On Inferring the Geographic Properties of the Internet

Lakshminarayanan Subramanian
University of California at Berkeley
lakme@cs.berkeley.edu

Acknowledgments

This M.S. thesis is a collection of two pieces of work done under the guidance of Dr. Venkata N. Padmanabhan at Microsoft Research and Prof. Randy H. Katz. The first piece of work is on *IP-Geography Mapping*. I worked on this problem as an intern at Microsoft Research under the mentorship of Venkata Padmanabhan. Later, we studied the *Geographic Properties of Internet Routing*. The valuable comments of the anonymous reviewers of the papers based on these works have been incorporated in this thesis. In addition, Vern Paxson shepherded our USENIX paper and helped us largely improve the presentation. Prof. Randy H. Katz, Dr. Venkata N. Padmanabhan and Prof. Ion Stoica read an earlier draft of this thesis report and provided valuable feedback.

Arvind Arasu, B. R. Badrinath, Mary Baker, Paul Barford, John Byers, Imrich Chlamtac, Mike Dahlin, Kevin Jeffay, Craig Labovitz, Paul Leyland, Karthik Mahesh, Vijay Parthasarathy, Jerry Prince, Amin Vahdat, Srinivasan Venkatachary, Geoff Voelker, Marcel Waldvogel, and David Wood helped us obtain access to a geographically distributed set of measurement hosts used in both these studies.

For our *IP-Geography Mapping* work, George Chung, Alex Daunys, Cheng Ku, and Dave Quick helped us obtain the Hotmail, bCentral, and FooTV data sets. Craig Labovitz helped us gather BGP data. Donald Wysocki provided us with geographic data for U.S. cities and zip codes. Vern Paxson made his 1995 traceroute data set available to us for our study on *Geographic Properties of Internet Routing*. I would like to thank them all.

Contents

Chapter 1

Introduction

The Internet is a vast and complex network formed as a conglomerate of thousands of smaller networks owned by separate administrative entities. These smaller networks are referred to as either *autonomous systems* (ASes). An Internet Service Provider (ISP) can either be a collection of one or more ASes connected to each other and owned by a single administrative entity or can be a small access provider depending on other ASes for advertising its routes. Autonomous systems vary both in network size and geographic spread. At one end, we have tier-1 ISPs and global transit providers whose networks spread across continents, while at the other end we have customer ISPs whose spread is restricted to a very small geographic region (like a university campus). In the middle, we have regional and national providers who have points of presence spanning an entire country or a relatively large geographic region within a country.

There have been many studies which analyze different properties of the Internet. From a networking perspective, a large chunk of these studies have analyzed either the performance aspects of the Internet or its underlying network topology structure. Given the large and diverse geographic spread of the Internet, very few studies have quantified or measured the geographic properties behind this complex structure. This can be attributed to two reasons:

- *Geographic information is hard to obtain.* Addresses used for identifying end-nodes in this network (IP address) does not inherently contain an indication of its geographic location. The network topology structure of an ISP provides little information of its geographic spread.
- A common perception in the networking community is that geography has little relationship to performance. Since performance has been a driving force behind many of these studies, geography has not been an important topic of study in this context.

Geography, however, has been integral part of many *location-based* services like the Global Positioning System (GPS). The GPS technology is widely used for object-tracking and navigational purposes. Object tracking systems are very common in many military applications and navigation is an essential component in many transportation systems like ships, automobiles and planes. In the context of the Internet, *location awareness* has become increasingly important. Many Internet services as of today would benefit by knowing the geographic locations of the end-hosts (clients). However, only recently have there been efforts directed towards building a GPS-like mapping service for the Internet. In our work, we investigate different geographic mapping techniques for Internet hosts and study the trade-offs for these different approaches. We extend our work by studying the use of geography as a tool for quantifying different properties of the Internet. In particular, we show how to

use geographic information to infer certain properties of Internet routing like circuitous routing and routing policies of ISPs like hot-potato routing. Many of these properties are not characterizable using purely network-centric metrics. We also examine the fault tolerance of the topologies of many ISPs from a geographic perspective.

1.1 Overview of our Work

We classify our work on inferring geographic properties of the Internet into three categories. They are:

- **IP-location mapping:** Given the IP address of an Internet host, can we determine its geographic location?
- **Geographic properties of routing:** Given the geographic information of Internet routes, can we infer certain properties which are not quantifiable using purely network-centric metrics?
- **Geographic fault tolerance:** How would we characterize fault tolerance of network topologies in the presence of infrastructure failures in a geographic region? (Multiple nodes and links which are geographically co-located will simultaneously fail)

We will now briefly describe an overview of these individual pieces.

1.1.1 IP-location mapping

Building an IP address to location mapping service (the *location mapping* problem for short) is an interesting problem in its own right. Such a service would also enable a large and interesting class of location-aware applications for Internet hosts, just as systems such as GPS [9] have for mobile devices. By knowing the location of a client host, an application, such as a Web service, could send the user location-based targeted information on local events, regional weather, etc. (*targeted advertising*), classify users based on location (e.g., count "hits" based on the region the user is located in), or control the availability of data based on user location (*territorial rights management* akin to TV broadcast rights). Each application may have a different requirement on the resolution of location information needed.

We present several novel techniques, collectively referred to as *IP2Geo* [24], that approach the location mapping problem from different angles. These techniques exploit various properties of and observations on the Internet such as hierarchical addressing and correlation between delay and distance. We have analyzed a variety of data sets both to refine these techniques and evaluate their performance. To the best of our knowledge, ours is the first research effort in the open literature that studies this problem in detail.

Our first technique, *GeoTrack*, tries to infer location based on the DNS names of the target host or other nearby network nodes. The DNS name of an Internet host sometimes contains clues about the host's location. Such a clue, when present, could indicate location at different levels of granularity such as city (e.g., *corerouter1.SanFrancisco.cw.net* indicates the city of San Francisco), state (e.g., *www.state.ca.us* indicates the state of California), or country (e.g., *www.un.cm* indicates the country of Cameroon).

Our second technique, *GeoPing*, uses network delay measurements made from geographically distributed locations to infer the coordinates of the target host. It is based on the premise that the delay experienced by packets traveling between a pair of hosts in the network is, to first order, a function

of the geographic separation between the hosts (akin to the relationship between signal strength and distance exploited by wireless user positioning systems such as RADAR[2]). This is, of course, only an approximation. So our delay-based technique relies heavily on empirical measurements of network delay, as discussed in Section 3.3.

Our third technique, *GeoCluster*, combines partial (and possibly inaccurate) IP-to-location mapping information with BGP prefix information to infer the location of the host of interest. For our research, we obtained the host-to-location mapping information from a variety of sources, including a popular Web-based email site, a business Web hosting site, and an on-line TV guide site. The data thus obtained is *partial* in the sense that it only includes a relatively small number of IP addresses. We use BGP prefix information to expand the coverage of this data by identifying clusters of IP addresses that are likely to be located in the same geographic area. This technique is self-calibrating in that it can offer an indication of how accurate a specific location estimate is likely to be.

We have evaluated these techniques using extensive and varied data sets. While none of the techniques is perfect, their performance is encouraging. The median error in our location estimate varies from 28 km to several hundred kilometers depending on the technique used and the nature of the hosts being located (e.g., well-connected clients versus proxy clients). This precision is very good for applications like targeted advertisements.

1.1.2 Geographic Properties of Routing

Routing across ASes is accomplished using the Border Gateway Protocol (BGP), a protocol for propagating routes between ASes. The network path between two end-hosts typically traverses multiple ASes. BGP is flexible in allowing each AS to apply its own local preferences, and export and import policies for route selection and propagation. The characteristics of an end-to-end path are very much dependent on the policies employed by the intervening ASes.

Previous work on Internet routing has focused on studying properties such as end-to-end performance, routing stability, and routing convergence that are affected by routing policies. We present a different way of analyzing certain properties of Internet routing. We show how *geographic* information can provide insights into the structure and functioning of the Internet, including the interactions between different autonomous systems [32]. In particular, geographic information can be used to quantify well-known network properties such as hot-potato routing. It can also be used to quantify and substantiate prevalent intuitions about Internet routing, such as the relative optimality of intra-ISP routing compared to inter-ISP routing.

To analyze geographic properties of routing, it is necessary to first determine the *geographic* path of an IP route. The geographic path is obtained by stringing together the geographic locations of the nodes (i.e., routers) along the network path between two hosts. For instance, the geographic path from a host in Berkeley to one in Harvard may look as follows: Berkeley → San Francisco → New York → Boston → Cambridge. The level of detail in the geographic path would depend on how precisely we are able to determine the locations of the intermediate routers in the path. We use GeoTrack [24], a tool we have developed for determining the geographic path of routes. Our study is based on extensive traceroute data gathered from 20 hosts distributed across the U.S. and Europe and also traceroute data gathered by Paxson [53] in 1995.

Internet routes can be highly circuitous [26]. For instance, we observed a route from a host in St. Louis to one in Indiana (328 km away) that traverses a total distance of over 3500 km (Section 4.1.2). By tracing the geographic path, we are able to automatically flag such anomalous routes, which would be difficult to do using purely network-centric information such as delay. We compute the *linearized distance* between two hosts as the sum of the geographic lengths of the individual links of

the path. We then compute the ratio of the linearized distance of the path to the geographic distance between the source and destination hosts, which we term the *distance ratio*. A large ratio would be indicative of a circuitous and possibly anomalous route. In Section 4.1, we study circuitousness of paths as a function of the geographic and network locations of the end-hosts.

Our results indicate that the presence of multiple ISPs in a path is an important contributor to circuitous routing. We also find intra-ISP routing to be far less circuitous than inter-ISP routing. Our study of circuitousness of paths provides some insights into the peering and routing policies of ISPs. Although circuitousness may not always relate to performance, it can often be indicative of a routing problem that deserves more careful examination.

There are two extremes to the routing policy that an ISP may employ: *hot-potato* routing and *cold-potato* routing. In hot-potato routing, the ISP hands off packets to the next ISP as quickly as possible. In cold-potato routing, the ISP carries packets on its own network as far as possible before handing them off to the next ISP. The former policy minimizes the burden on the ISP's network whereas the latter gives the ISP greater control over the end-to-end quality of service experienced by the packets. As we discuss in Section 4.2.4, geographic information provides a means to quantify these notions by using the geographic distance traversed within an ISP as a proxy for the amount of work performed by the ISP. In addition, we can also evaluate the degree to which an individual ISP contributes in the routing of packets end-to-end. Our analysis of properties of paths that traverse multiple ISPs is presented in Section 4.2.

1.1.3 Geographic Fault Tolerance

Another aspect of routing that bears careful examination is its fault tolerance. Fault tolerance has generally been studied in the context of node or link failures based on network-level topology information. However, such topology information may be incomplete in that two seemingly independent nodes may actually be susceptible to correlated failures. For instance, a catastrophic event such as an earthquake or a major power outage might knock out all of an ISP's routers in a geographic region. Geographic information can help in identifying routers that are co-located. In order to analyze the impact of correlated failures, we consider ISP topologies at the geographic level, where each node represents a geographic region such as a city. Using the geographic topology information of several commercial ISPs gathered from CAIDA [49], we analyze the fault tolerance properties of individual topologies and the topology resulting from the combination of the individual ISP networks [32]. We find that many tier-1 ISPs have highly skewed degree distributions which may make them highly susceptible to single geographic node failures. The combined topology of these ISPs however seems to exhibit better tolerance to such failures.

1.1.4 Summary

In summary, we believe geography is an interesting means for analyzing and quantifying network properties. We believe that a significant contribution of our *IP2Geo* work is a systematic study of a broad spectrum of techniques and a discussion of the fundamental challenges in determining location based just on the IP address of a host. Our analysis of geographic fault tolerance of routing provides additional evidence for existing intuition about certain properties of Internet routing (e.g., hot-potato routing, circuitous paths). An important contribution of this work is a methodology for quantifying such intuitions using geographic information. Such quantification enables us, for instance, to automatically flag circuitous paths, something that would be hard to using purely network-centric metrics (and no geographic information). Finally, our analysis of the topological structure of ISPs reveal that certain tier-1 ISPs may have very low tolerance to even single node

failures.

1.2 Current State of the Art

In this section, we will describe related work to different aspects of our work. We classify related work into three categories: (a) Location mapping services (b) Internet routing (c) Topology discovery and mapping.

1.2.1 Location Mapping

There has been been much work on the problem of locating hosts in wireless environments. The most well-known among these is the Global Positioning System (GPS) [9]. However, GPS is ineffective indoors. There have been several systems targeted specifically at indoor environments, including Active Badge [15], Bat [16], and RADAR [2]. As we discuss later, our GeoPing technique uses a variant of one of the algorithms we had developed for RADAR. However, in general these techniques are specific to wireless networks and do not readily extend to the Internet.

In the Internet context, an approach that has been used to determine location is to seek the user's input (e.g., by requiring the user to register with and/or log in to the site, by storing the user's credentials in client-based cookies, etc.). However, such approaches are likely to be (a) burdensome on the user, (b) ineffective if the user uses a client other that the one where the cookie is stored, and (c) prone to errors due to (possibly deliberate) inaccuracies in the location information provided by an *individual* user. (In Section 3.4, we discuss how GeoCluster deals with such inaccuracies by aggregating information derived from individual users.)

An alternative approach is to build a service that maps an IP address to the corresponding geographic location [28]. There are several ways of doing this:

1. Incorporating location information (e.g., latitude and longitude) in Domain Name System (DNS) records.
2. Using the *Whois* [14] database to determine the location of the organization to which an IP address was assigned.
3. Using the *traceroute* [17] tool and mapping the router names in the path to geographic locations.
4. Doing an exhaustive tabulation IP address ranges and their corresponding locations.

The DNS-based approach was proposed in RFC 1876 [33]. This work defines the format of a new Resource Record (RR) for the DNS, and reserves a corresponding DNS type mnemonic (LOC) and numerical code (29). The DNS-based approach faces deployment hurdles since it requires a modification of the record structure of the DNS records. This also burdens administrators with the task of entering the LOC records. Moreover, there is no easy way of verifying the accuracy of the location entered.

An approach used widely in many tools is to query Whois servers [14]. Tools such as IP2LL [44] and NetGeo [22] use the location information recorded in the Whois database to infer the geographic location of a host.

There are several problems with Whois-based approaches. First, the information recorded in the Whois database may be inaccurate or stale. Also, there may be inconsistencies between multiple

servers that contain records corresponding to an IP address block. Second, a large (and geographically dispersed) block of IP addresses may be allocated to a single entity and the Whois database may contain just a single entry for the entire block. For example, the 4.0.0.0/8 IP address block is allocated to BBN Planet (now known as Genuity) and a query to ARIN Whois database returns the location as Cambridge, MA for any IP address within this range.

An alternative approach is based on the traceroute tool. The basic idea here is to perform a traceroute from a source to the target IP address and infer location information from the DNS names of routers along the path. A router name may not always contain location information. Even when it does, it is often challenging to identify the location information since there is no standard naming convention that is used by all ISPs. We discuss these issues in more detail when we present GeoTrack in Section 3.2. Examples of location mapping tools based on traceroute include VisualRoute [58], Neotrace [52], and GTrace [27].

Finally, there are location mapping services, such as EdgeScape from Akamai [34] and TraceWare from Digital Island [38]. Given the extensive relationship that these large content distribution networks enjoy with several ISPs, it is conceivable that these location mapping services are based on an exhaustive tabulation of IP address ranges and the corresponding location. However, the algorithms employed by EdgeScape and TraceWare are proprietary, so it is difficult for us to compare them to our research effort.

1.2.2 Internet routing

There are several properties of Internet routing that are of interest: end-to-end performance, routing stability, routing convergence, etc. Previous work on Internet routing has focused either on measuring these properties or on modifying certain aspects of routing with a view to improving performance. Our work shows how geographic information can be used to measure and quantify certain routing properties such as circuitous routing, hot-potato routing and geographic fault tolerance.

Network path information, obtained using the *traceroute* tool [17], has been used widely to study the dynamics of Internet routing. For instance, Paxson [26] studied various aspects of Internet routing using an extensive set of traceroute data. They include: routing pathologies, stability of routing, and routing asymmetry. In relation to our work, he studies circuitous routing by determining the geographic locations of the routers in his dataset and uses geographic distance as a metric to quantify it. In addition, he uses the number of different geographic locations along a path to analyze the effect of hot-potato routing as a potential cause for routing asymmetry. We extend this work by studying circuitousness as a function of the geographic and network location of end-hosts. We also analyze the effects of multiple ISPs in a path on its circuitousness. The distance ratio metric that we define can be used to automatically flag anomalies such as the large-scale route fluttering identified in [19, 26].

Overlay routing has been proposed as a means to circumvent the default IP routing. Savage et al. [31] study the effects of the routing protocol and its policies on the end-to-end performance as seen by the end-hosts. They show that for a large number of paths in the Internet, there exist paths that exhibit significantly better performance in terms of latency and packet loss rate. Recently, Andersen et al. [1] have proposed specific mechanisms for finding alternate paths with better performance characteristics using an overlay network. By actively monitoring the quality of different paths, their alternate path selection mechanism can quickly recover from network failures and optimize application specific performance metrics.

Consistent with these findings, our measurements indicate the existence of highly circuitous paths in the Internet. We also find that the circuitousness of a path is correlated with the minimum end-to-end

latency along the path.

1.2.3 Topology discovery and mapping

Discovering and analyzing Internet structure has been the subject of many studies. Much of the work has focused on studying topology purely at the network level, without any regard to geography. Recently several tools have been developed to map network nodes to their corresponding geographic locations. A few Internet mapping projects have used such tools to incorporate some notion of geographic location in their maps.

The Mercator project [12] focuses on heuristics for Internet Map Discovery. The basic approach is to use traceroute-like TTL limited probe packets coupled with source routing to discover routers. A key component of Mercator is the set of heuristics used to resolve *aliases*, i.e., multiple IP addresses corresponding to (possibly different interfaces on) a single router. The basic idea is to send a UDP packet to a non-existent port on a router and wait for the ICMP *port unreachable* response that it elicits. In general, the destination IP address of the UDP packet and the source IP address of the ICMP response may not match, indicating that the two addresses correspond to different interfaces on the same router. In our work we use geographic information to identify points of sharing in the network. We view this as complementary to network-level heuristics such as the ones employed in Mercator.

The Internet Mapping Project [5] at Bell Labs also uses a traceroute-based approach to map the Internet from a single source. The map is colored according to the octets of the IP address, so portions corresponding to the same ISP tend to be colored similarly. The map, however, is not laid out according to geography. Other efforts have produced topological maps that reflect the geography of the Internet. Examples include the MapNet [49] and Skitter [55] projects at CAIDA and the commercial Matrix.Net service [50].

1.3 Organization of the Report

In Chapter 2, we will present our experimental methodology used in our studies. We will describe our measurement testbed and the the datasets that we used for studying specific geographic properties. In Chapter 3, we will detail our list of IP-Geography mapping techniques and analyze the characteristics of each one of them. Using GeoTrack (one of the IP-Geography Mapping tools), we extract the geographic paths of Internet routes and analyze specific properties of Internet routing like hot-potato routing. We present our results in Chapter 4. Our analysis of geographic fault tolerance properties of ISP topologies is presented in Chapter 5. Finally, we present our conclusions in Chapter 6.

Chapter 2

Experimental Setup and methodology

In this chapter, we discuss our experimental setup and methodology. We present the details of our measurement test bed and the data sets we gathered for our studies on IP-location mapping and geographic properties of Internet routing. We will use *GeoRoute* to refer to our study on geographic properties of routing while we refer to our collection of location-mapping techniques as *IP2Geo*.

2.1 Measurement testbed

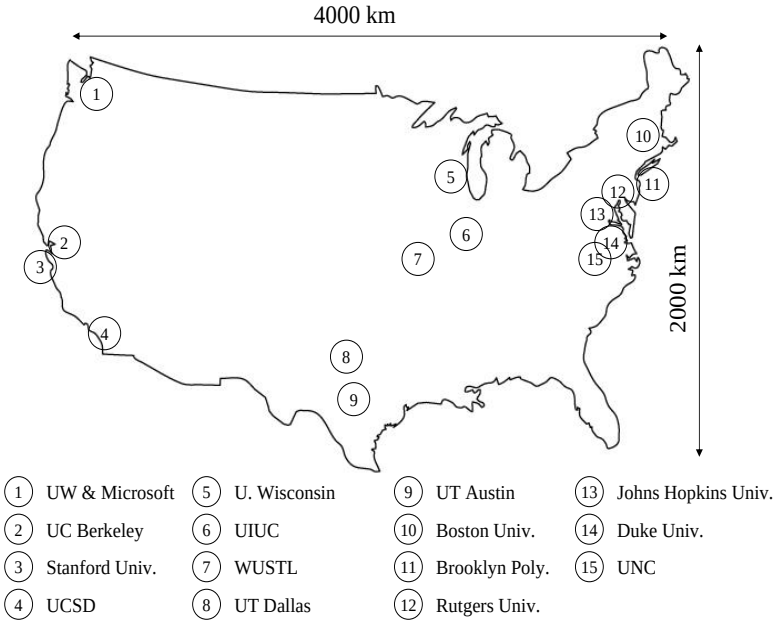

Figure 2.1: Locations of our probe machines in the U.S. Note that there were 17 hosts in 15 locations (two hosts each in Seattle and Berkeley).

The measurement testbed is mostly common for both the *IP2Geo* and *GeoRoute* studies. We used 20 geographically distributed hosts as the sources for our traceroutes. 17 of these hosts were located in the U.S. (Figure 2.1) while 3 were located in Europe (at Stockholm (Sweden), Bologna (Italy), and Budapest (Hungary)). The geographical diversity in source locations enables us to study the variations in routing properties as seen from different vantage points. For logistical reasons, it was convenient for us to locate the traceroute sources on university campuses. 18 out of the 20 traceroute

sources fell into this category. Furthermore, 9 of the 15 university locations we considered in the U.S. were connected by the Internet2 backbone [42]. To add some diversity, we had one source in Berkeley, CA connected to a home cable modem network (in addition to a host at the University of California at Berkeley) and another in Seattle, WA connected to the Microsoft Research network (in addition to a host at the University of Washington at Seattle). These two pairs of sources allow us to study what impact, if any, the nature of the source's connectivity has on Internet routing.

Our analysis of different techniques in IP2Geo was restricted to the U.S. The main reason for this restriction is that, as of the time of this writing, the bulk of the data sets and probe machines that we have pertain to or are located in the U.S. While there may be limitations to studying a single country, the U.S. still offers a large and varied testbed for our research. We use 14 of these probe sources in different geographic locations to investigate some of the techniques in IP2Geo. These probe machines were used to make delay measurements for GeoPing and to initiate traceroutes for GeoTrack. In GeoRoute, we use these probe machines to initiate traceroutes to a large variety of destination end-hosts as we discuss next in Section 2.2.1.

2.2 GeoRoute:Methodology

Since the goal of our work is to study the geographic properties of Internet routing, much of our measurement work has focused on gathering network path data using the traceroute tool [17]. We are not interested in studying the dynamic properties of Internet routing (e.g., how routes change over time), so we only record a single snapshot of the network path between a given pair of hosts. It may possible that some of the routes in our dataset are backup paths due to failures at the time of our measurement. However, we do not expect the aggregate statistics reported in this paper to be affected by such failures since our measurements were spread over a $2-$month time period. We use traceroute to determine the network path between 20 traceroute sources and thousands of geographically distributed destination hosts.

Once we have gathered the traceroute data, we use the GeoTrack tool to determine the location of the nodes along each network path where possible. GeoTrack reports the location at the granularity of a city. We will discuss GeoTrack in more detail in Section 3.2. We then use an on-line latitude-longitude server [40] to compute the geographic distance between the source and destination of a traceroute as well as between each pair of adjacent routers along the path. The latter enables us to compute the *linearized distance*, which we define as the sum of the geographic distances between successive pairs of routers along the path. So if the path between A and D passes through B and C, then the linearized distance of the path from A to D is the sum of the geographic distances between A & B, B & C, and C & D.

As we discuss in Section 3.2.2, we are typically able to determine the location of most but not all routers. We simply skip the routers whose locations we are unable to determine. So in the above example, if the location of C is unknown, then we compute the linearized distance of the path from A to D as the sum of the geographic distances between A & B and B & D. Clearly, skipping over C would lead us to underestimate the linearized distance. However, as noted in Section 3.2.2, most of the skipped nodes are in the vicinity of the either the source or the destination, so the error introduced in the linearized distance computation is small.

For studying properties of Internet routing, it is necessary to a large dataset of Internet routes to draw any reasonable conclusion. Also, one requires diversity in the set of destination end-hosts. For GeoRoute, we initiate traceroutes from the 20 geographically diverse hosts, to a set of destination hosts.

2.2.1 Destination End-hosts

We carefully chose the set of destination hosts to account for both network diversity and geographic diversity. The destination set for the traceroutes comprised several thousand hosts. These destinations hosts fell into 4 categories:

1. *UnivHosts:* 265 Web servers and other hosts located on university campuses in the U.S. The hosts were distributed across 44 of the 51 states in the U.S.
2. *LibWeb:* 1,205 Web servers of public libraries [46] distributed across 49 states in the U.S. We also ensured that the distribution of the geographic locations of these libraries is not skewed.
3. *TVHosts:* 3,100 client hosts in the U.S. that connected to an on-line TV program guide. A majority of these clients were located on non-academic networks such as America Online (AOL).
4. *EuroWeb:* 1,092 Web servers [48] distributed across 25 countries in Europe.

For ease of exposition, we sometimes refer to UnivHosts, LibWeb, and TVHosts as the U.S. hosts and EuroWeb as the European hosts.

This diverse set of destination hosts enables us to investigate the properties of Internet routing in the context of a large set of ISPs. In all, we traced approximately 84,000 end-to-end paths between our traceroute sources and the destination hosts during October-December 2000. Our data is available on-line at [54].

2.3 IP2Geo :Methodology

IP2Geo comprises of 3 techniques: GeoTrack, GeoPing and GeoCluster. As we explain later in Section 3.3, GeoPing is primed using a database of delay measurements from the probe machines to several target machines at known locations. To obtain such a database, we used the UnivHosts as the set of end hosts. The selection of university servers as target hosts offered the advantage that we were quite certain of their actual geographic location. The UnivHosts data set is also used to evaluate the performance of GeoTrack and GeoCluster. Other than this, GeoCluster also needs BGP data and partial location mapping information to determine the geographic location of an IP address. We will briefly describe these data sets below.

2.3.1 BGP Data

BGP routing information was derived from dumps taken at two routers at BBN Planet [36] and MERIT [51]. Since GeoCluster only requires the *address prefix (AP)* information, we constructed a superset containing address prefix information derived from both sources. In all there were 100,666 APs in our list.

2.3.2 Partial Location Mapping Information

We obtained partial IP-to-location mapping information from three sources. The data sets we obtained were partial in the sense that they only covered a small fraction of IP address space in use. Note that in no case did we have access to user IDs or other user-specific information. Our data sets only contained IP address and location information. So our work did not compromise user privacy in any way.

1. *Hotmail*: Hotmail [41] is a popular Web-based email service with several million active users. Of the over 1 million (anonymous) users for whom we obtained information, 417721 users had registered their location as being in the U.S. We restrict our analysis to this subset of users. The location information we obtained from the users' registration records was at the granularity of U.S. states. In addition, we obtained a log of the client IP addresses corresponding to the 10 most recent user logins (primarily in the first half of 2000). We combined the login and registration information to obtain a partial IP-to-location mapping.

2. *bCentral*: bCentral [37] is a business Web hosting site. Location information at the granularity of zip codes was derived from HTTP cookies. In all we obtained location information corresponding to 181246 unique IP addresses seen during (part of) a day in October 2000.

3. *FooTV*: FooTV is an on-line TV program guide where people look up program listings for specific zip codes. (We do not reveal the name of the site here due to anonymity requirements.) From traces gathered over a two-day period in February 2000, we obtained a list of 142807 unique client IP addresses and 336181 (IP,zip) pairs corresponding to the client IP address and the zip code that the user specified in his/her query. A subset of the IP addresses had more than one corresponding zip code, which were usually clustered together geographically.

In the case of bCentral and FooTV, we mapped the zip code information to the corresponding (approximate) latitude and longitude using information from the U.S. Census Bureau [56]. In the case of Hotmail, we computed the *zipcenter* of each state by averaging the coordinates of the zip codes contained within that state.

The partial IP-to-location mapping obtained from these sources may contain inaccuracies. For instance, in the case of Hotmail and bCentral users may have registered incorrect location information or may connect from locations other than the one they registered. In the case of FooTV, users may enquire about TV programs in areas far removed from their current location, although we believe this is unlikely. Regardless, we explain in Section 3.4 how GeoCluster is robust to such inaccuracies in location information.

Chapter 3

IP-Geography Mapping Techniques

In this chapter, we will present different techniques collectively referred to as *IP2Geo*, for determining the geographic location of Internet hosts. Such a service would enable a large and interesting class of location-aware applications. This is a challenging problem because an IP address does not inherently contain an indication of location.

We present and evaluate three distinct techniques in IP2Geo. The first technique, *GeoTrack*, infers location based on the DNS names of the target host or other nearby network nodes. The second technique, *GeoPing*, uses network delay measurements from geographically distributed locations to deduce the coordinates of the target host. The third technique, *GeoCluster*, combines partial (and possibly inaccurate) host-to-location mapping information and BGP prefix information to infer the location of the target host. Using extensive and varied data sets, we evaluate the performance of these techniques and identify fundamental challenges in deducing geographic location from the IP address of an Internet host.

3.1 Fundamental Limitation due to Proxies

Before we describe our techniques, we will illustrate a fundamental limitation imposed by proxies in solving the IP-location mapping problem. Many Web clients are behind proxies or firewalls. So the client IP address seen by the external network may actually correspond to a proxy, which may be problematic for location mapping. In some cases the client and the proxy may be in close proximity (e.g., a caching proxy on a university campus). However, in other cases they may be far apart. An example of the latter is the AOL network [35], which has a centralized cluster of proxies at one location (Virginia) for serving client hosts located all across the U.S. Figure 3.1 shows the cumulative distribution function (CDF) of the distance between the AOL proxies and clients. (The *likely* location of clients was inferred from the data sets described in Section 2.3.2.) We observe that a significant fraction of the clients are located several hundred to a few thousand kilometers from the proxies.

Proxies impose a fundamental limitation on all location mapping techniques that depend on client IP address. This includes techniques based on Whois, traceroute (e.g., GeoTrack), and network delay measurements (e.g., GeoPing). Not only are these schemes unable to determine the true location of a client, they are also oblivious to the error (i.e., these schemes would incorrectly return the location of the proxy without realizing the error). Our GeoCluster technique is an exception in that it is often able to automatically tell when its location estimate is likely to be erroneous. So rather than incorrectly deducing the location of the client based on the IP address of the proxy, GeoCluster

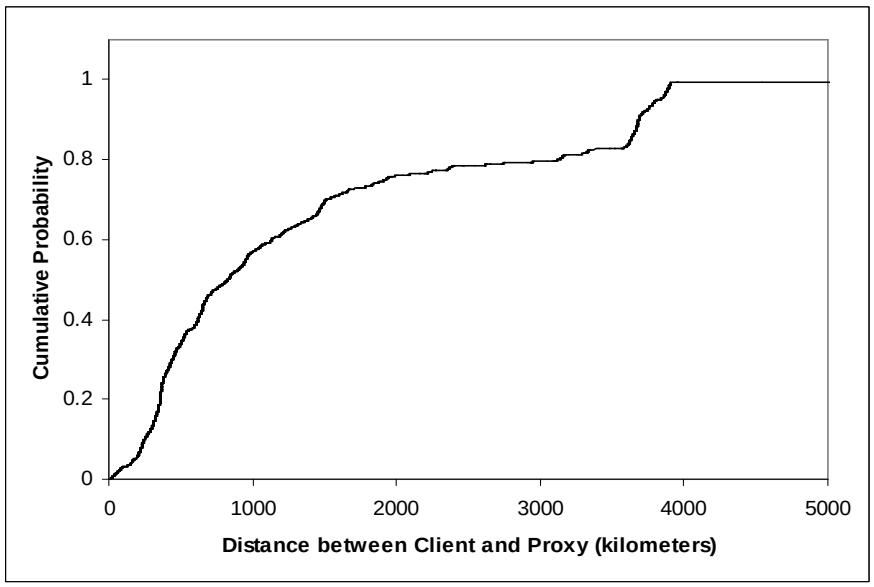

Figure 3.1: Distribution of distance between AOL proxies and clients.

would refrain from making a location estimate at all. We discuss this issue is more detail in Section 3.4.3.

3.2 The GeoTrack Technique

The GeoTrack technique tries to infer location based on the DNS names of the host of interest or other nearby network nodes. Network operators often assign geographically meaningful names to routers[1], presumably for administrative convenience. For example, the name *corerouter1.SanFrancisco.cw.net* corresponds to a router located in San Francisco. We stress that having geographically meaningful router names is *not* a requirement or a fundamental property of the Internet. Rather it simply an observation that is generally supported by empirical data.

We define a router to be *recognizable* if its geographic location can be inferred from its DNS name. Routers whose IP address cannot be mapped to a DNS name or whose DNS name does not contain meaningful location information are considered as not being recognizable.

GeoTrack uses these geographic hints to estimate the location of the target host. First, it determines the network path between a probe machine and the target host using the traceroute tool. Traceroute reports the DNS names of the intermediate routers where possible. Then GeoTrack extracts location information from the DNS names of recognizable routers along the path. Thus, it traces the *geographic path* to the target host. Finally, GeoTrack estimates the location of the target host as that of the last recognizable router in the path (i.e., the one closest to the target).

As noted in Section 1.2.1, traceroute-based approaches that extract geographic hints from router names have been proposed before (e.g., GTrace [27], VisualRoute [58]). However, we are not aware of work in the open literature on a quantitative evaluation of the traceroute-based approach to determining the geographic location of hosts. Our goal is precisely to do such an evaluation. Due to the logistic difficulties associated with obtaining and running existing traceroute-based tools, we decided to write our own tool based on GeoTrack to do large-scale experimentation. We have tested

[1]To be precise, DNS names are associated with router *interfaces*, not routers themselves. However, for ease of exposition we only use the term “router”.

our tool over a large sample of IP addresses and found that its coverage is comparable to VisualRoute within the U.S. and in Europe.

3.2.1 Extracting Geographic Information from Router Names

Geographic information is typically embedded in the DNS name of a router in the form of a *code*, which is usually an abbreviation for a city, state, or country name. There is no standard naming convention for these codes. Each ISP tends to use its own naming convention. This makes the task of extracting location information from DNS names challenging.

Based on empirical data, we have observed that there are basically three types of codes that indicate location: city codes, airport codes, and country codes. Some ISPs assign DNS names to routers based on the airport code of the city they are located in. Since airport codes are a worldwide standard, such a naming convention greatly eases the task of determining the router's location. For example *sjc2-cw-oc3.sjc.above.net* refers to a router in San Jose, CA (airport code *sjc*). However, many ISPs use non-standard codes for cities. We have noticed that the city of Chicago, IL has at least 12 different codes associated with it (e.g., *chcg, chcgil, cgcil, chi, chicago*). We have also observed that many routers outside the United States have the country codes embedded in their names. For example, the router with the name *asd-nr16.nl.kpnqwest.net* is located in the Netherlands (country code *nl*). The country information can be very useful in (partially) validating the correctness of the location guessed based on city or airport codes.

We examined several thousand distinct router names encountered in the large set of traceroutes that we performed from our 14 probe locations. We compiled a list of approximately 2000 airport and city codes for cities in the U.S. and in Europe. Of the entire set of airport codes [45], our list only includes a relatively small fraction of codes that are actually used in router names. Since GeoTrack deduces location by doing a string match of router names against the codes, constructing a list with as few superfluous codes as possible decreases the chances of an inadvertent match.

To further reduce the chances of an inadvertent match, we divided the list of location codes into separate pieces corresponding to each major ISP (e.g., AT&T, Sprint, etc.). When trying to infer location from a router name associated with a particular ISP, GeoTrack only considers the codes in the corresponding subset.

There is the question of how router names are matched against the location codes. Simply trying to do a string match without regard to position of the matching substring may be inappropriate. For example, the code *charlotte*, which corresponds to Charlotte, NC in the eastern U.S., would incorrectly match against the name *charlotte.ucsd.edu*, which corresponds to a host in San Diego, CA in the western U.S. Through empirical observation, we have defined ISP-specific parsing rules that specify the position at which the location code, if any, must appear in router names associated with a particular ISP. We split the router name into multiple pieces separated by dots. The ISP-specific parsing rules specify which piece(s) should be considered when looking for a match. For example, the rule for Sprintlink specifies that the location code, if present, will only be in the first piece from the left (e.g., *sl-bb10-sea-9-0.sprintlink.net* containing the code *sea* for Seattle). The rule for AlterNet (UUNET) specifies that the code, if present, will only appear in the third piece from the right (e.g., *192.atm4-0.sr1.atl5.alter.net* containing the code *atl* for Atlanta).

3.2.2 Coverage of GeoTrack

Of the 11,296 *.net* router names in our traceroute data set, 7,842 were recognizable (approximately 70%). We compiled a list of 13 major ISPs with nationwide backbones in the U.S. or with inter-

national coverage: Sprintlink, AT&T, Cable and Wireless, Internet2, Verio, BBNPlanet[2], Qwest, Level3, Exodus, PSINet, UUNET/Alter.net, VBNS, and Global Crossing. We found that 5,966 of the 6,859 router names for these major ISPs were recognizable (87%). In some individual cases, such as AT&T and UUNET, the recognizability was in excess of 95%.

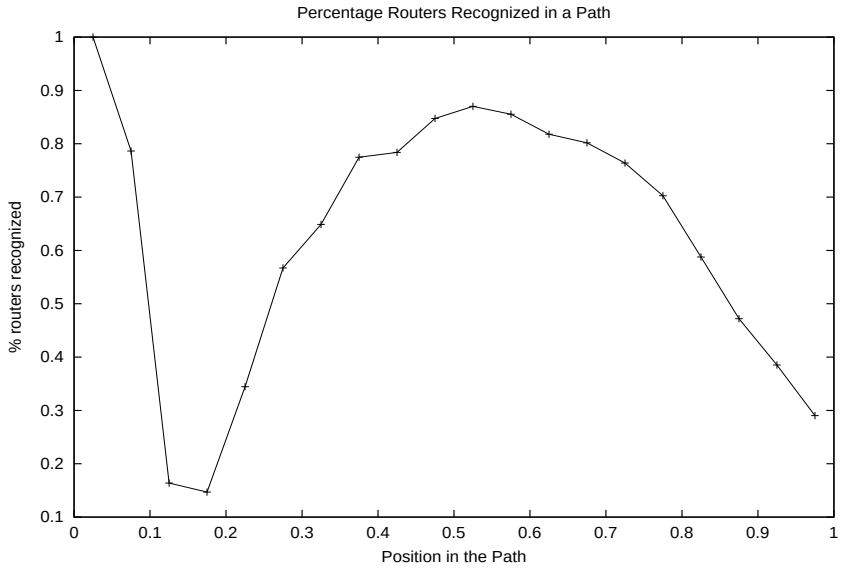

Figure 3.2: The recognizability of router names as a function of the position of the router in the end-to-end path. The position is quantified by dividing the number of hops leading up to the router by the total number of hops end-to-end.

By manual inspection, we found that a large chunk of the router names which are unrecognizable by our tool have no meaningful codes to decipher their locations. Many unrecognizable router names tend to be concentrated in regional or campus networks. (For example, *cmu.psc.net* is a node in Pittsburgh, PA. However, since it does not contain a valid city or airport code, GeoTrack is unable to recognize its location.[3]) Figure 3.2 shows that recognizability is lowest close to the start and the end of the path. (The peak corresponding to the very beginning of the path is due to the source location always being known.) Thus most of the unrecognizable nodes are typically located in the vicinity of the source or the destination, so the resulting error in linearized distance is minimal.

In the case of the 1995 data set, GeoTrack is able to recognize 1,289 out of 1,531 router names (approximately 84%). Interestingly, we noticed a huge difference in the naming convention used in 1995 and 2000. Hence we needed to create a new set of codes for the 1995 data set.

3.2.3 Performance Evaluation

We compare the performance of GeoTrack and a Whois-based tool, NetGeo [22], both for university hosts drawn from the UnivHosts data set and for a more diverse set of hosts drawn from the FooTV data set. The latter consists of a random sample of 2380 client IP addresses drawn from the FooTV data set. While many of the FooTV clients connected via proxies, none of the university hosts was

[2]BBNPlanet is now called Genuity, but the router names are still in the *bbnplanet.net* domain.

[3]Of course, it is possible to include *psc* and *cmu* as codes. However, we refrain from doing so since we only want to include those codes in GeoTrack that inherently indicate location. Doing otherwise would lead us down the path of exhaustive tabulation, which is undesirable.

behind a proxy. For this experiment, we used the probe machine at UNC in Raleigh, NC as the source of all traceroutes.

We quantify the accuracy of a location estimate using the *error distance*, which we define as the geographic distance between the actual location of the destination host and the estimated location. In the case of FooTV, the "actual" location corresponds to the zip code recorded in the FooTV data set which, as noted in Section 2.3.2, may not be entirely accurate. Also, an IP address may be associated with multiple locations, either because it was allocated dynamically (say using DHCP [8]) or because it belonged to a proxy host (such as a Web proxy or a firewall). GeoTrack, on the other hand, would only make a single location estimate for a particular IP address. In our evaluation, we compute separate error distances corresponding to the many "actual" locations associated with an IP address.

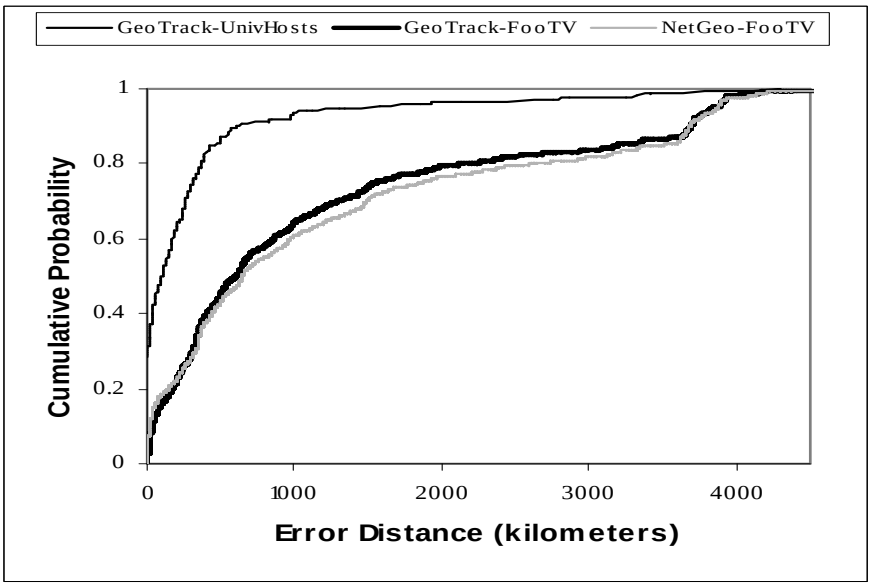

Figure 3.3: CDF of the error distance for GeoTrack and NetGeo.

Figure 3.3 shows the CDF of error distance for both GeoTrack and NetGeo. It is very interesting to note the similarity between the "NetGeo-FooTV" and "GeoTrack-FooTV" curves beyond the 70th percentile mark, and the distribution of distance of AOL clients from their proxies in Figure 3.1. GeoTrack determines the location of the AOL proxies as Washington, DC while NetGeo returns the location as Sterling, VA. The similarity in the curves can be attributed to the fact that these two locations are only about 35 km apart. (Moreover, AOL's proxies are also located in the same vicinity.)

We also observe that the performance of GeoTrack is only slightly better than that of NetGeo. GeoTrack exhibits a median error distance of 590 km and NetGeo a median of 650 km. Since many of the FooTV clients are behind proxies, neither Geotrack nor NetGeo is able to estimate the client's location accurately.

It is interesting to note that there is a significant difference in the performance of GeoTrack for the well-connected UnivHosts hosts as compared to that for FooTV clients. For instance, the median error distance is 102 km for the former while is is 590 km for the latter. The reason for this difference is that (a) none of the hosts in UnivHosts is behind a proxy, and (b) these hosts are well connected in the sense that a traceroute to them generally completes and yields a last recognizable router that tends to be close to the target host.

3.3 The GeoPing Technique

The GeoPing technique seeks to determine the geographic location of an Internet host by exploiting the relationship between network delay and geographic distance. GeoPing measures the delay to the target host from multiple sources (e.g., probe machines) at known locations and combines these delay measurements to estimate the coordinates of the target host.

3.3.1 Correlation between Network Delay and Geographic Distance

Conventional wisdom in the networking community has suggested that there is poor correlation between network delay and geographic distance [3]. This has largely been attributed to the presence of circuitous geographic paths in the Internet and bottlenecks links that cause congestion (and hence delay). However, in recent years, the Internet has grown at a very rapid pace, in terms of bandwidth as well as coverage (witness the rapid growth in the number and capacity of high-speed links, ISP points of presence, etc.). The richer connectivity (at least in well-connected portions of the Internet such as in the U.S.) often implies less circuitous routes.

To quantify impact of richer connectivity, we traced the network paths from several known locations to hosts in the UnivHosts data set. For each pair of hosts, we defined the *linearized distance* as the sum of the lengths of the individual hops along the path between the hosts. (We used GeoTrack to determine the geographic location of the intermediate nodes. We skipped over nodes whose locations could not be determined, so in general we might underestimate the linearized distance.) We compute the ratio of the linearized distance to the geographic distance between the hosts. The closer the ratio is to 1, the more "direct" (i.e., less circuitous) the network path is. Figure 3.4 shows the cumulative distribution of this ratio for paths originating from 3 different locations. The main observation we make here is that the ratio of linearized distance to geographic distance is close to 1 in the vast majority of cases. This implies that the corresponding network paths are not very circuitous.

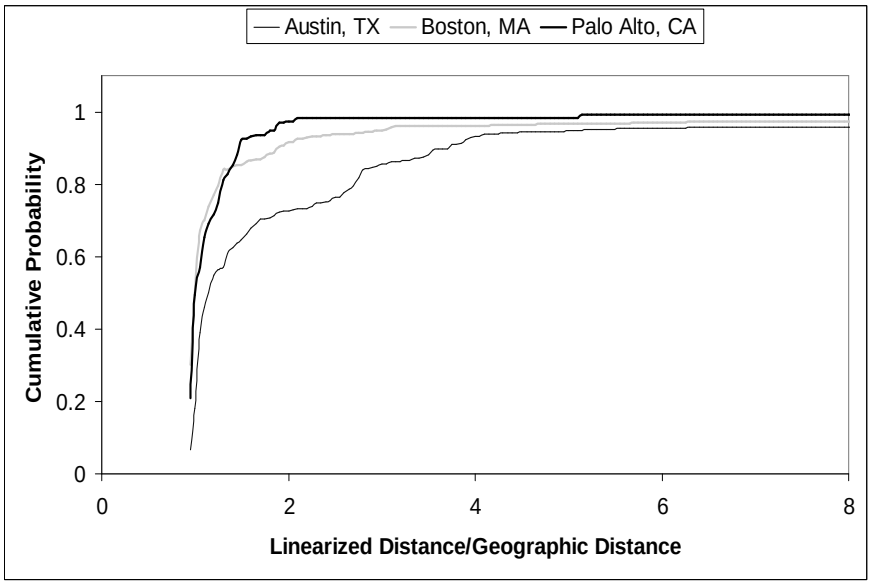

Figure 3.4: CDF of the ratio of linearized distance to geographic distance for Internet paths originating from three locations.

Congestion in the network may lead to significant queuing delays, which would also disrupt the relationship between network delay and geographic distance. To alleviate this problem, we gather

several samples for the delay between two hosts and then pick the minimum among them. (This approach has been used in several networking protocols before, e.g., TCP Vegas [4].) While not perfect, picking the minimum enables us to eliminate much of the effect of congestion. Our experiments suggest that the minimum delay stabilizes once we have at least 10-15 delay samples.

The above approach would fail in the presence of special links (e.g., dialup, satellite, etc.) that have an inherent large delay that does *not* necessarily correlate with geographic distance. We discuss possible approaches to solving this problem in Section 3.3.3.

In the following sub-sections, we present delay measurements that support our contention that there is significant correlation between (the minimum) network delay and geographic distance. Although the correlation is not perfect, we are still able to exploit it to determine location at a coarse granularity. We present a robust algorithm for this in Section 3.3.2. We present experimental results that quantify the accuracy of this algorithm and also indicate the fundamental limitations of a delay-based approach.

Experimental Setting

We use the UnivHosts data set for performing our measurements. We perform traceroutes and ping measurements from 14 different sources (All sources shown in Figure 2.1 except Seattle, WA) to all the 265 university servers in UnivHosts. After identifying the path from a given source to a host, we determine the *round-trip* delay to all intermediate routers using ping measurements. From multiple delay samples, we compute the minimum RTT to the destination and to each intermediate router in the path. We use GeoTrack to determine the physical location of intermediate routers. Using the data gathered for each source, we construct a large data set of [minimum delay, geographic distance] pairs corresponding to the paths from that source to the hosts in UnivHosts (and the intermediate routers).

CDF of Distance given Network Delay

We investigate whether there is a model that would enable estimation of geographic distance based on knowledge of network delay. For this purpose, we divide the delay range into several 10 ms wide bins and compute the CDF of geographic distance within each bin. (We decided to have a separate bin for the 0-5 ms delay range because we observed empirically that 5 ms often defines the threshold for a "metropolitan area". For instance, we found that more than 90% of the nodes within an RTT of 5 ms are located within a range of 50 km from the source.) So the delay bins we used to classify our measurements were: 0-5 ms, 5-15 ms, 15-25 ms, . . ., 125-135 ms.

Figure 3.5 shows the CDF of geographic distance for our source host located in Seattle. Many of the delay bins exhibit distinct "cliffs" (i.e., sharp upswings) in the cumulative probability distribution for specific distance values. For example, the cliff around 1300 km for the 25-35 ms delay bin is mainly contributed by locations in the San Francisco Bay Area. The other noticeable trend is that as the delay increases from 0 to 80 ms, the cliff in the CDF shifts to the right. We observed similar trends for the probes at other locations as well.

While there is a definite trend in the cliffs of the CDF for each delay range, our results suggest that the relationship between delay and distance is not strong enough to be captured in a precise mathematical model. For small delay values (under 10 ms), we found that most of the hosts (over 90%) are within a radius of 300 km from the source. However for delay values more than 40 ms, we observed an error of at least 300-400 km to obtain a 70% confidence in the distance estimate. We validated this for the data sets obtained from each of the 14 probe locations.

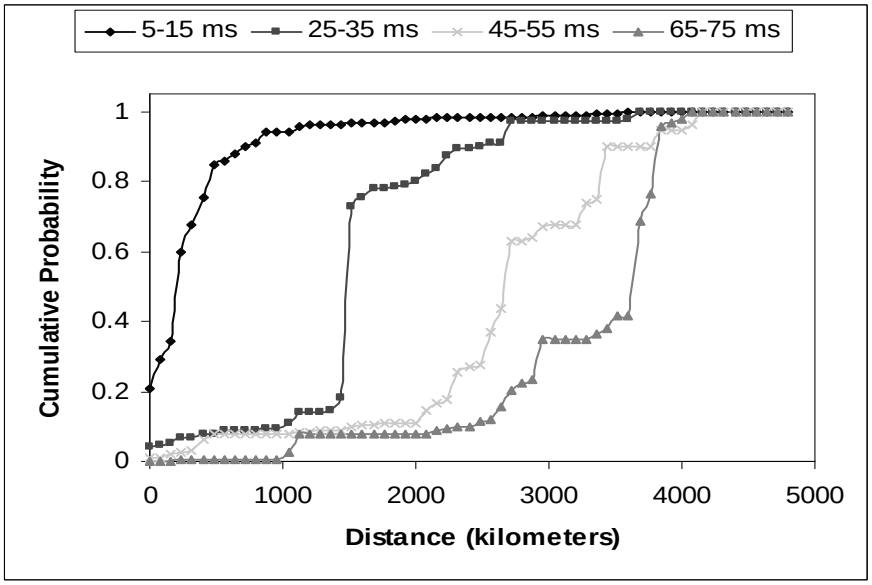

Figure 3.5: Cumulative Distribution of geographic distance for multiple delay ranges based on data gathered at the Seattle, WA probe location.

We also investigated how the relationship between delay and distance varies when we consider hosts belonging to distinct organizations located in the same geographic area. For this purpose, we considered probes located at Duke University and the University of North Carolina (UNC) located in the same metropolitan area on the U.S. east coast, and similarly Berkeley and Stanford on the west coast. We compared the CDFs corresponding to various delay ranges for each of these probe locations. The cliffs of the CDFs for Duke and UNC matched each other, and likewise for Berkeley and Stanford. This suggests that the cliffs of the CDFs are largely a function of the geographic location of a probe rather than the specific probe itself.

One limitation of our measurements is that most of our probe machines were located at university sites, many of which were connected to the high-speed Internet2 backbone [42]. The delay-distance relationship for nearby locations might not match quite as well if the probes were located at more heterogeneous sites with differing ISP connectivity. However, as we discuss next, our methodology for determining location is robust to such differences since we do not attempt to map directly from delay measurements to distance estimates.

3.3.2 Nearest Neighbor in Delay Space (NNDS)

We now discuss how GeoPing exploits the relationship between delay and distance to determine the geographic location of a host. Since we are unable to construct a precise and compact mathematical model that captures the relationship, we use an empirical approach, which we term *nearest neighbor in delay space (NNDS)*. NNDS is patterned after the nearest neighbor in signal space (NNSS) algorithm we had developed in the RADAR [2], a system to locate hosts in wireless LANs.

NNDS is motivated by the observation that hosts with similar network delays with respect to other fixed hosts tend to be located near each other. So the first step is to construct a *delay map* that records the relationship between delay and location. Each entry of the delay map contains: (a) the coordinates of a host at a known location, and (b) a delay vector, $DV = (d_1, \ldots, d_N)$, containing the measured (minimum) delay to the host from N probes at known locations. The delay map constitutes the *training* data and is constructed offline. Given a new target host, T, whose location

is to be determined, we first measure the network delay to it from the N probes. We then construct a delay vector for T as $DV' = (d'_1, \ldots, d'_N)$. Finally, we search through the delay map to find a delay vector, DV, that matches DV' the best. To find the best match, we consider the delay vectors in the delay map as forming an N-dimensional *delay space* and find the "nearest" neighbor of DV' in this space. We use Eucledian distance as the measure of distance in delay space — the Eucledian distance between DV and DV' is $\sqrt{(d_1 - d'_1)^2 + \ldots + (d_N - d'_N)^2}$. Once the nearest neighbor in delay space has been found, the corresponding location recorded in the delay map is then GeoPing's estimate of the location of the target host T.

Several aspects of NNDS contribute to its robustness: (a) delay is measured from multiple distributed locations rather than a single location, (b) the minimum among several delay samples is considered rather than the individual delay samples, and (c) the delays measurements are used as a "signature" of a geographic location rather than being directly translated into distances and location coordinates.

Typically, the delay vectors corresponding to geographically proximate locations are clustered together in delay space. However, this is not essential for NNDS to be effective. Sites located in the same city but connected via different ISPs may form more than one distinct cluster in delay space. However, as long as the number of clusters remains small, NNDS will still be effective.

We now turn to evaluating the performance of GeoPing employing NNDS.

Experimental Results

We use the delay measurements from the 14 probe machines to the 265 hosts in UnivHosts to populate the delay map. We also use the hosts in UnivHosts as the target hosts for performance evaluation. Given a target host, T, in UnivHosts whose location we are trying to determine, we exclude all data points corresponding to T in the delay map before applying the NNDS algorithm. We study the impact of the number and distribution of probe machines on the accuracy of the location estimate. For a given number of probes (say n), we compute the mean error distance as the average over all the error distances corresponding to several geographically distributed placements of n probe locations chosen from the set of 14 possible locations. For example, for 2 probes, we average the error distance over different placements of 2 probes in geographically dispersed locations among the 14 possible locations. Due to the large number of possible combinations for certain values of n (such as $n = 7$), we do not necessarily consider all possible choices of n probes out of the set of 14.

Figure 3.6 shows several percentile levels of the error distance as a function of the number of probes. For example, the 75th percentile curve corresponds to the distance at which the CDF plot of mean error distance crosses the 0.75 probability mark. From Figure 3.6, we infer that the error distance initially decreases sharply as the number of probes increases, then stabilizes and reaches an optimal value between 7 and 9 probe locations, and finally increases slightly for higher values. This suggests that having 7 to 9 probes would be ideal for the NNDS algorithm. It is also encouraging to note that NNDS with 7 probes has an error distance of only about 150 km at its 25th percentile. Our results suggest that network delay can indeed be used to determine geographic location, albeit at a coarse granularity. We expect NNDS to perform even better with a delay map constructed using a more extensive training data set and plan to investigate this is future research.

We have also investigated the impact of various probe placement strategies on the accuracy of location estimation. We have examined the effects of probe placement on the error distribution. Our findings indicate that a geographically well-distributed set of probes yields better accuracy than a clustered set of probes. For instance, the median error distance with a probe each at Stanford and Illinois was about 19% lower (i.e., better) than a probe each at Berkeley and San Diego (both of

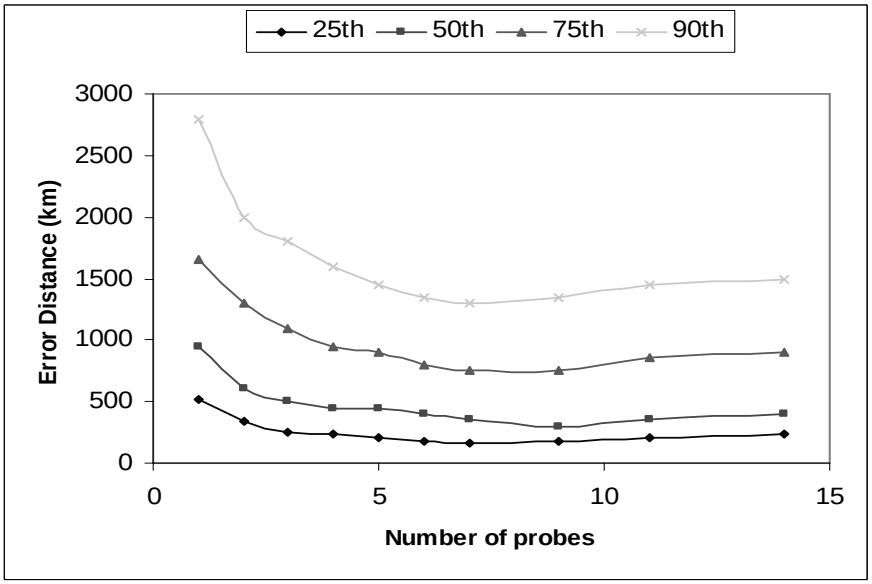

Figure 3.6: Error distance versus number of probes.

which are on the U.S. west coast). However, we found that the placement of probes has a smaller impact on performance than the number of probes.

3.3.3 Miscellaneous Issues

Finally, we discuss a few miscellaneous issues pertaining to GeoPing and NNDS.

Other Statistical Methodologies

Besides the NNDS approach, we have also investigated other statistical techniques for estimating location from delay measurements. In particular, we tried constructing an approximate model that captures the relationship between delay and distance by generating a probability density function for every source based on a large set of measurements. However, none of the alternative techniques was able to match NNDS in terms of accuracy.

Impact of Congestion and ICMP Traffic

As mentioned earlier, network congestion can introduce significant noise in the delay measurements, thereby degrading the accuracy of GeoPing (and other delay-based approaches). Our experiments suggest that 10-15 delay samples are generally (but not always) sufficient to determine the minimum delay with high confidence (i.e., the minimum delay generally did not get any lower beyond the first 10-15 samples). However, sending or receiving several ICMP packets (for ping) to the target host from each probe location may be undesirable, both because it may aggravate congestion and because it may raise a flag with intrusion detection systems. We discuss a way of alleviating this problem in Section 3.4.5.

The Last Mile

In our evaluation of GeoPing we have only considered hosts in the UnivHosts data set. These are

typically well-connected hosts on university campuses. The correlation between delay and distance may break down when we consider hosts with a "last-mile" link that has a large delay (for example, a dialup link or a satellite link). While this is clearly problematic for GeoPing, we may be able to work around it in certain situations. For instance, if we are able to tell that the user is on a dialup line (say based on the observed bandwidth to the user or traceroute measurements), we could use GeoPing to determine the location of the last router (typically located at the dialup ISP's point-of-presence) along the path to the target host. This location may serve as a good approximation for the location of the target host since users tend to dial in to modem banks in their local area.

3.4 The GeoCluster Technique

The GeoCluster technique is different from GeoTrack and GeoPing in that it does not depend on active network measurements. Instead it uses knowledge of network routing information and location information for a few hosts to build a location map for a large subset of the IP address space.

GeoCluster operates as follows. First, the IP address space is broken up into clusters such that all hosts with IP addresses within a cluster are likely to be co-located[4], i.e., the addresses form a *geographic cluster*. Then, knowing the location corresponding to a few hosts in a cluster (and assuming the locations are largely in agreement), GeoCluster deduces the location of the entire cluster.

The key to the operation of GeoCluster is IP-to-location mapping information obtained from sources such as the ones mentioned in Section 2.3.2. (We discuss the general problem of obtaining such data in Section 3.4.5.) However, this mapping information tends to be *partial* in coverage (since it includes location information only for a relatively small subset of the IP address space) and possibly *inaccurate*. These problems limit the utility of the IP-to-location mapping data.

GeoCluster addresses both of these problems by clustering IP addresses according to their (likely) location. Clustering helps expand the coverage of the partial IP-to-location mapping information. The aggregation of location information also enables us to identify and eliminate outliers caused by inaccuracies in the individual location data points.

As an example, suppose we know that 128.127.126.0/24[5] forms a geographic cluster. Furthermore assume that the partial mapping information tells us that the location corresponding to 10 different IP addresses in this cluster is Seattle while that corresponding to one other IP address is Boston. Then we can reasonably deduce that the Boston data point is erroneous and that all of the (256) IP addresses in this cluster (if they are indeed in use) are likely to correspond to hosts in (or near) Seattle.

3.4.1 Identifying Geographic Clusters

Identifying geographic clusters is a challenging problem. The basic approach used by GeoCluster is to combine partial IP-to-location mapping information with network routing information. We build on the work presented in [18] on identifying *topological* clusters. Address allocation and routing in the Internet is hierarchical. Routing information is aggregated across hosts that are under a single administrative domain (also known as an *autonomous system (AS)*). For example, the routes for hosts on a university campus would typically be advertised to the rest of the Internet as a single aggregate, say as the address prefix 128.127.0.0/16, rather than as 65536 individual IP addresses. Thus

[4]The granularity of the location depends on the application context.

[5]The notation *a.b.c.d/m* denotes an address slice with a prefix of length *m* bits specified.

knowledge of the *address prefixes (APs)* used by the routing protocol enables us to identify *topological* clusters, as observed in [18]. We surmise that APs are also likely to constitute *geographic* clusters. We elaborate on this below.

We derive information on APs from the *border gateway protocol (BGP)* used for inter-domain (i.e., inter-AS) routing in the Internet. Each entry in the BGP table at a router specifies a destination AP and the AS-level path leading to it. For our purposes, we are only interested in the AP information, so we construct a list of unique APs (over 100000 APs, as mentioned in Section 2.3.1). The number of APs is an order of magnitude larger than the number of ASs. This is because an AS, such as an ISP, may advertise more specific routes (say for certain customers) due to policy and/or performance considerations (e.g., for load balancing).

An AS (and its associated AP(s)) often corresponds to a geographical cluster such as a university campus or a company office. Even when the AS is an ISP with large geographic coverage, the associated APs that are advertised via BGP may be more specific (say corresponding to individual customers), as explained above. In both these cases, GeoCluster is in a good position able to identify geographic clusters from AP information. However, large ISPs (e.g., AT&T, Sprint, UUNet, etc.) often advertise only aggregate APs for reasons of scalability. In such cases, a single AP may span a large geographical area. This problem would be alleviated if we had more detailed knowledge of how a large aggregate is subdivided by the intra-domain routing protocol used within the ISPs. However, obtaining such information was not feasible for us, so we only use inter-domain routing information derived from BGP.

In summary, our baseline GeoCluster algorithm, which we term *BGPonly*, discovers APs based on BGP data and surmises that these APs are geographic clusters. However, as explained above this conjecture may not be always correct, for instance when ISPs only advertise large aggregates. We now present a sub-clustering algorithm designed to address this problem. We term the variant of GeoCluster that incorporates this algorithm as *BGP+subclustering*.

3.4.2 Sub-clustering Algorithm

The BGP+subclustering variant of GeoCluster depends only on inter-domain BGP data just like BGPonly. But the novel idea is to use partial IP-to-location mapping information to subdivide APs that have a large geographic spread. For each original AP obtained from E-BGP, we use the IP-to-location mapping information to determine whether their is "significant" consensus on the geographic location of the AP. If there is, then we declare the AP to be a geographic cluster. If not, we subdivide the AP into two halves (e.g., the AP 152.153.0.0/16 would be subdivided into 152.153.0.0/17 and 152.153.128.0/17) and repeat the test on each half. We stop when the subdivision contains too few IP-to-location mapping data points for a reliable determination of geographic clustering to be made. In the end, we obtain a mapping from APs (both original and subdivided ones) to location. Given an IP address, we first find the matching AP using longest prefix match and then report the corresponding location as the location of the IP address.

Here is the pseudocode for GeoCluster, including the sub-clustering algorithm. Let `IPLoclist` be the list of IP-to-location mapping data points sorted by IP address, `BGPAPlist` be the list of APs obtained from E-BGP information,
`IPLocAPlist` be the sorted list obtained by augmenting the entries in `IPLoclist` with the APs corresponding to the longest prefix match, `newAPLoclist` be the new list mapping APs to location obtained by (possibly) subdividing the original APs, and `cthresh` be the minimum threshold on the number of IP-to-location mapping data points within a subdivision.

```
/* initialization */
```

```
IPLoclist = sorted IP-to-location mapping
BGPAPlist = APs derived from E-BGP info
/* determine matching APs */
foreach ((IP,location) in IPLoclist) {
  AP = LongestPrefixMatch(IP,BGPAPlist)
  Add (IP,location,AP) to IPLocAPlist
}
/* subdivide APs using IPLocAPlist */
sameAPlist = EMPTY
curAP = AP in first entry of IPLocAPlist
foreach ((IP,location,AP) in IPLocAPlist) {
  if (AP in (IP,location,AP) == curAP) {
    /* contiguous list with same AP */
    Add (IP,location,AP) to sameAPlist
  } else {
    /* Subdivide curAP as appropriate */
    if (|sameAPlist|≥ cthresh) {
      if (sameAPlist is geographically clustered) {
        avgLocation = average location of cluster
        Add (curAP,avgLocation) to newAPLoclist
      } else {
        Divide curAP into two equal halves
        Divide sameAPlist accordingly
        Recursively test whether either/both of
subdivisions form a geographic cluster
      }
    }
    /* reset/reinitialize sameAPlist */
    sameAPlist = NULL
    Add (IP,location,AP) to sameAPlist
  }
}
newAPLoclist is the new list used for
IP-to-location mapping
```

Here is a simple example that illustrates the operation of the sub-clustering algorithm (assume that $cthresh = 15$). Consider an ISP who owns the address space 152.153.0.0/16. Suppose that the ISP has allocated half of the address space (152.153.0.0/17) to a customer in New York, and a quarter each (152.153.128.0/18 and 152.153.192.0/18) to customers in Dallas and San Francisco, respectively. Suppose that the partial IP-to-location mapping information indicates that the location is New York for 50 IP addresses in 152.153.0.0/17, Dallas for 20 addresses in 152.153.128.0/18, and San Francisco for 10 addresses in 152.153.192.0/18. The ISP only advertises the 152.153.0.0/16 prefix via BGP, so the sub-clustering algorithm starts with 152.153.0.0/16 as the presumed geographic cluster. However, there is not sufficient consensus on the location of this cluster, so the cluster is subdivided into two halves, 152.153.0.0/17 and 152.153.128.0/17. There is sufficient consensus for the former address prefix, so the algorithm declares 152.153.0.0/17 as a geographic cluster with its location as New York. However, 152.153.128.0/17 still lacks consensus, so it is subdivided into

152.153.128.0/18 and 152.153.192.0/18. There is sufficient consensus on the location corresponding to 152.153.128.0/18, so it is declared as a geographic cluster with its location as Dallas. However, there are fewer than $cthresh$ IP-to-location data points for 152.153.192.0/18, so the algorithm terminates without declaring it as a geographic cluster.

The effectiveness of the sub-clustering algorithm depends on the richness of the partial IP-to-location mapping data available. If insufficient data is available for certain APs, these will not be included in `newAPLoclist`. So GeoCluster will be unable to determine the location of IP addresses that match those APs.

We have not specified how it is determined whether a set of locations is geographically clusterered or how the consensus location of a cluster is computed. The answers to both of these questions are context-dependent — dependent on the granularity of the location information contained in the partial IP-to-location mapping and on the needs of the application.

In case the location information is relatively fine-grained (e.g., zip codes), the location of the individual points is quantifiable using latitude and longitude. So we compute a *composite* location using linear averaging of the latitudes and longitudes[6]. We also compute a dispersion metric as follows: $dispersion = \sum_{l \in L} dist(l, l_{avg}) / \mid L \mid$, where L is set of location data points corresponding to the cluster, l_{avg} is the composite location computed via averaging, and $dist(x, y)$ is the geographic distance between the locations x and y. Intuitively, the dispersion quantifies the geographic extent or spread of a cluster. We decide whether a set of locations is geographically clustered by checking whether the dispersion is smaller than a threshold.

In case location information is coarse-grained (e.g., states), we test whether there are at least $cthresh$ data points in the cluster and whether at least a threshold fraction, $fthresh$, of the points agree on location. If both conditions are met, then the consensus location is assigned to the entire cluster. As mentioned earlier, this aggregation procedure helps eliminate errors due to erroneous location information.

3.4.3 Impact of Proxies and Firewalls

Many Internet clients lie behind proxies and/or firewalls that separate the corporate or ISP network from the rest of the Internet. In such a setting, the proxy or firewall typically connects to external Internet hosts, such as Web servers, on behalf of the client hosts. The IP address of the client hosts remains hidden from the external network. As such there is no direct way to map from IP address to location for such clients. (After all we are interested in the location of the client, not that of the proxy or the firewall.)

The sub-clustering algorithm in GeoCluster deals with this issue elegantly. If the set of clients that connect via a group of proxies (having IP addresses that are contained within an address prefix AP) is clustered geographically (say at location L), then given a sufficient number of IP-to-location data points, the sub-clustering algorithm will (correctly) deduce an association between the address prefix AP and the location L. This is what happens say in the case of clients on a university or corporate campus, or clients of an ISP that connect via a local or regional proxy. However, there are instances, such as with the ISP America Online (AOL), where clients in geographically dispersed locations share a common pool of proxies. (With AOL we have seen clients thousands of kilometers apart connect via a proxy with the same IP address!) In such a case, our sub-clustering algorithm will not find sufficient consensus to be able to identify any geographic clusters, so it will not try to map the "client" IP address to a location. We believe this is an important property of the sub-

[6]While not strictly correct, such averaging is a good approximation when the individual points are close to each other

clustering algorithm because for many applications a highly inaccurate location estimate may be strictly worse than no location estimate at all. For instance, displaying a generic advertisement on a New York user's screen would probably be better than mistakenly displaying an advertisement tailored for California residents.

3.4.4 Experimental Results

We now analyze the performance of GeoCluster in several ways using a variety of data sets. We compare the performance of GeoCluster with that of GeoTrack and GeoPing. We analyze two variants of GeoCluster: (1) only using AP information derived from BGP tables (*BGPonly*), and (2) post-processing the BGP tables using the sub-clustering algorithm discussed in Section 3.4.2 (*BGP+subclustering*). We compare both variants against a simplistic approach that ignores BGP information and assumes that all APs to have a 24-bit prefix length (*/24-clusters*).

Locating hosts in UnivHosts

We first analyze the ability of the BGPonly variant of GeoCluster in determining the location of hosts in the UnivHosts data set (Section 2.2.1). We use partial IP-to-location mapping data contained in the FooTV data set as input. We convert each zip code contained in the FooTV data to the corresponding (approximate) latitude and longitude. We then cluster the (IP,latitude,longitude) data points using BGP address prefix (AP) information and compute the composite location for each AP (Section 3.4.2). Given a target IP address, we find the matching AP using longest prefix match and declare the corresponding (latitude,longitude) pair as the location estimate. We quantify the accuracy of the location estimate using the error distance.

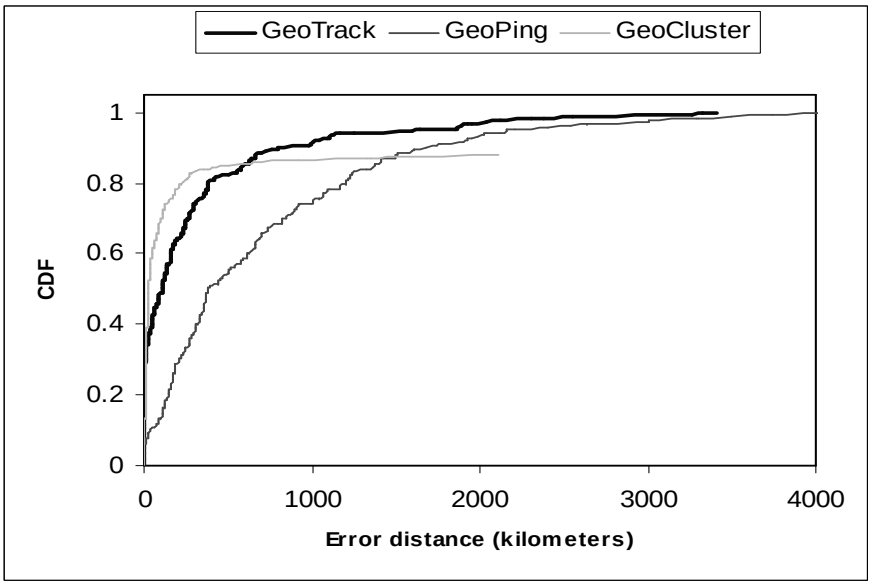

Figure 3.7: CDF of the error distance computed over the UnivHosts data set for GeoTrack, GeoPing, and GeoCluster.

Figure 3.7 shows the CDF of error distance for GeoCluster computed over the 265 university hosts. We also show the CDFs of GeoTrack and the best case of GeoPing (using 9 probe machines) for comparison. GeoCluster is able to deduce the location of only 233 out of the 265 university hosts (i.e., about 88% of the hosts). This is because the IP-to-location mapping data derived from FooTV is partial in coverage, and despite the clustering performed using BGP data, we still have no loca-

tion information for about 12% of the hosts. However, for the vast majority of hosts whose location it is able to determine, GeoCluster significantly outperforms both GeoTrack and GeoPing. For instance, the median and 80th percentile marks for GeoCluster are 28 km and 226 km, respectively. The corresponding numbers are 102 km and 384 km for GeoTrack, and 382 km and 1201 km for GeoPing.

GeoCluster performs well on the UnivHosts data set because these hosts are often clustered together geographically on university campuses. Moreover, many universities have distinct address allocations (e.g., 150.131.0.0/16 for the University of Montana) that are advertised via BGP as distinct address prefixes (APs). So GeoCluster is able to identify the universities as geographic clusters with relative ease.

Locating hosts in bCentral

We now analyze the performance of GeoCluster using the much larger bCentral data set. This data set contains 181246 unique IP addresses and their corresponding zip codes. (As noted in Section 2.3.2, the zip code information may not be entirely accurate. Hence, unlike the case of university hosts, we are not entirely certain of the true locations of the bCentral client hosts.) As before, we use the BGPonly variant of GeoCluster, with the FooTV and the BGP data sets as inputs to prime the GeoCluster algorithm.

For each IP address in bCentral, we estimate its location and then compute the error distance. The error distance, with the IP addresses sorted in increasing order of error distance, is shown in Figure 3.8. We observe that GeoCluster is only able to estimate location for about 77% of the 181246 hosts. The 25th, 50th (median), and 75th percentile marks of the error distance are 84 km, 685 km, and 3056 km respectively. In other words, GeoCluster performs much worse for the bCentral data set than for the UnivHosts data set.

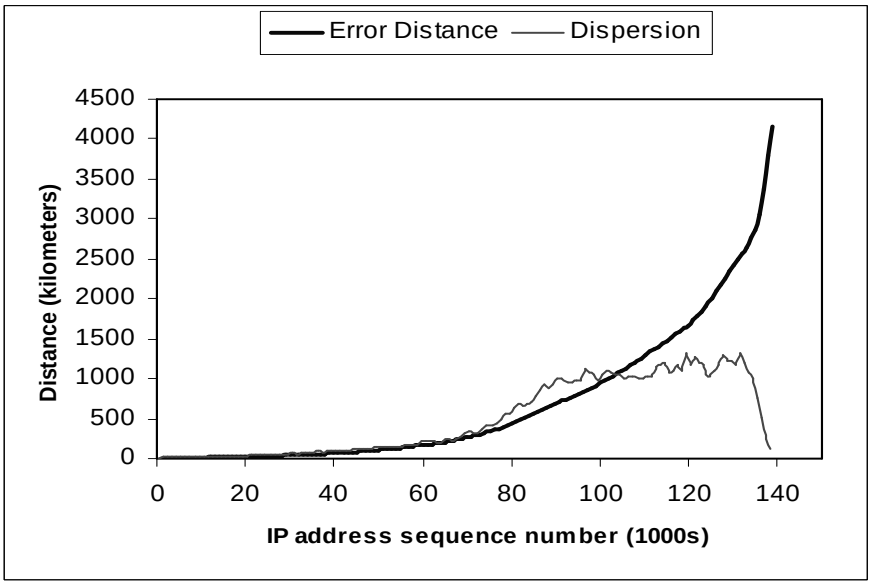

Figure 3.8: The error distance and the dispersion for hosts in bCentral.

The main reason for the worse performance is that the bCentral data set is much more diverse than the UnivHosts data set. Unlike UnivHosts, many of the IP addresses in bCentral fall within APs corresponding to large and
geographically-dispersed ISPs (e.g., 12.0.0.0/8 belonging to AT&T WorldNet) or belong to proxies

or firewalls (e.g., AOL proxies). Hence GeoCluster is only able to determine location accurately for a smaller fraction of the hosts.

Given the wide range of error distances for different hosts, it would be useful to be able to tell when GeoCluster's estimate is accurate and when it is not. For this purpose, we compute the *dispersion* metric for each AP (Section 3.4.2). We would expect that the larger the dispersion is, the less accurate GeoCluster's estimate of location would be. This is borne out by Figure 3.8, which depicts the (smoothed version of) dispersion curve for the bCentral data set. In fact, the dispersion curve matches the error distance curve quite well (except for hosts at the extreme right). This makes intuitive sense since the error in location estimation results from the geographic spread of APs, and it is exactly this spread that the dispersion quantifies.

At the extreme right of the graph, we see that error distance shoots up while the dispersion drops sharply. To better understand this puzzling phenomenon, we took a closer look at the corresponding (IP,zip) data points in bCentral. Based on this examination, we have come to the conclusion that the discrepancy is caused mainly by clients that dial in remotely. For example, bCentral contains the IP address 140.247.147.42 (DNS name *roam147-42.student.harvard.edu*), which presumably corresponds a dial up connection at Harvard University in the northeastern corner of the U.S. (and which is what GeoCluster deduces the location to be). However, the corresponding location recorded in the bCentral data set is Portland, Oregon, 4000 km away in the northwestern corner of the U.S. We hypothesize that this discrepancy is due to a user in Portland remotely dialing in to a modem bank at Harvard and then connecting to bCentral. However, it is difficult to know for sure — the Portland location may simply be erroneous, in which case the (large) error distance would be misleading.

Our results suggest that GeoCluster would not perform as well for a diverse set of hosts as for the university hosts. Still the error distance is relatively small (within a couple of hundred kilometers) for a substantial fraction (around 40%) of the hosts. And, quite importantly, GeoCluster is self-calibrating in the sense that it is often able to tell when a location estimate is likely to be accurate and when it is not.

Importance of the sub-clustering algorithm

Thus far we have considered the BGPonly variant of GeoCluster, which only uses AP information derived directly from BGP data. We now turn to the BGP+subclustering variant that employs the sub-clustering algorithm (Section 3.4.2) to construct an AP-to-location mapping. This algorithm makes use of both BGP data and partial IP-to-location mapping information. We are interested in studying what benefit, if any, the sub-clustering algorithm offers.

We use the partial IP-to-location mapping data obtained from Hotmail (Section 2.3.2) as input to the sub-clustering algorithm. Recall that the location information in Hotmail is at the granularity of states. As discussed in Section 3.4.2, we deem an AP to correspond to a geographic cluster if it contains at least $cthresh$ data points drawn from the IP-to-location mapping data set and at least a fraction $fthresh$ of those data points agree on location (i.e., correspond to the same state). In most of the results shown here, we set $cthresh = 20$ and $fthresh = 0.7$ and denote this as $(20, 0.7)$. We also briefly discuss results for the (5,0.6) setting.

We use bCentral as the test data. The location information in bCentral is at the granularity of zip codes whereas that in Hotmail is at the granularity of states. This raises the question of how to quantify accuracy. We decided to do all of our calculation at the granularity of the states. We map the zip codes in bCentral to the corresponding states. We then compute the *zipcenter* of each state by averaging the coordinates of the zip codes contained within that state (Section 2.3.2). The error distance is then computed as the distance between the zipcenters of the actual and deduced states.

So the error distance is zero if the state is deduced correctly and non-zero otherwise.

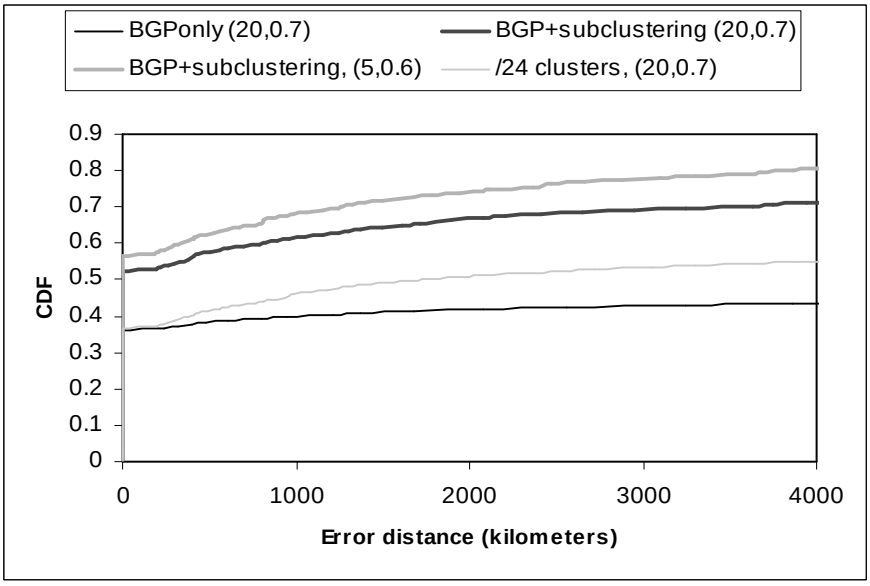

Figure 3.9: CDF of the error distance (computed at the granularity of states) for the BGPonly and BGP+subclustering variants of GeoCluster, and for the /24-clusters method.

Figure 3.9 shows the CDF of error distance. We observe that BGP+subclustering significantly outperforms BGPonly. In particular, with the $(20, 0.7)$ setting BGP+subclustering gets the state right (i.e., an error distance of zero) for 53% of the hosts while BGPonly does so only for 36% of the hosts. The reason is that BGPonly is often stuck with large and geographically dispersed APs obtained directly from BGP data while the sub-clustering algorithm is often able to break these down into smaller and geographically more compact APs. It is interesting to note that even /24-clusters, which completely ignores BGP data, outperforms BGPonly slightly, although it is still much worse than BGP+subclustering.

Finally, we see that BGP+subclustering performs slighly better with the $(5, 0.6)$ setting compared to $(20, 0.7)$ (the correct state is deduced for 56% of the hosts compared to 53%). Nevertheless we believe that a $(5, 0.6)$ setting may be too aggresive in the sense that it may often misidentify geographic clusters (after all $(5, 0.6)$ requires just 3 out of 5 data points to agree on location for an AP to be deemed a geographic cluster). We are presently investigating this issue further.

3.4.5 Discussion

In summary, GeoCluster employs a novel algorithm that combines partial IP-to-location mapping information with BGP routing information to make an intelligent determination of a client's location. The algorithm is able to tolerate a limited amount of inaccuracy in the IP-to-location mapping information and remain effective in certain situations where clients connect via proxies or firewalls.

An interesting question is how one would obtain partial IP-to-location mapping information in general. There are several possible ways one might do this.

1. The *likely* location of a user can be inferred from the kind of information accessed or queries issued by the user (for example, as in the case of FooTV). Since it only considers such information in an aggregated form (corresponding to clusters), GeoCluster is able to tolerate a limited amount of inaccuracy in the inference.

2. Certain Web sites, such as Yahoo [59], offer a mix of generic content (e.g., news) and user-specific content (e.g., email). Partial IP-to-location mapping information may be derived from accesses made by registered users to the latter content and then used in conjunction with GeoCluster to infer the location of (the presumably much larger number of) registered and casual users who access generic content.

In general, we expect that there will be a relatively small number of content providers and "location servers" (akin to advertisement servers such as DoubleClick [39]) that will employ GeoCluster (and possibly other techniques) to map IP addresses to geographic locations. The vast majority of Web sites would simply subscribe to the services provided by the location servers and so would not need to be concerned with the details of the location mapping techniques.

On a final note, we believe that the idea in GeoCluster of clustering hosts together based on geographic location may be quite useful in conjunction with GeoTrack and GeoPing. Both GeoTrack and GeoPing conduct *active* measurements by injecting traffic into the network. This may be undesirable for several reasons (network load, security, etc.). Clustering can alleviate this problem by making it unnecessary to do pings or traceroutes to *each* new target host. It may suffice to do these measurements to just a fraction of the hosts within an address prefix cluster. In fact, GeoTrack and GeoPing, used in this manner, can help GeoCluster construct the partial IP-to-location mapping that it needs.

Chapter 4

Geographic Properties of Internet Routing

In this chapter, we study the geographic properties of Internet routing. Our work is distinguished from most previous studies of Internet routing in that we consider the geographic path traversed by packets, not just the network path. We examine several geographic properties including the circuitousness of Internet routes, how multiple ISPs along an end-to-end path share the burden of routing packets, and the geographic fault tolerance of ISP networks. We evaluate these properties using extensive network measurements gathered from a geographically diverse set of probe points. Our analysis shows that circuitousness of Internet paths depends on the geographic and network locations of the end-hosts, and tends to be greater when paths traverse multiple ISP. Using geographic information, we quantify the degree to which an ISP's routing policy resembles hot-potato or cold-potato routing. We find evidence of certain tier-1 ISPs exhibiting hot-potato routing.

4.1 Circuitousness of Internet paths

In this section, we examine the nature of circuitous routes in the Internet. Since there is not a standard measure of circuitousness, we define a metric, *distance ratio*, as the ratio of the linearized distance of a path to the geographic distance between the source and destination of the path. The distance ratio reflects the degree to which the network path between two nodes deviates from the direct geographic path between the nodes. A ratio of 1 would indicate a perfect match (i.e., an absolutely direct route) while a large ratio would indicate a circuitous path.

We present several different analysis with a view to studying the impact of spatial factors as well as temporal factors. Under spatial factors, we study the effect of the geographic and network locations of end-hosts on the circuitousness of paths. To study temporal properties, we compare the circuitousness of paths drawn from Paxson's 1995 data set to the ones drawn from our 2000 data set. Finally, we analyze the relationship between the minimum delay between two end-hosts and the linearized distance along their path.

4.1.1 Effect of network location

In this section, we will vary the network location of the end-hosts (source and destination) and study its effect on the distance ratio of paths. In our first analysis, we fix a source and compare the distance ratio of paths to destinations in different networks. In our second analysis, we compare the distance ratio of paths from different sources in the same geographic location but with different network

connectivities to a set of end-hosts in the same network.

Paths from a single source

We consider paths from our traceroute sources in U.S. universities to two varied set of end-hosts: UnivHosts and TVHosts. Many of the hosts in UnivHosts (including our sources) connect to the Internet2 high-speed backbone via a local GigaPOP. So much of the wide-area path between our sources and a host in UnivHosts traverses the Internet2 backbone. On the other hand, TVHosts is a more diverse set that includes hosts located in various commercial networks (AOL, MSN, @Home, etc.) as well as university campuses. So the wide-area paths from our sources to the hosts in TVHosts typically traverse one or more commercial ISP backbones.

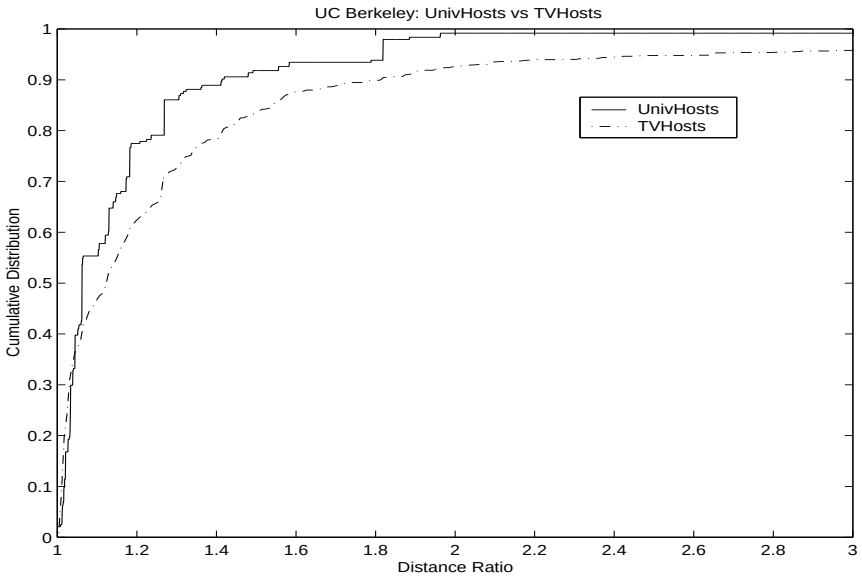

Figure 4.1: CDF of distance ratio for paths from UC Berkeley to UnivHosts and TVHosts.

This difference between the two groups of destination hosts is reflected in the cumulative distribution function (CDF) of the distance ratio for the two cases. As Figure 4.1 shows (for source in UC Berkeley), the distance ratio is close to 1 for many of the destinations. The ratio is 1.1 or less (corresponding to a linearized distance that exceeds the end-to-end geographic distance by no more than 10%) for 55% of the destinations in UnivHosts and 45% in TVHosts. This finding is consistent with the rich Internet connectivity of the San Francisco Bay Area (where UC Berkeley is located). The area includes several public Internet exchanges (e.g., MAE-West, PAIX, etc.) as well as private peering points. So a path from the UC Berkeley host to a destination host is often (but not always) able to transition to the latter's ISP within the SF bay area itself. So there is little need to take a detour through another city just to transition to the destination's ISP.

There is a far more pronounced difference between the UnivHosts and TVHosts cases if we look at the tail of the distribution. For instance, at the 90th percentile mark, the distance ratio is 1.41 in the case of UnivHosts but 1.72 in the case of TVHosts; in other words, the detour is 1.75 times as large for TVHosts destinations as it is for UnivHosts (72% versus 41%). The paths to some of the hosts in TVHosts tend to be more circuitous because they traverse multiple commercial ISPs whose peering relationships may cause detours in the end-to-end path. We discuss this issue in more detail in Section 4.2. We observe qualitatively the same trends for other university sources as well; i.e., the distance ratio tends to be smaller for paths leading to UnivHosts compared to TVHosts.

Multiple sources in the same location

We now consider paths from pairs of hosts in the same location but on entirely different networks to destinations in the UnivHosts set. We consider two such pairs of traceroute sources: (a) a machine on the Berkeley campus and another also in Berkeley but on @Home's cable modem network, and (b) a machine at the University of Washington (UW) campus in Seattle and another on the Microsoft Research network 10 km away.

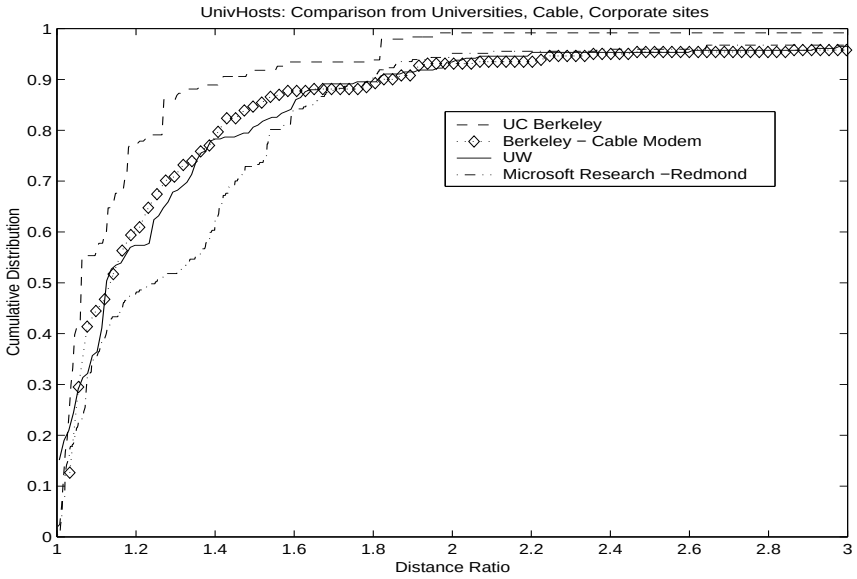

Figure 4.2: CDF of distance ratio for paths from pairs of co-located sources to UnivHosts.

Figure 4.2 shows the CDF of the distance ratio for all 4 sources. For the two sources located in Berkeley, we find that the one on the university campus has a significantly smaller distance ratio, especially at the tail of the distribution. For instance, the 90th percentile of the distance ratio for the UC Berkeley source is 1.41 while that for the cable modem source is 1.83. Since the destination set is UnivHosts, the UC Berkeley source tends to have more direct routes (via Internet2) than the cable modem client has (via @Home and other commercial ISPs).

We observe a similar trend for the UW-Microsoft pair. The UW source has more direct routes to other university hosts than does the Microsoft source. For instance, the path from Microsoft to the University of Chicago follows a highly circuitous route through BBNPlanet's (Genuity) network. The geographic path traversed includes Los Angeles, Carlton (TX), Indianapolis and Chicago (in that order). The linearized distance of the path is 4976 km while the geographic distance between Seattle and Chicago is only 2795 km. In contrast, the path from UW (via Internet2) is far more direct: it passes through Denver, Kansas City, Indianapolis, and finally Chicago, for a total linearized distance of 3533 km.

These results indicate that the nature of network connectivity of the source and the destination has a significant impact on how direct or circuitous the network paths are.

4.1.2 Effect of geographic location

The geographic location of a source indirectly determines its network connectivity. Sources near well-connected geographic locations like the Bay Area can potentially have less circuitous routes since many commercial ISPs will have a POP very close to them. To better understand the effect of

geographic location, we compare the distance ratios of sources in different locations to a common set of destination end-hosts. We extend this analysis to study the role of network structures in different continents (U.S and Europe) on the circuitousness of paths.

Multiple sources in different locations

We consider paths from sources in three geographically distributed locations in the U.S.: Stanford, Washington University at St. Louis (WUSTL), and the University of North Carolina (UNC). The destination set is LibWeb, which is a larger and more diverse set than the UnivHosts set considered in Section 4.1.1.

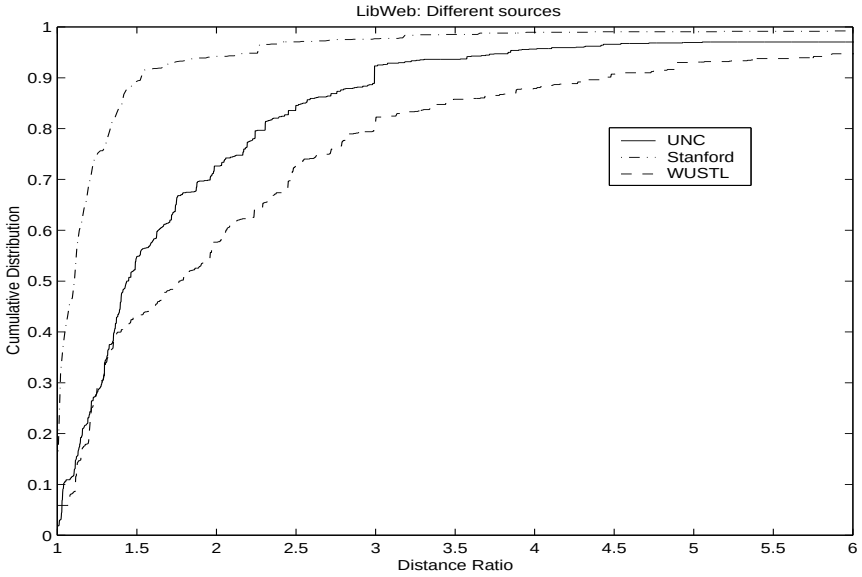

Figure 4.3: CDF of distance ratio for paths from multiple sources to LibWeb.

As shown in Figure 4.3, the distance ratio tends to be the smallest for paths originating from Stanford and the largest for those originating from WUSTL. Stanford, like Berkeley, is located in the San Francisco Bay area, which is well served by many of the large ISPs with nationwide backbones. In contrast, WUSTL is much less well connected. Almost all paths from WUSTL enter Verio's network in St. Louis and then take a detour either to Chicago in the north or Dallas in the south. At one of these cities, the path transitions to another major ISP such as AT&T, Cable & Wireless, etc. and proceeds to the destination. Any detour is particularly expensive in terms of the distance ratio because the central location of St. Louis in the U.S. means that the geographic distance to various destinations is relatively small.

In general, paths (such as those from WUSTL) that traverse significant distances in the backbones of two or more large ISPs tend to be more circuitous than paths (such as those from Stanford) that traverse much of the end-to-end distance in the backbone of a single ISP (regardless of who the ISP is). One example of a highly circuitous path we found involved two large ISPs, Verio and AT&T. The path originates in WUSTL in St. Louis and terminates at a host in Indiana University, 328 km away. However, the geographic path goes from St. Louis to New York via Chicago, all on Verio's network. In New York, it transitions to AT&T's network and then retraces its path back through Chicago to St. Louis, before finally heading to Indiana. The linearized distance is 3500 km, more than 10 times as much as the geographic distance. We examine the impact of multiple ISPs in greater detail in Section 4.2.

While the specific findings pertaining to Stanford and WUSTL may not be important in general, our results suggest that the distribution of the distance ratio is consistent with our intuition about the richness of connectivity of hosts in different geographic locations.

U.S. versus Europe

We now analyze the distance ratios for paths in Europe and compare these to the distance ratios for paths in the U.S. We consider paths from the 17 U.S. sources to destinations in the LibWeb set and also paths from the 3 European sources to destinations in the EuroWeb set. Thus, all of these paths are contained either entirely within the U.S. or entirely within Europe. We do not consider paths from U.S. sources to European destinations (or vice versa) because the distance ratio for such paths tends to be dominated by long transatlantic links (which tends to push the ratio towards 1).

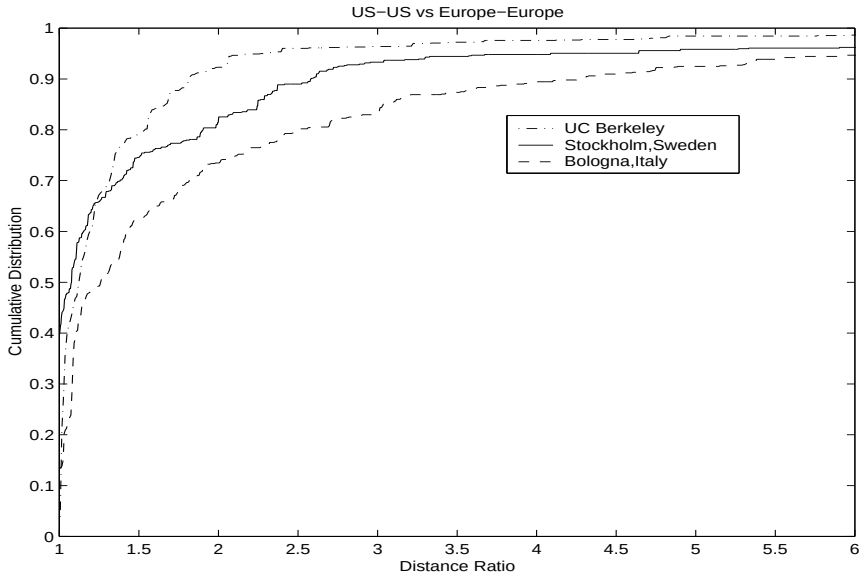

Figure 4.4: CDF of distance ratio for paths within the U.S. and those within Europe.

In Figure 4.4, we show the distribution of the distance ratio for three sources: Berkeley in the U.S., and Stockholm (Sweden) and Bologna (Italy) in Europe. We observe that the distance ratio tends to be larger for the European sources compared to Berkeley, especially in the tail of the distribution. We attribute this to three causes.

First, paths in Europe tend to traverse multiple regional or national ISPs. The complex peering relationships between these ISPs often results in convoluted paths. For instance, a path from Bologna to a host in Salzburg, Austria traverses 3 ISPs – GARR (Italian Academic and Research Network), Eqip/Infonet, and KPNQwest (a leading pan-European ISP based in the Netherlands) – and passes through Milan (Italy), Geneva (Switzerland), Paris (France), Amsterdam (Netherlands), Frankfurt (Germany), and Vienna (Austria). The linearized distance of the path is 2506 km whereas the geographic distance between Bologna and Salzburg is only 383 km.

Second, in some cases the path from a European source to a European destination passes through nodes in the U.S. For instance, a path from Stockholm (Sweden) to Zagreb (Croatia) passes through a node in New York City belonging to Teleglobe, a large international ISP. In the event that the ISPs in Europe have better connectivity to ISPs in U.S., it would be appropriate for them to route their traffic through U.S. though the route may be more circuitous. Third, geographic distances in Europe

tend to be smaller than the ones in U.S. As in the case of St Louis in Section 4.1.2, small detours in routing can be particularly expensive in terms of the distance ratio for paths between end-hosts in Europe.

4.1.3 Temporal properties of routing

To better understand some of the temporal properties of routing, we compare the distribution of the distance ratio computed from our 2000 data set with that computed from Paxson's 1995 data set [43]. The paths in the 1995 data set correspond to traceroutes conducted amongst the 33 nodes (mainly at academic locations) that were part of the testbed. We considered 340 paths between the subset of 20 nodes that were located in the U.S. The 1995 data set includes multiple traceroute measurements between each pair of hosts. In our study, we only use data from one successful traceroute between each pair of hosts. To keep the nature of the measurement points similar, in the 2000 data set we only consider paths between the 15 source hosts located at universities and the 265 hosts in the UnivHosts set.

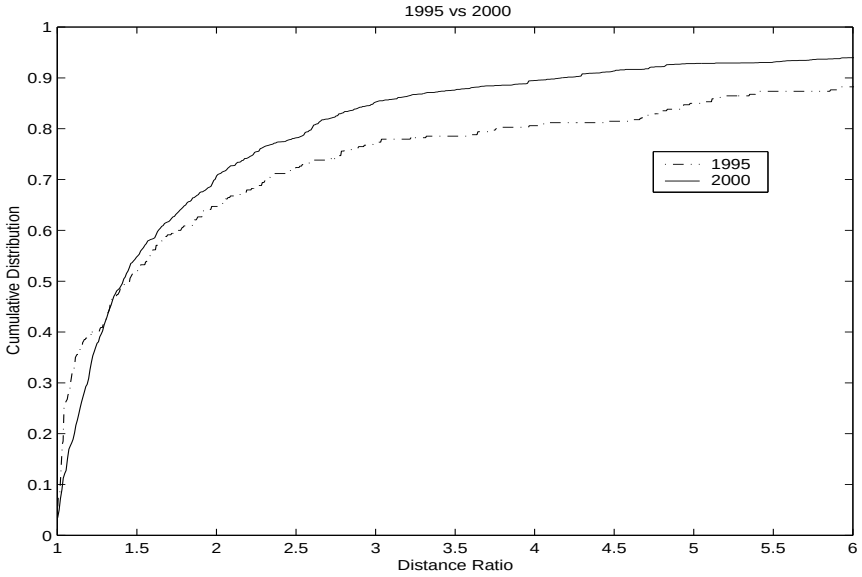

Figure 4.5: CDF of distance ratio for paths in Paxson's 1995 data set and our data set from 2000.

Figure 4.5 plots the CDF of the distance ratio for the 1995 and 2000 data sets. By observing the tail of the cumulative distribution, we find that the distance ratios tend to be smaller in the 2000 data set. This improvement is not surprising because the Internet is more richly connected today than it was 5 years ago. There now exist direct point-to-point links between locations that were previously connected only by an indirect path.

4.1.4 Correlation between delay and distance

Finally, we analyze the relationship between geography and the end-to-end delay along a path. Though geography by itself cannot provide any information about many performance characteristics like bandwidth, congestion along a path, the linearized distance of a path does enforce a minimum delay along a path (propagation delay along a path).

To study this correlation, we use the TVHosts data set since it represents a wide variety of end-hosts. In our traceroute data, we obtain 3 RTT samples for every router along the path. Since not all routers

in a path are recognizable, we consider the minimum RTT, geographic distance and linearized distance to the last recognizable router along the path. In this analysis, we restrict ourselves to the list of probes in the U.S.

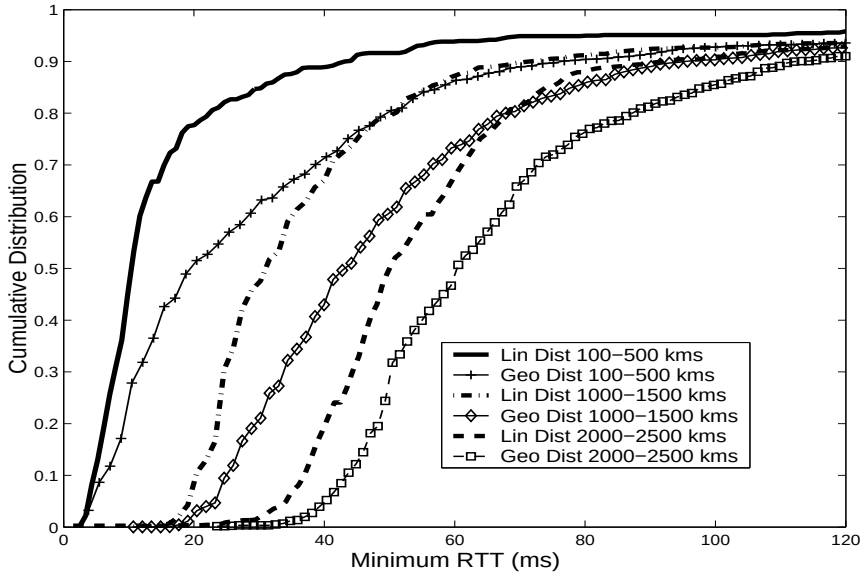

Figure 4.6: CDF of minimum end-to-end RTT to TVHosts for different ranges of linearized distances and geographic distances of paths

Figure 4.6 illustrates the correlation of the minimum RTT along a path to the linearized distance of a path and the geographic distance between the end-hosts. We make three important observations. First, at low values of the linearized distance there exists a strong correlation between the delay and linearized distance for a large fraction of end-hosts especially for small values of linearized distances. We expect this correlation to be much stronger as we compute the minimum over a larger number of samples. Second, linearized distance along a path does enforce a minimum end-to-end RTT which is an important performance metric for latency sensitive applications. Third, the minimum RTT between two end-hosts has lesser correlation to the geographic distance between them as compared to the linearized distance of the path connecting them. We observe that for a given range of linearized distance of a path, the RTT variation is much smaller than its variation for the same range of geographic distance between the end-hosts. Hence linearized distance of a path conveys more about the minimum RTT characteristics of a path than merely the geographic distance between the end-hosts. We also verified that these observations hold across the other data sets we collected. The coarse correlation between minimum delay and geographic distance was used in building GeoPing, an IP-to-location mapping service [24].

4.1.5 Summary of Results

From Sections 4.1.1 and 4.1.2, we observe that the circuitousness of a route depends on both the geographic and network location of the end-hosts. In many cases, the trends we observe in the distance ratio are consistent with our intuition. A large value of the distance ratio enables us to automatically flag paths that are highly circuitous, possibly (though not necessarily) because of routing anomalies. Finally, we show that the minimum delay between end-hosts and the linearized distance of their path are strongly correlated. This relationship indicates that the circuitousness of a route does have an effect on the delay observed along the route (though this does not completely

dictate the performance along the route).

4.2 Impact of multiple ISPs

Our analysis in Section 4.1 focused on the characteristics of the end-to-end path from a source to a destination. The end-to-end path typically traverses multiple autonomous systems (ASes). Some of the ASes are stub networks such as university or corporate networks (where the source and destination nodes may be located) whereas others are ISP networks. The relationships between these networks is often complex. There are customer-provider relationships (such as those between a university network and its ISP or between a regional ISP and a nationwide ISP) and peering relationships (such as those between two nationwide ISPs). A stub network may be multi-homed (i.e., be connected to multiple providers). Two nationwide ISPs may peer with each other at multiple locations (e.g., San Francisco and New York).

These complex interconnections between the individual networks have an impact on end-to-end routing. In this section, we show that geography can indeed be used as a means to analyze these complex interconnections. Specifically, we investigate the following questions: (a) are Internet paths within individual ISP networks as circuitous as end-to-end paths?, (b) what impact does the presence of multiple ISPs have on the circuitousness of the end-to-end path?, (c) what is the distribution of the path length within individual ISP networks, and (d) can geography shed light on the issue of hot-potato versus cold-potato routing?

4.2.1 Circuitousness of end-to-end paths versus intra-ISP paths

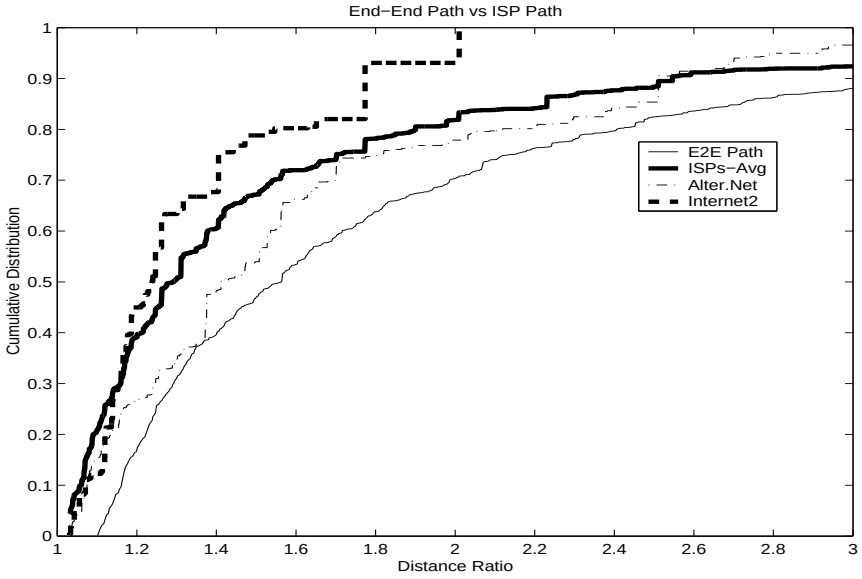

Figure 4.7: CDF of distance ratio of end-to-end paths versus that of sections of the path that lie within individual ISP networks.

We now take a closer look at the circuitousness of end-to-end Internet paths, as quantified by the distance ratio. We compare the distance ratio of end-to-end paths with that of sections of the path that lie within individual ISP networks. We consider paths from the U.S. sources to the LibWeb data set for this analysis.

As shown in Figure 4.7, the distance ratio of end-to-end paths tend to be significantly larger than

that of intra-ISP paths. In other words, end-to-end paths tend to be more circuitous than intra-ISP paths. Furthermore, the distribution of the ratio tends to vary from one ISP to another, with Internet2 doing much better than the average and Alter.Net (part of UUNET) doing worse.

We believe the reason that end-to-end paths tend to more circuitous is that the peering relationship between ISPs may create detours that would otherwise not be present. Inter-domain routing in the Internet largely uses the BGP [29] protocol. BGP is a path vector protocol that operates at the level of ASes. It offers limited visibility into the internal structure of an AS (such as an ISP network). So the actual cost of an AS-hop (in terms of latency, distance, etc.) is largely hidden at the BGP level. As a result the end-to-end path may include large detours.

Another issue is that ISPs typically employ BGP policies to control how they exchange traffic with other ISPs (i.e., which traffic enters or leaves their network and at which ingress/egress points). The control knobs made available by BGP include import policies such as assigning a local preference to indicate how favorable a path is and export policies such as assigning a multiple exit discriminator to control how traffic enters the ISP network [11]. These policies are often influenced by business considerations. For instance, packets from a customer of ISP A to a customer of ISP B in the same city might have to go via a peering point in a different city simply because a local service provider in the origin city who peers with both ISP A and ISP B does not provide transit service between the two ISPs.

Such BGP policies may partly explain the example mentioned in Section 4.1.2, where packets from a host in St. Louis to a nearby location had to travel on Verio's network all the way to New York to enter AT&T's network. We have seen several other such examples: a path from Austin, TX to Memphis, TN where the transition from Qwest to Sprintlink happens in San Jose, CA; a path from Madison, WI to St. Louis, MO where the transition from BBNPlanet to Qwest happens in Washington DC. We do not have specific information on the policies that were employed by these ISPs, so we cannot make a definitive claim that BGP is to blame. However, in view of the complex policies that come into play in the context of inter-domain routing, it is not surprising that end-to-end paths tend to be more circuitous.

In contrast, routing within an ISP network is much more controlled. Typically, a link-state routing protocol, such as OSPF [23], is used for intra-domain routing. Since the internal topology of the ISP network is usually known to all of its routers, routing within the ISP network tends to be close to optimal. So the section of an end-to-end path that lies within the ISP's network tends to be less circuitous. Referring again to the example in Section 4.1.2, both the St. Louis → Chicago → New York path within Verio's network and the New York → Chicago → St. Louis path within AT&T's network are much less circuitous than the end-to-end path.

However, this does not mean that intra-ISP paths are never circuitous. As noted in Section 4.1.1, we found a circuitous path through BBNPlanet (Genuity), from Microsoft Research in Seattle to the University of Chicago, that has a linearized distance of 4976 km whereas the geographic distance is only 2795 km. This does not imply that the path is necessary sub-optimal. In fact, the circuitous path may be best from the viewpoint of network load and congestion. The point is that while geography provides useful insights into the (non-)optimality of network paths, it only presents part of the picture.

Impact of path length on circuitousness

One question that arises from the above analysis is whether there is a connection between the circuitousness of a path and its length (i.e., the geographic distance between the two ends of the path). In other words, are longer paths inherently more circuitous, regardless of whether they traverse one

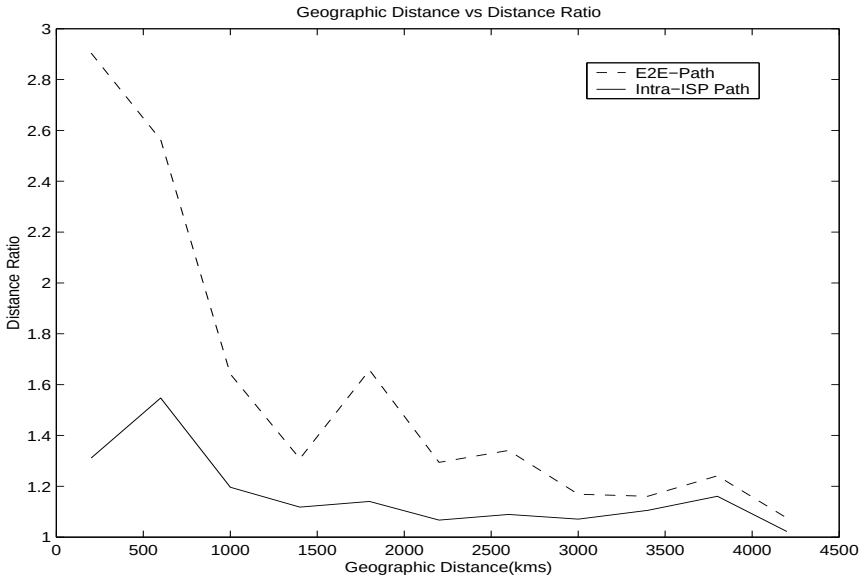

Figure 4.8: Distance ratio versus the geographic distance between the ends of a path. The median distance ratio is computed over 400 km buckets (0-400 km, 400-800 km, and so on). A minimum distance threshold of 100 km is imposed to prevent the ratio from blowing up, so the first bucket is actually 100-400 km.

ISP or many? If so, the fact that end-to-end paths tend to be longer than intra-ISP paths may explain the greater circuitousness of the former.

However, as shown in Figure 4.8, the trend is quite the opposite. The distance ratio tends to decrease as the geographic distance increases.[1] The reason is that the impact of a detour is smaller (in relative terms) in the context of a longer path. The distance ratio for the end-to-end path tends to be greater than that for the intra-ISP path, regardless of geographic distance. Thus the greater circuitousness of end-to-end paths is most likely due to the presence of multiple ISP networks in the path.

4.2.2 Impact of multiple ISPs on circuitousness

In Section 4.2.1 we hypothesized that the presence of multiple ISPs in an end-to-end path contributes to the circuitousness of the path. We now examine this issue more carefully. We classify end-to-end paths into two categories – non-circuitous (distance ratio < 1.5) and circuitous (distance ratio > 2).[2] For each path in either category, we identify the top two ISPs that account for most of the end-to-end linearized distance. We then compute the fraction of the end-to-end linearized distance that is accounted for by the top two ISPs, and denote these fractions by $\max_1$ and $\max_2$. For example, if an end-to-end path with a linearized distance of 1000 km traverses 400 km in AT&T's network and 300 km in UUNET's network (and smaller distances in other networks), then $\max_1 = 0.4$ and $\max_2 = 0.3$. Note that it is possible for $\max_1$ to be 1.0 (and so $\max_2$ to be 0.0) if the entire end-to-end path traverses just one ISP network. We note that local-area networks confined to a city (e.g., a

[1]The jaggedness of the curves arises because of the large variance in distance ratio for small values of geographic distance. The 5th and 95th percentile marks for the 100-400 km bucket are (1.00,20.50) for the end-to-end case and (1.00,4.22) for the intra-ISP case. The corresponding marks for the 4000-4400 km bucket are (1.01,1.57) for the end-to-end case and (1.00,1.18) for the intra-ISP case.

[2]While the choice of these thresholds is arbitrary, they capture the intuitive notion of circuitous and non-circuitous routes. Note that there may be paths that do not fall into either category.

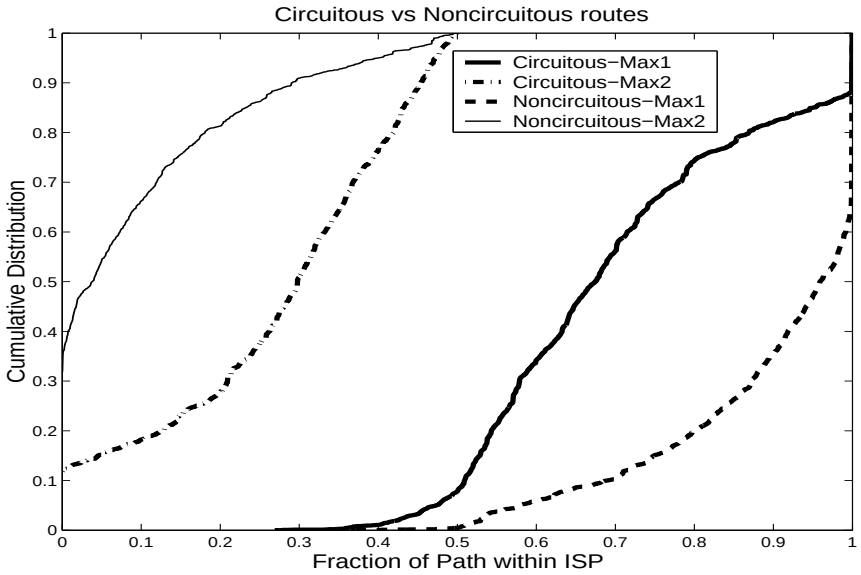

Figure 4.9: CDF of the fraction of the end-to-end path that lies within the top 2 ISPs in the case of circuitous paths and non-circuitous paths.

university network) contribute nil to the linearized distance and therefore are ignored.

Figure 4.9 shows the CDF of $\max_1$ and $\max_2$ for the circuitous and non-circuitous paths. The difference in the characteristics of these two categories of paths is striking. The $\max_1$ and $\max_2$ curves are much closer together in the case of circuitous paths than in the case of non-circuitous paths. In other words, in the case of circuitous paths, the end-to-end path traverses substantial distances in each of the top two ISPs (and perhaps other ISPs too). In contrast, non-circuitous paths tend to be dominated by a single ISP. For instance, the median values of $\max_1$ and $\max_2$ in the case of circuitous paths is approximately 0.65 and 0.3, respectively. In other words, the top two ISPs account for 65% and 30%, respectively, of the end-to-end path in the median case. However, the fractions for the non-circuitous paths are approximately 95% and 4%, respectively – much more skewed in favor of the top ISP.

We also consider the impact of the number of *major* ISPs traversed along an end-to-end path on the distance ratio. Figure 4.10 shows a clear trend: the distance ratio tends to increase as the path traverses a greater number of ISPs. For instance, the median distance ratios are 1.18, 1.25, and 1.38, respectively with 1, 2, and 3 major ISPs. The 90th percentile of the distance ratio is 1.81, 2.26, and 2.35, respectively. A path that traverses a larger number of major ISPs may span a greater distance. However, as noted in Section 4.2.1, this would not explain the larger distance ratio. In fact, a greater geographic distance would tend to make the distance ratio smaller, not larger

These findings reinforce our hypothesis that there is a correlation between the circuitousness of a path (as quantified by the distance ratio) and the presence or absence of multiple ISPs that account for substantial portions of the path.

4.2.3 Distribution of ISP path lengths

In this section, we further examine the distribution of the end-to-end linearized distance that is accounted for by individual ISPs. We wish to understand how the effort of carrying traffic end-to-end over a wide-area path is apportioned between different ISPs. For each of the 13 nationwide ISPs

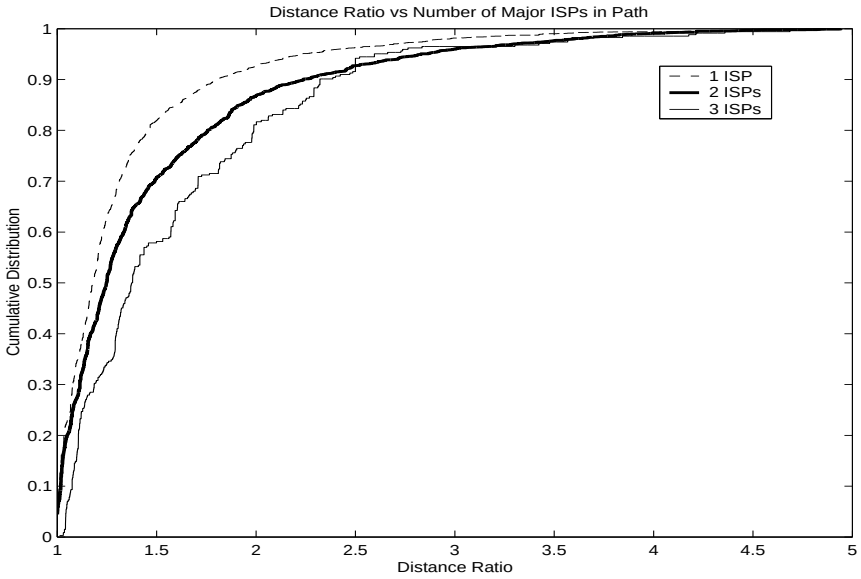

Figure 4.10: CDF of the distance ratio as a function of the number of major ISPs traversed along an end-to-end path. There were few paths that traversed more than 3 major ISPs.

in the U.S. listed in Section 3.2.2, we consider the set of paths that traverse one or more nodes in that ISP's network. For each such path, we compute the fraction of the end-to-end path that lies within the ISP's network.

Figure 4.11 plots the CDF of this fraction for a few ISPs. In each case, we consider the paths from the U.S. university sources to the LibWeb data set. We observe that the distributions look very different. For instance, the median fraction of the end-to-end path that lies within Sprintlink is only about 0.35 whereas the corresponding fraction for UUNet is 0.75 and for Internet2 is over 0.9. Internet2 is a high-speed backbone network that connects many university campuses in the U.S. An end-to-end path that traverses Internet2 typically originates and terminates at university campuses. Therefore, the Internet2 backbone accounts for an overwhelming fraction of such end-to-end paths. UUNET accounts for a larger fraction of the paths that traverse its backbone than any other commercial ISP we considered. This may reflect the close relationship between UUNET's parent company, Worldcom (which runs the vBNS backbone [57]), and academic sites.

The much smaller fraction in the case of Sprintlink is harder to explain definitively. From our conversations with people at Sprint [7, 21], we have learned that academic sites are not their major customers, so Sprintlink participates minimally in carrying academic traffic. The location of our traceroute sources at academic sites may explain why Sprintlink only accounts for a small fraction of the end-to-end path.

We stress, however, that the point of our analysis is not to make general claims about certain ISPs being better or worse than others. Rather it is to show that geographic analysis of end-to-end paths yields interesting insights into the role played by multiple ISPs in specific contexts (e.g., academic sites) and that these insights are consistent with our intuition.

4.2.4 Hot-potato versus Cold-potato routing

Finally, we investigate whether geographic information can be helpful in assessing whether ISP routing policies in the Internet conform to either hot-potato routing or cold-potato routing. In hot-

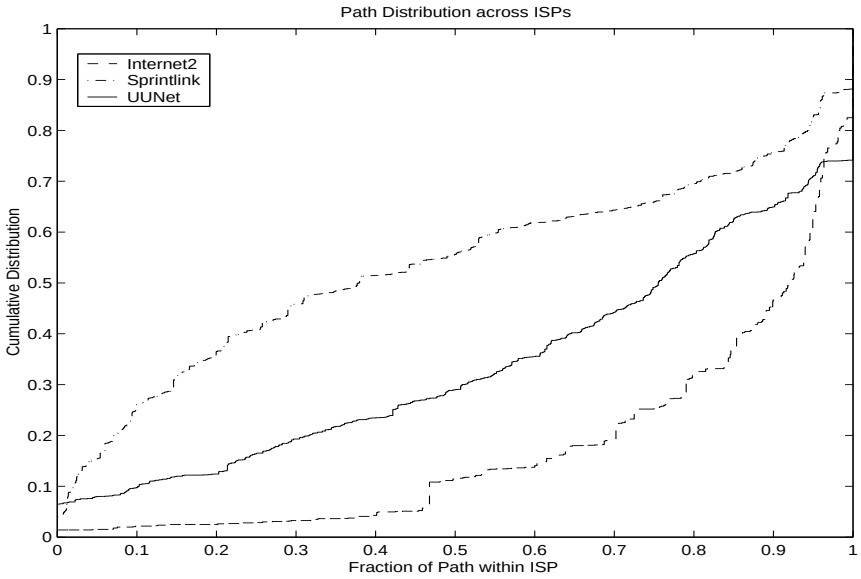

Figure 4.11: CDF of the fraction of the end-to-end path that lies within individual ISP networks.

potato routing, an ISP hands off traffic to a downstream ISP as quickly as it can. Cold-potato routing is the opposite of hot-potato routing where an ISP carries traffic as far as possible on its own network before handing it off to a downstream ISP. These two policies reflect different priorities for the ISP. In the hot-potato case, the goal is to get rid of traffic as soon as possible so as to minimize the amount of work that the ISP's network needs to do. In the cold-potato case, the goal is carry traffic on the ISP's network to the extent possible so as to maximize the control that the ISP has on the end-to-end quality of service. In general, an ISP's routing policy would lie somewhere in between the extremes of hot-potato and cold-potato routing.

We consider the set of paths from U.S. sources to TVHosts. For each path that traverses two or more major ISPs (with nationwide backbones), we compute the fraction of the end-to-end path that lies within the first major ISP (ISP1) and the second major ISP (ISP2) in sequence. We use these fractions as measures of the amount of work that these ISPs do in conveying packets end-to-end. The distributions of these fractions is plotted in Figure 4.12. We observe that the fraction of the path that lies within the first ISP tends to be significantly smaller than that within the second ISP. For instance, the median is 0.22 for the first ISP and 0.64 for the second ISP. This is consistent with hot-potato routing behavior because the first ISP tends to hand off traffic quickly to the second ISP who carries it for a much greater distance.

Figure 4.12 also plots the distributions of the path lengths in the case where the first ISP is Sprintlink. We find that the difference between the ISP1 and ISP2 curves is even greater in this case. Again, this is consistent with hot-potato routing behavior on the part of Sprintlink for routes from academic locations.

4.2.5 Summary

In this section, we have used geographic information to study various aspects of wide-area Internet paths that traverse multiple ISPs. We found that end-to-end Internet paths tend to be more circuitous than intra-ISP paths, presumably because of the peering relationships between ISPs. Furthermore,

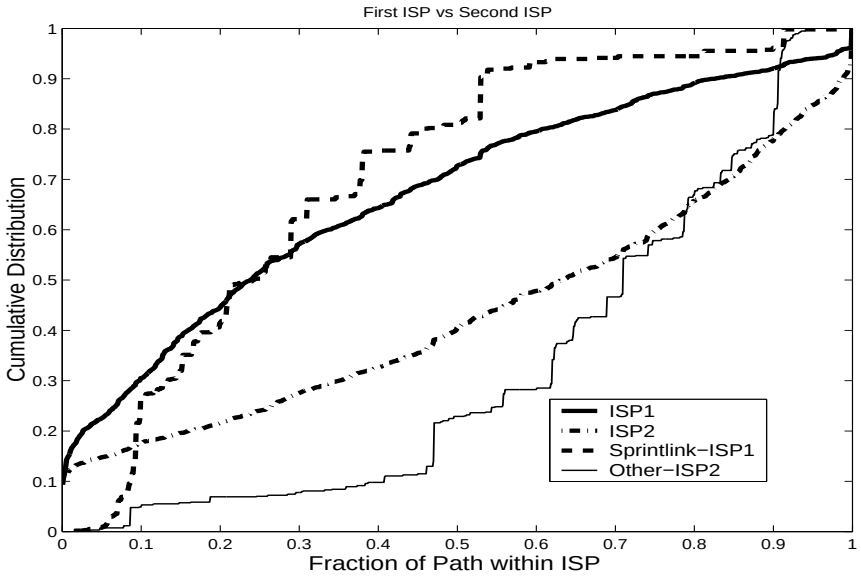

Figure 4.12: CDF of the fraction of the end-to-end path that lies within the first and second ISP networks in sequence.

paths that traverse substantial distances within two or more ISPs tend to be more circuitous than paths that largely traverse only a single ISP. Some of this circuitous routing behavior can be attributed to sub-optimal geographic peering between ISPs. Finally, the findings of our geography-based analysis are consistent with the hypothesis that ISPs generally employ hot-potato routing. The presence of hot-potato routing may also explain for why some major ISPs only account for a relatively small fraction of the end-to-end path.

4.3 Limitations and Possible Inaccuracies

Our analysis of geographic properties of Internet routing suffers from a few limitations. We will first describe some of the possible inaccuracies in our results due to our methodology and how we reduce the effect of these inaccuracies.

4.3.1 Possible Inaccuracies

First, the city codes used in GeoTrack for computing the location of router given its label are manually determined and encoded. Hence there is always a possibility that the location of a router as determined by GeoTrack is incorrect. However, we have greatly reduced the possibility of such errors by using delay-based verification, ISP specific parsing rules and manual inspection. In delay-based verification, we perform the following simple check: if the difference between the minimum RTTs to two adjacent routers in a path is not high, the distance between them cannot be large. This simple check helped us distinguish between two cities named *Geneva* that had similar city codes — one in Switzerland and the other in Texas. We have enumerated specific rules for 52 different ISPs (all major ISPs in our data set) which specify the exact position where a city code is embedded in a label. This, in conjunction with ISP specific city-codes, greatly reduces the chances of a wrong location output. We have also manually inspected the geographic paths corresponding to a large sample of our traceroute data to check for any possible errors.

Second, the linearized distance computed can be distorted if the geographic locations of many

routers in a path are unknown. We reduce this distortion by restricting our analysis to paths that have at least 4 recognizable intermediate routers. The linearized distance of a path can also be skewed due to intra-metro distances. Intra-metro distances will affect our analysis only for small values of linearized distances. To reduce this skew, we only consider paths with a linearized distance greater than 100 kms in our study.

4.3.2 Limitations

We now discuss the limitations of our study arising both due to the inherent limitations of geographic information and due to limitations of our experimental methodology.

1. **Geography does not determine performance:** There is not a perfect relationship between geographic distance and network performance. It is possible that a circuitous path yields better performance than a less circuitous one. For instance, the most optimal path between certain countries may be via the U.S. even if that means a large detour in geographic terms. However, in Section 4.1.5, we show that there exits a strong correlation between the minimum end-to-end delay between two end-hosts and the linearized distance of their connecting path. In light of this, we view our geographic analysis of network paths as providing (a) hints on paths that are *potentially* anomalous and should be examined more closely to determine if they are indeed anomalous, (b) an indication of how much improvement there could be in end-to-end latency if a non-circuitous path between source and destination were feasible, and (c) a way to quantify network properties such as hot-potato routing, which may provide new insight into these properties.

2. **IP-level topology is incomplete:** Our linearized distance computation only considers the router-level (i.e., IP-level) topology. We have no way of discovering the underlying physical topology (which may be based on ATM, SONET, or other technologies), so in general we would underestimate the linearized distance. While this is a limitation of our methodology, we note that the trend in high-speed networks (OC-48 and faster) is away from separate layer-2 and layer-3 architectures (e.g., IP-over-ATM) and towards an all-IP network [30]. This trend increases the applicability of our methodology.

Chapter 5

Geographic Fault Tolerance

An important component of studying Internet routing is to understand its fault tolerance aspects. Fault tolerance of a network is normally studied at the granularity of router or link failures. However such a failure model does not capture the fact that two seemingly independent routers can be susceptible to correlated failures.

We ask the question: what is the tolerance of an ISP's network to a *total* network failure in a geographic region, i.e., a failure that affects all paths traversing the region? We refer to such a failure as a *geographic failure*. Potential reasons for such a failure include natural calamities such as earthquakes or power blackouts.

By using the geographic location information of the routers, we can identify routers that are co-located and thereby construct a *geographic topology* of an ISP. In this topology, each geographic region is associated with a node and an edge between two nodes signifies the existence of at least one long-haul backbone link that connects the corresponding geographic regions.

We obtained the geographic topologies for 9 of the 13 major ISPs listed in Section 3.2.2 from the CAIDA MapNet site [49]. These are: AT&T, Cable and Wireless, Sprintlink, Genuity, Qwest, PSINet, UUNet, Verio and Exodus. Many of these topologies are obtained from information published at the ISPs' Web sites and are between 6-12 months out of date. Although it may be possible to construct an ISP's geographic topology using extensive traceroute measurements, it would be hard to assess the completeness of the constructed topology. Hence we restrict ourselves to the geographic topologies obtained from CAIDA. However, as acknowledged by CAIDA [49], it is possible that these topologies may themselves be incomplete. This may be due to limited tracing or the presence of backup paths in routing. We will perform our analysis under the assumption that these topologies are reasonably complete and only have a few missing links.

5.1 Degree distributions

The degree of a node provides a first-level quantification of the fault tolerance of that node in a given topology. A node with a degree k can tolerate up to k geographic failures before getting completely disconnected from all other nodes in the topology. In particular, a leaf node is not resilient to the geographic failure of its neighbor, but the failure of a leaf node itself has minimal impact on the rest of the network. On the other hand, the failure of a node with a very high degree would impact its many neighbors (corresponding to many different geographic regions).

Given complete freedom in placing $E = k * N$ edges on N nodes, it is possible to construct a topology that has a minimum vertex-cut of $2k$. In other words, the E edges can be placed in such a

way that even in the presence of any $2k-1$ node failures in the graph, the resulting topology will still remain connected. We term such a placement of edges that maximizes the size of the vertex cut as an *optimal placement*. In the optimal placement, all the vertices have the same degree, viz. $2*k$. For the simple case of $k=1$, the optimal placement results in a ring topology. Although this optimal placement may be difficult to construct due to practical constraints, it provides us a nice reference point for comparing the fault tolerance of ISP topologies. In order to contrast an ISP's topology from the optimal scenario, we look at the degree distribution of the nodes. We say that a graph has a *skewed* degree distribution if its node degrees are distributed over a wide range with a few large node degrees and a high percentage of the nodes are leaves. The Internet topology exhibits a skewed degree distribution which can be characterized by a power law as described in [10].

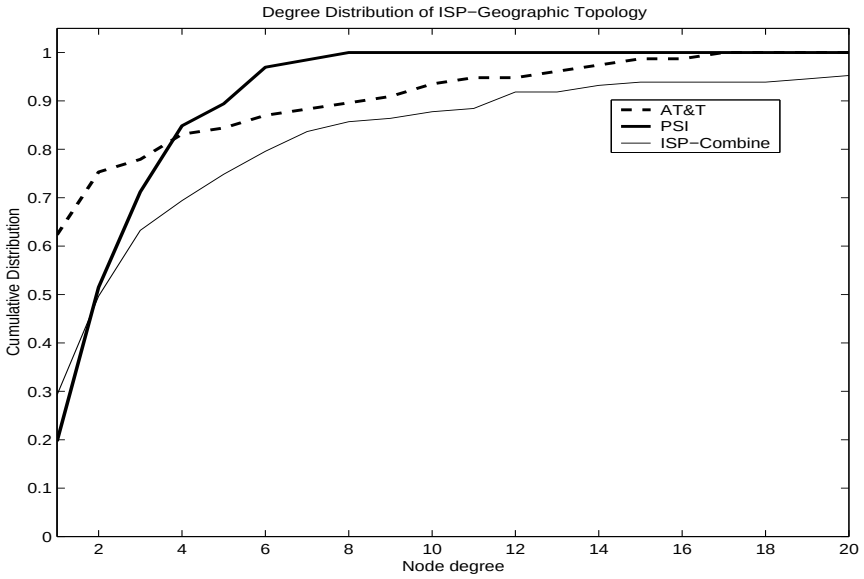

Figure 5.1: Degree Distribution of Geographic Topologies of ISPs

Among the 9 commercial ISPs, some of them such as AT&T and Genuity have a very skewed degree distributions while other ISPs such as PSINet and Verio have much less skewed degree distributions (closer to optimal). The degree distribution will not be affected much due to a few missing links. Figure 5.1 shows the degree distributions of AT&T and PSINet. AT&T's topology has the maximum percentage of leaves among the 9 ISP topologies (62%) and has a few nodes with a degree greater than 12 (Chicago, Dallas). On the other hand, more than 50% of PSINet's nodes have a degree of either 2 or 3. This matches the optimal degree for Verio given that it has an edge to node ratio $k=1.5$, which corresponds to an optimal degree of $2*k=3$. The ISP-Combine curve shows the degree distribution of the geographic topology obtained by combining the topology graphs of all 9 ISPs. The geographic nodes corresponding to the same city in the individual ISP topologies map to a single node in the combined topology. The combined topology still has a significant skew in its degree distribution. 29% of the nodes continue to be leaves. This happens despite the combined topology having an edge to node ratio of $k=2.5$, which corresponds to an optimal degree of 5. On the other hand, nodes located in the important networking hubs of U.S. (e.g, San Jose, Washington DC, Chicago) have a degree of more than 20 in the combined topology.

5.2 Failure of high connectivity nodes

The skewed degree distributions of many tier-1 ISPs indicate that many geographic regions of an ISP may get disconnected if some high connectivity geographic nodes fail. To evaluate this, we consider the failure scenario where the f nodes of highest degrees in a graph fail.

We define a pair of geographic nodes that are connected by a network path and can communicate with each other as a *communicating pair*. A connected topology of N nodes can support $N(N+1)/2$ communicating pairs. (Since each node represents a geographic *region*, we also consider intra-node communication of a node with itself.) Under the scenario where the f nodes of highest degrees fail, the graph is disconnected into a forest where a node can only communicate with other nodes in its connected component. A connected component with $m < N$ nodes can support $m * (m + 1)/2$ communicating pairs. In the simple case where the parent of a leaf node fails, it produces a connected component of size 1 which supports exactly one communicating pair.

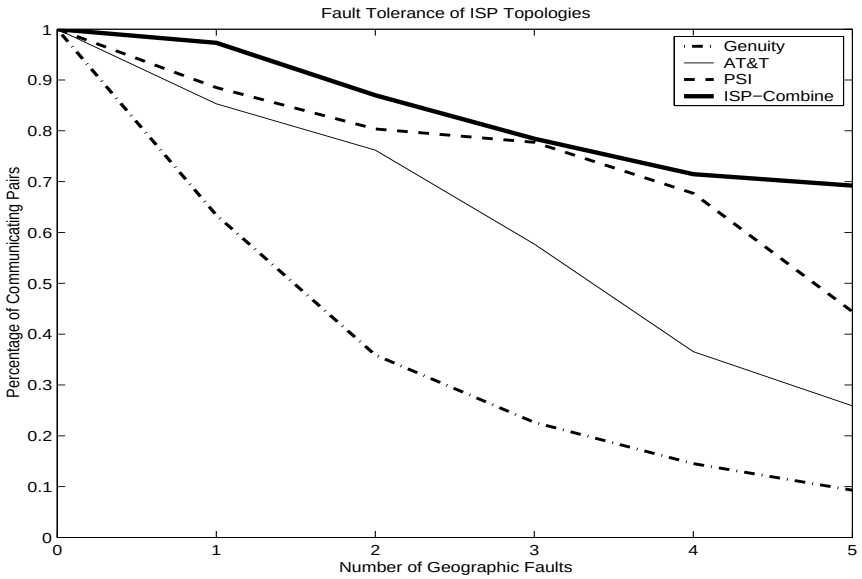

Figure 5.2: Tolerance to Geographic Failures

Figure 5.2 shows the percentage of communicating pairs supported in the various ISP networks in face of a varying number of geographic failures. The combined topology of the 9 ISPs supports 68% of the communicating pairs even after the removal of 5 important networking hubs in the US (San Jose, New York, Washington DC, Chicago, Los Angeles). Among the 9 ISPs, while Genuity and PSINet exhibit the least and the best fault tolerance characteristics. In the face of a single node failure, most of the ISPs lose between 15% and 30% of their communicating pairs in the worst case.

It is important to note that these results may represent a near-worst case failure scenario for the ISPs. If, however, many backup links are missing from our topology, the fraction of communicating pairs may be much higher than what we have portrayed. However, our essential message from this analysis is that a balanced degree distribution is a good feature for building a fault tolerant topology for an ISP.

Chapter 6

Conclusions

In this work, we have analyzed and inferred different geographic properties of the Internet. We classify our work into three categories:

- **IP-Geography mapping:** An investigation of geographic mapping techniques of Internet hosts.
- **Geographic properties of routing:** Quantify different properties of routing like circuitous routing and hot-potato routing which cannot be quantified without geographic information.
- **Geographic Fault Tolerance:** Analyze the fault tolerance of an ISP in the presence of geographic node failures.

6.1 IP-Geography mapping

In IP-Geography mapping, we have examined the interesting but challenging problem of determining the geographic location of an Internet host knowing only its IP address. We have designed and evaluated three distinct techniques, collectively referred to as IP2Geo, to address this problem: (a) *GeoTrack*, which extracts location information from DNS names of hosts and routers, (b) *GeoPing*, which determines location using network delay measurements made from several known locations, and (c) *GeoCluster*, which combines partial IP-to-location mapping information with BGP routing data to determine location. These techniques span a broad spectrum. Our evaluation of these techniques was based on extensive and varied data sets.

Our findings suggest that GeoCluster is the most promising one of the IP2Geo techniques. The median error distance for GeoCluster varies from 28 km for well-connected university hosts to a few hundred kilometers for a more heterogeneous set of clients. Importantly, however, GeoCluster is self-calibrating in that the *dispersion* metric offers an indication of how accurate a location estimate is likely to be. Furthermore, the *sub-clustering* technique is often able to infer more fine-grained (geographic) structure in Internet address ranges than is present in BGP routing data. Both these features make GeoCluster more suitable than the other techniques in the presence of clients that connect via proxies. Finally, GeoCluster is passive in that it does not inject extra traffic into the network.

Our investigation of GeoTrack and Whois-based techniques reveals the fundamental limitation due to proxies. Our evaluation of GeoPing suggests that contrary to conventional wisdom there is a

significant correlation between network delay and geographic distance that can be exploited to determine coarse-grained location. We believe this will be the case even more as the Internet becomes richly connected.

Our study also indicates that geography can be an interesting tool for analyzing the behavior of network routing. The ratio of *linearized* distance to geographic distance is indicative of how "direct" a network route is. A large ratio may be indicative of an anomalous route. For instance, by computing this ratio, GeoTrack was able to automatically flag an a highly circuitous route from Austin, Texas to Kentucky via California, New Jersey, and Indiana!

Besides the specific techniques that we have developed, we believe an important contribution of our paper is that the systematic study of the IP-to-location mapping problem using a wide range of interesting data sets.

6.2 Geographic properties of routing

Our study on geographic properties of routing concentrated mainly on quantifying those aspects of Internet routing which are not characterizable using network-centric metrics like delay and bandwidth. First, our analysis based on extensive traceroute data shows the existence of many circuitous routes in the Internet. From the end-to-end perspective, we observe that the circuitousness of routes depends on the geographic and network locations of the end-hosts. We also find that the minimum delay along a path is more strongly correlated with the linearized distance the path than it is with the geographic distance between the end-points. This suggests that the circuitousness of a path does impact its minimum delay characteristics, which is an important end-to-end performance metric. In ongoing work, we are studying the correlation between geography and network performance.

Second, a more careful examination shows that many circuitous paths tend to traverse multiple major ISPs. Although many of these major ISPs have points of presence in common locations, the peering between them is restricted to specific geographic locations, which causes the paths traversing multiple ISPs to be more circuitous. We also found that intra-ISP paths are far less circuitous than inter-ISP paths. An important requirement to reduce the circuitousness of paths is for ISPs to have peering relationships at many geographic locations.

Third, the fraction of the end-to-end path that lies within an ISP's network varies widely from one ISP to another. Furthermore, when we consider paths that traverse two or more major ISPs, we find that the path generally traverses a significantly shorter distance in the first ISP's network than in the second. This finding is consistent with the hot-potato routing policy. Using geographic information, we are able to quantify the degree to which an ISP's routing policy resembles hot-potato routing. The traceroute data we collected is available on-line at [54].

6.3 Geographic Fault Tolerance

Finally, our analysis of geographic fault tolerance of ISPs indicates that the (IP-level) network topologies of many tier-1 ISPs exhibit skewed degree distributions which may induce a low tolerance to the failure of a single, critical geographic node. The combined topology of multiple ISPs exhibits better fault tolerance characteristics, assuming that the ISPs peer at all geographic locations that are in common. In our analysis, we assume that the published topologies of ISPs are reasonably complete.

6.4 Directions for Future Work

An important dimension that we have not carefully explored in our study is the relationship between geography and performance. In our analysis of routing properties, we found a strong correlation between the end-to-end delay between two end-hosts and the linearized distance of the path connecting them. This seems to suggest that geography may have a certain level of correlation with some performance metrics like delay. One open question that arises is the usefulness of Geography-based service redirection. Though this basic idea has been suggested in previous works [13], the trade-offs of such a solution have not been carefully studied. The main advantage of geography based redirection is its simplicity and flexibility. Not surprisingly, geography is a simple and understandable user interface which is used by popular web servers to make end users choose the closest site for downloading large data content like software distributions.

Some aspects of our IP-Geography mapping work need further exploration. We are trying to see whether we can combine the different techniques proposed in our work to build a mapping service which has a much better accuracy than the individual techniques themselves. Also, we are exploring alternatives to overcome the fundamental limitations imposed by proxies. With the advent of IPv6, we expect a better way of allocating IP addresses to end-hosts. We hope the IP-Geography mapping would be easier to solve for IPv6 addresses.

Finally, our geographic fault tolerance analysis of ISP topologies is a very preliminary study and can be expanded across many dimensions. First, we found that the combined topology of ISPs has much better tolerance to geographic failures than individual ISP topologies. Though this is true from a topology perspective, we require the underlying ISPs to peer at all common geographic locations to realize this level of fault tolerance. An associated optimization problem is to determine the optimal set of peering locations between ISPs to realize a certain level of fault tolerance. Second, from a single ISP's perspective, there exists a trade-off between the fault tolerance of its topology and the amount of fiber that needs to be laid. Similar to the previous case, we can optimize the fault tolerance of an ISP's topology given the corresponding economic constraints.

Bibliography

[1] D. G. Andersen, H. Balakrishnan, R.Morris, and F. Kaashoek. Resilient Overlay Networks, *ACM SOSP*, November 2001.

[2] P. Bahl and V.N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. *IEEE Infocom*, March 2000.

[3] G. Ballintijn, M. van Steen, and A.S. Tanenbaum. Characterizing Internet Performance to Support Wide-area Application Development. *Operating Systems Review, 34(4):41-47*, October 2000.

[4] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. *ACM SIGCOMM*, August 1994.

[5] B. Cheswick, H. Burch, and S. Branigan. Mapping and Visualizing the Internet, *USENIX Technical Conference*, June 2000.

[6] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE project. *ACM Mobicom*, August 2000.

[7] C. Diot. Personal communication, November 2001.

[8] R. Droms, Dynamic Host Configuration Protocol, *RFC-1531, IETF*, October 1993.

[9] P. Enge and P. Misra, The Global Positioning System, *Proc. of the IEEE*, January 1999.

[10] M. Faloutsos, P. Faloutsos and C. Faloutsos. On Power-Law Relationships of the Internet Topology. *ACM SIGCOMM*, August 1999.

[11] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE Global Internet*, November 2000.

[12] R. Govindan and H. Tangmunarunkit. Heuristics for Internet Map Discovery. *IEEE Infocom*, March 2000.

[13] J. Gwertzman and M. Seltzer. The Case for Geographical Pushcaching . *Proc. of the 1995 Workshop on Hot Operating Systems*, Orcas Island, WA, May 1995, pp 51-55.

[14] K. Harrenstien, M. Stahl, E. Feinler, NICKNAME/ WHOIS, *RFC-954, IETF*, October 1985.

[15] Andy Harter and Andy Hopper. A Distributed Location System, for the Active Office. *IEEE Network* Vol.8 No.1, January 1994.

[16] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, The Anatomy of a Context-Aware Application, *ACM Mobicom*, August 1999.

[17] V. Jacobson, Traceroute software, 1989, *ftp://ftp.ee.lbl.gov/traceroute.tar.gz*

[18] B. Krishnamurthy and J. Wang. On Network Aware Clustering of Web Clients. *ACM SIGCOMM*, August 2000.

[19] C. Labovitz, J. Malan, and F. Jahanian. Internet Routing Instability. *ACM SIGCOMM*, August 1997.

[20] J.Y. Li et al. A Scalable Location Service for Geographic Ad-Hoc Routing. *ACM Mobicom*, August 2000.

[21] B. Lyles. Personal communication, August 2001.

[22] D. Moore et.al. Where in the World is netgeo.caida.org? *INET 2000*, June 2000.

[23] J. Moy. OSPF Version 2. *RFC-2328, IETF*, April 1998.

[24] V. N. Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. *ACM SIGCOMM*, August 2001.

[25] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking*, Vol.5, No.5, pp. 601-615, October 1997.

[26] V. Paxson. Measurements and Analysis of End-to-End Internet Dynamics. Ph.D. dissertation, UC Berkeley, 1997. *ftp://ftp.ee.lbl.gov/papers/vp-thesis/dis.ps.gz*

[27] R. Periakaruppan, E. Nemeth. GTrace – A Graphical Traceroute Tool. *Usenix LISA*, Nov 1999.

[28] U. Raz. How to find a host's geographic location. *http://www.private.org.il/IP2geo.html*

[29] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). *RFC-1771, IETF*, March 1995.

[30] C. Semeria. Traffic Engineering for the New Public Network. Juniper Networks Whitepaper, September 2000.

[31] S. Savage, A. Collins, E. Hoffman, J. Snell and T. Anderson. The End-to-end Effects of Internet Path Selection, *ACM SIGCOMM*, pp 289-299, September, 1999.

[32] L.Subramanian, V. N. Padmanabhan and R. H. Katz. Geographic Properties of Internet Routing. To appear in *USENIX Annual Technical Conference*, Monterey, CA, June 2002.

[33] D. C. Vixie, P. Goodwin and T. Dickinson. A Means for Expressing Location Information in the Domain Name System, *RFC-1876, IETF*, January 1996.

[34] Akamai Inc. *http://www.akamai.com/*

[35] America Online (AOL). *http://www.aol.com/*

[36] BBNPlanet publically available route server, *telnet://ner-routes.bbnplanet.net*.

[37] bCentral. *http://www.bcentral.com/*

[38] Digital Island Inc. *http://www.digitalisland.com/*

[39] DoubleClick, *http://www.doubleclick.com/*

[40] http://geography.about.com/

[41] Hotmail. *http://www.hotmail.com/*

[42] Internet2. *http://www.internet2.org/*

[43] Internet Traffic Archive. http://ita.ee.lbl.gov/

[44] IP to Latitude/Longitude Server, University of Illinois *http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll*

[45] List of Airport Codes. *http://www.mapping.com/airportcodes.html*

[46] List of Public Libraries in the U.S. *http://sunsite.berkeley.edu/Libweb*

[47] List of Public Traceroute Servers *http://www.traceroute.org/*

[48] List of Web servers in Europe *http://pauli.uni-muenster.de/w3world/Europe.html*

[49] MapNet: Macroscopic Internet Visualization and Measurement. *http://www.caida.org/tools/visualization/mapnet/*

[50] Matrix.Net, *http://www.matrix.net*

[51] MERIT Network, *http://www.merit.edu/*

[52] NeoTrace, A Graphical Traceroute Tool *http://www.neoworx.com/products/neotrace/default.asp*

[53] NPD-Routes data set, Internet Traffic Archive. *http://ita.ee.lbl.gov/html/contrib/NPD-Routes.html*

[54] Traceroute data used in this paper. *http://sahara.cs.berkeley.edu/rawtraces*

[55] Skitter project at CAIDA. *http://www.caida.org/tools/measurement/skitter/*

[56] U.S. Gazetteer, U.S. Census Bureau, *http://www.census.gov/cgi-bin/gazetteer*.

[57] vBNS: very high performance Backbone Network Service. *http://www.vbns.net/*

[58] VisualRoute, Visualware Inc., *http://www.visualroute.com/*

[59] Yahoo Inc., *http://www.yahoo.com/*