase the number of legitimate locations, although the maximum number of legitimate locations (i.e., 6000) is still only about 0.05% of the total number of created fake identities (13 million), the performance of the technique is improved because more legitimate identities are competing with fake identities.

### B. Impact of Size of Reference Delay Space

In this experiment, we fix the number of the legitimate locations at 4000. Then we vary the size of reference delay space from 500 to 3000. Figure 13 shows the average fraction of selected fake identities. We can observe that generally the larger the reference delay space, the better the performance.

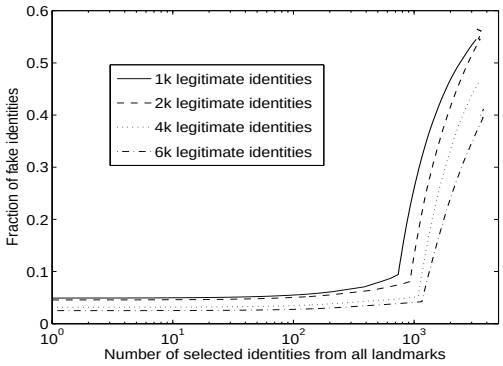


Fig. 12. Fraction of fake identities from all landmarks.

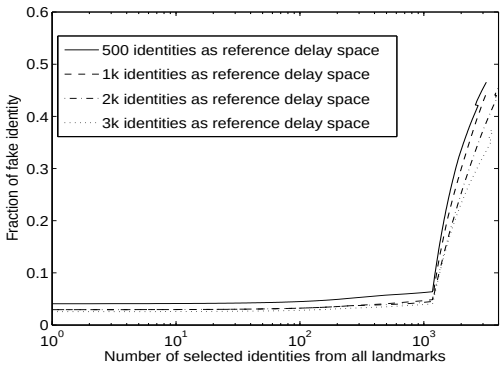


Fig. 13. Fraction of fake identities from all landmarks.

We can also observe that the benefit of increasing the size of the reference delay space diminishes. The performance of using a 1000-node reference delay space is very close to the performance of using a 3000-node reference delay space. This indicates that even if the landmark only probes 1000 IP addresses on the Internet, it still can provide good performance. Thus, the overhead of constructing a sufficient reference delay space is reasonably low.

### C. Performance Sensitivity to Parameters

Three configurable parameters are used in our technique: the number of landmarks, the associativity threshold $T$ and confidence threshold $t$. Due to the space limit, we will not be able to show any graph of the results, instead we will just briefly summarize our findings. All experiments in this section use 3500 nodes as the reference delay space and use the other 3500 nodes as the legitimate nodes in the system.

We first study the performance of our technique using different number of landmarks. We use $T = 30$ ms and $t = 3$ ms, then we vary the number of landmarks from 20 to 100. We found that when we only select a small number of identities, the performance of using different number of landmarks does not differ too much; but when we select more and more identities, the performance of using fewer landmarks becomes worse sooner. This is because by using fewer landmarks, we cover fewer legitimate identities. Thus, when we select more and more identities, the non-vulnerable landmarks are exhausted quicker when there are fewer landmarks. However, before the non-vulnerable landmarks are exhausted, the technique can effectively limit the fraction of selected fake identities even when a small number of landmarks (e.g., 20) are used.

Next, we study how the associativity threshold $T$ affects the performance of our technique. We use 100 landmarks and $t = 3$ ms, then vary $T$ from 10 ms to 50ms. We found that when a smaller $T$ is used, the fraction of fake identities selected also becomes smaller. This is because when a smaller $T$ is used, a malicious node can be associated with a smaller fraction of landmarks as shown in Figure 1. The problem of using a small $T$ is that many legitimate nodes will not be able to associate with any landmark. At $T = 10$ ms, 57% of nodes cannot associate with any landmark. By using a larger $T$, more nodes will be able to associate with some landmarks, but correspondingly the malicious node can also manage to associate with more landmarks, which is not desirable.

Finally, we study how the confidence threshold $t$ affects the performance of our technique. We use 100 landmarks and $T = 30$ ms, then we vary $t$ from 1 ms to 5 ms. The finding is that when a larger $t$ is used, the fraction of fake identities selected is smaller. The reason is that a larger $t$ will cause more fake identities to get eliminated after we select one identity.

## V. Discussion

Our technique relies on two delay space properties, then the first concern is: will those delay space properties be stable over time? First, it should be clear that because of the speed-of-light delay lower bound, Property 1 is always true. We argue that Property 2 should also remain true. Triangle inequality violation in the Internet delay space is caused by the routing policy of the Internet. While the routing policy may evolve over time, the amount of triangle inequality violation should not dramatically increase since ISPs have strong incentives to provide customers with low end-to-end delay. In addition, the distribution of nodes in the Internet is highly likely to remain very heterogeneous and clustered. Areas where few people live and where the ocean covers will most likely have very few nodes. Thus, a manipulated fingerprint is still likely to appear unusual for the foreseeable future.

The proposed technique also requires a node to be associated with a nearby landmark. One concern is whether the technique discriminates against nodes with big last-hop delays caused by the access network such as cable modem and DSL. Actually, for a node with a big last-hop delay, we may use its last-hop router to represent it. That is, we can assign the fingerprint of the node's last-hop router to it. A recent study [9] shows that last-hop routers of residential broadband nodes have much smaller delays to other nodes in the Internet.

Another issue is, in addition to network location diversity, some applications may also have application-specific requirements for peer selection in order to achieve better performance. For example, Pastry [26] employs proximity neighbor selection (PNS) strategy to reduce the average latency of overlay paths. Similarly in the proactive object replication application, peers with good connectivity such as high bandwidth are more preferred. We argue that our technique should be combined with

other peer selection requirements. Specifically, our technique is first used to select a list of peers with distinct locations, other techniques should be then applied to do further selection. Basically, we need strike a balance between the network location diversity and performance.

Finally, botnets [7] [22] are a serious threat to the security of P2P systems. What if an attacker compromised multiple nodes from diverse network locations? Fortunately, many effective botnet defenses (e.g., [12], [16]) have been proposed and they should be used in conjunction with our technique.

## VI. Related Work

Checking the distinctness of IP addresses is the most prevalent technique to avoid multiple identities from the same network location nowadays (e.g., [11]). This approach assumes that one node has only one IP address. However, this assumption does not always hold in reality because a malicious node may steal multiple IP addresses from its own network or hijack [13] [5] [33] a large number of IP addresses from diverse network locations.

Another class of approach tries to associate each node with a physical location and then uses physical locations to differentiate nodes. To do so, the system may require each participating node to provide some private information (e.g., SSN or driver's license number) to a centralized authority (CA) who can verify its real physical location according to the provided information. This approach compromises users' privacy and anonymity, so it is not desirable in many scenarios. In addition, the required CA is not always available in a decentralized distributed system. Another idea is to use the publicly available database such as WHOIS database [3] and Quova [2] to map each IP address to a geographical location. However, these kind of database is not 100% accurate and a malicious attacker can always hijack IP prefixes from all over the world. On the other hand, Octant [30] tries to infer one IP address's physical location by probing it from a number of landmarks whose physical locations are known. However, Octant assumes the probed node is always honest, so it cannot handle delay inflation from malicious nodes.

Bazzi and Konjevad [6] propose to use network coordinates [20] [8] as location certificates of nodes to distinguish them. [6] uses a set of beacons to probe each node and then computes a coordinate for it. This technique relies on the assumption that the Internet delay space conforms to the metric properties (symmetry, definiteness, triangle inequality), so each node can be assigned a secure coordinate by measuring its distances to a set of beacons. However, the assumed metric properties do not hold for real Internet delays.

## VII. Conclusions

In this paper, we propose a novel approach that uses Internet delays as "fingerprints" to identify distinct network locations. Distributed systems can pick nodes with different "fingerprints" to select nodes from distinct network locations to enhance their performance and robustness. Even when an adversary is trying to mis-guide the selection process by manipulating network delays, we show that properties of the Internet delay space can be used to greatly limit a node's ability to fake its network location. Finally, our experiments show that the technique is effective under realistic settings.

## References


[1] eMule. http://www.emule-project.net/.
[2] Quova. http://www.quova.com/.
[3] WHOIS. http://www.whois.net/.
[4] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *SOSP*, 2001.
[5] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the internet. In *ACM SIGCOMM*, 2007.
[6] R. Bazzi and G. Konjevod. On the establishment of distinct identities in overlay networks. In *ACM PODC*, July 2005.
[7] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding, detecting, and disturbing botnets. In *First Workshop on Steps to Reducing Unwanted Traffic on the Internet*, 2005.
[8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *ACM SIGCOMM*, August 2004.
[9] M. Dischinger, A. Haeberlen, K. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *ACM IMC*, October 2007.
[10] John Douceur. The sybil attack. In *IPTPS*, July 2002.
[11] M. J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *ACM CCS*, November 2002.
[12] Honeynet project and research alliance. know your enemy: Tracking botnets. http://www.honeynet.org/papers/bots/.
[13] Xin Hu and Morley Mao. Accurate real-time identification of ip prefix hijacking. In *IEEE Symposium on Security and Privacy*, May 2007.
[14] S. Lee, Z. Zhang, S. Sahu, and D. Saha. On suitability of euclidean embedding of internet hosts. In *ACM SIGMETRICS*, June 2006.
[15] Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, and X. Li. An Empirical Study of Collusion Behavior in the Maze P2P File-sharing System. In *IEEE ICDCS*, 2007.
[16] X. Liu, X. Yang, and Y. Lu. To Filter or to Authorize: Network-Layer DoS Defense Against Multimillion-node Botnets. In *SIGCOMM*, 2008.
[17] E. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft. On the accuracy of embeddings for internet coordinate systems. In *IMC*, 2005.
[18] Cristian Lumezanu, Dave Levin, and Neil Spring. Peerwise discovery and negotiation of faster paths. In *ACM HotNets*, November 2007.
[19] Napster. http://www.napster.com/.
[20] T. S. E. Ng and H. Zhang. Predicting Internet networking distance with coordinates-based approaches. In *IEEE INFOCOM*, June 2002.
[21] PlanetLab. http://www.planet-lab.org.
[22] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *ACM IMC*, October 2006.
[23] V. Ramasubramanian and E. G. Sirer. Beehive: O(1) Lookup Performance for Power-law Query Distributions in Peer-to-peer Overlay. In *USENIX NSDI*, 2004.
[24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *ACM SIGCOMM*, 2001.
[25] Route views. http://www.routeviews.org/.
[26] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM Middleware*, 2001.
[27] M. Steiner, T. En-Najjary, and E. Biersack. Exploiting KAD: Possible Uses and Misuses. In *ACM SIGCOMM CCR*, 2007.
[28] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *ACM SIGCOMM*, 2001.
[29] G. Wang, B. Zhang, and T. S. E. Ng. Towards network triangle inequality violation aware distributed systems. In *IMC*, 2007.
[30] B. Wong, I. Stoyanov, and E. Sirer. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *USENIX NSDI*, 2007.
[31] M. Yang, Z. Zhang, X. Li, and Y. Dai. An Empirical Study of Free-riding Behavior in the Maze P2P File-sharing System. In *IPTPS*, 2005.
[32] B. Zhang, T. S. E. Ng, A. Nandi, R. Riedi, P. Druschel, and G. Wang. Measurement-based analysis, modeling, and synthesis of the internet delay space. In *ACM IMC*, October 2006.
[33] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis. A light-weight distributed scheme for detecting ip prefix hijacks in realtime. In *SIGCOMM*, 2007.