An Investigation into Better Techniques for Geo-Targeting

by

Shane Elliot Cruz

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author .....

Department of Electrical Engineering and Computer Science
May 17, 2002

Certified by .....

Brian K. Smith
Associate Professor, MIT Media Laboratory
Thesis Supervisor

Accepted by .....

Arthur C. Smith
Chairman, Department Committee on Graduate Students

An Investigation into Better Techniques for Geo-Targeting

by

Shane Elliot Cruz

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2002, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis discusses the possibility of determining the geographic location of Internet hosts solely from IP address information. The design, implementation, and evaluation of several unique components for geo-targeting led to an understanding of the strength and weakness of each strategy. Upon realization of the pros and cons of the location mapping designs, a new system was built that combines the best features of each technique. An overall evaluation of each component and the final system is then presented to demonstrate the accuracy gained with the use of a *Domain-Based Component Selection* methodology. These findings demonstrate that location mapping can be used as a useful tool for many common Internet applications.

Thesis Supervisor: Brian K. Smith
Title: Associate Professor, MIT Media Laboratory

Acknowledgments

It is difficult to overstate my gratitude to my advisor, Brian K. Smith, for the assistance he has given me throughout the entire thesis process. Without his open mind and insightful comments, this thesis would not be possible.

I would also like to thank all the members of the Critical Computing group at the MIT Media Lab. Erik, Tara, Anna, Tim, Jeana, and Nell have all provided a great environment in which I was able to pursue my research.

I am deeply indebted to all those who have helped me throughout my studies at MIT. This great circle of close friends helped me through times when I thought I could never make it. I would especially like to thank my roommate, Jang Kim, for being a great friend throughout both my undergraduate and graduate careers. Without his assistance and support, I don't know if my accomplishments would have been possible.

I thank the many others who have stood by me throughout my life. My close friends and family members have given me the motivation to be where I am today.

I would especially like to thank my brothers who have been my best friends throughout my life. Throughout the thick and the thin, I know that you will always be there for me.

Lastly, and most importantly, I would like to thank my parents. I wish I had the ability to express in words how much I owe you. Your love and encouragement have given me the opportunity to experience things that many could never imagine. To them I dedicate this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For many years following its introduction, the computer was used solely as a computational tool to solve complex mathematical and scientific problems at the corporate and university level. As both the price of the hardware and the size of these machines began to drop, computers started to make their way into residential homes as word-processing machines and entertainment devices. More recently, the connectivity presented by the Internet has shown that these devices can be used for much more than solving equations, typing a paper, or playing video games. The rapid growth of the World Wide Web (as seen in Figure A-1) has presented a whole new powerful information source that enables people across the globe to quickly access information that was once impossible to find.

This large expansion of online information and the diminishing cost of connecting to the Internet have created an online community that spans the globe. As computer prices continue to drop and as portable devices continue to gain Internet connectivity, the number of hosts connected to the Internet will only rise. This diverse global community, created by the vast array of networks, allows for international communication across an entirely new medium in a form that was once impossible to imagine. Figure A-2 indicates the exponential climb in Internet hosts over the past several years.

1.1 A Location-Transparent Web

While the Internet continues to provide useful information and services to a global community, the geographic topology of the World Wide Web remains insignificant. Due to geographically-blind packet routing protocols, a website visitor from London, England can access web services from a server in Boston, Massachusetts just as easily a student in the greater Boston area. This desirable location transparency has played a significant role in the flexibility and scalability of the Internet's networks.

Unfortunately, with this location transparency comes an interesting challenge for web developers. Since Internet packet forwarding techniques make use of geography-free routing information (and there is no direct correlation between IP address and geographic location), a server cannot currently determine the location of the client with which it is communicating. Therefore, web developers and service providers must generalize their content and services to appeal to visitors from around the globe. Language selection, advertising, and products displays are all statically chosen to be the default for every online visitor.

In realizing the hindering effect this transparency has on information presentation, recent research has attempted to determine the possibility of overcoming it. Efforts have been made, through various technical strategies, to discover possible methods of resolving the geographic location of Internet hosts solely from the IP address of the machine. These applications typically use a variety of active network services to systematically generate a mapping from IP address to geographic location. This search, for the ideal mapping between numbers and location, is commonly referred to as the *location mapping problem* and *geo-targeting*.

1.2 Research Contribution

This work is a study of the location mapping problem and different methods for solving it. The first step in this process involved the review of the various techniques that are currently used for geo-targeting Internet hosts. Strategies such as obtaining

domain registrant information, analyzing `ping` timing data, parsing the names of nearby network routers, and attempting to geographically cluster network addresses were all researched in this work. In addition, other less commonly used techniques and proposals were studied in order to comprehend the reasons they did not succeed.

After completion of the research review, three unique strategies were selected to be implemented and analyzed. Individual system components were built to implement a domain registrant parser (utilizing WHOIS server queries), a `traceroute` execution analyzer, and a method for geographically clustering Internet hosts based on IP address information. The design and implementation of each component gave insight to the strengths and weaknesses of each individual methodology.

Once the components had been built and preliminary evaluations had been performed, a better solution to the location mapping problem became evident. The use of a new rule-based system that selectively chooses the method used to geo-locate Internet hosts was developed to improve on previous research efforts. In this new application, the top-level domain of the the client being resolved is used as the indicator of component selection. After implementation of this new *Domain-Based Component Selection* methodology, it was apparent that its accuracy outperformed each individual component of the system in many ways.

This research concludes with an evaluation of each component and a comparison of them with the performance of the final system design. These evaluations demonstrate the accuracy gained with a Domain-Based Component Selection algorithm and gives insight to the many possible applications of this technology.

1.3 Thesis Outline

The rest of this thesis will explain the work that was done in this research effort and describe the implementation and evaluation of the systems that were built. Chapter 2 demonstrates several real-world applications of the location mapping technology and discusses the economic and social effects it could have on the Internet community. In Chapter 3, the motivations behind these studies and the various related research

efforts are discussed. Chapter 4 explains the design and implementation of each technique to be evaluated and presents the new system which implements the domain-based component selection methodology. After the discussion of the system design, Chapter 5 presents an analysis of these various system components and compares their performance to that of the final application created in this work. To end the document, Chapter 6 reviews the findings of this research and discusses the true capabilities of the location mapping technology.

Chapter 2

Extended Example

In order to understand the motivations that inspired this research, it is important to look at several ways in which this technology could be useful to the Internet community. This chapter presents a few examples of the many ways in which breaking the location-transparency barrier presented by the World Wide Web could benefit both online web users and also those companies who offer information and services on the Internet.

2.1 Dynamic Webpage Interface

Website visitors come from all over the globe. As previously mentioned in Chapter 1, a company whose server is located in the United States has no guarantee that its visitors are located strictly in the one of the fifty states. One of the greatest features of the Internet is that a person on the other side of the world can just as easily access content from a server as someone who is only a few miles away from it. The large amount of information that is available on the web, in combination with the ease of accessing it, often makes it a much more useful information resource than other options, such as driving down to the local library.

While this location transparency offers great advantages in creating large online communities and also eases the distribution of information, it currently restricts a web developer's ability to deliver useful content. Without knowing the geographic

location of an online visitor, a site can either present very broad and generalized content for any visitor from any country of the world, or it can present data that is useful to only a small subset of the site's possible visitors.

Fortunately, with the use of location mapping techniques, companies can begin to provide a more personalized service whose content can be dynamically adjusted based on the geographic location of the host. With the knowledge of where a visitor is likely to be located, a site can now adjust its content to a state that is most useful to the client.

2.1.1 Country-Specific Customization

The world is a very diverse community. The thousands of different languages and dialects spoken around the globe make it impossible for any person to learn to understand them all. Unfortunately for web developers, the Internet is composed of this diverse global community. This means that any given visitor to a site could potentially be from any country around the world, and may only speak any one of the thousands of different languages. How then, can the site present information that is useful to every possible visitor?

Fortunately, location mapping techniques can help site managers deliver content that will most likely be understandable to the visitors whom they select as a target audience. In many cases, knowledge of the visitor's country narrows language choices down to a select few. For example, it is a fairly safe assumption that a visitor from China can speak Chinese and a visitor from France can speak French. It is therefore possible to narrow the scope of possible languages and other country-specific details by determining the country in which the current visitor is located.

From this example, it is apparent that a solution to the location mapping problem can have a substantial impact on the presentation of information on the web. Just by accurately determining a visitor's country of origin, the site can specifically tailor its language, currency, metrics, and news to a format that will be understandable to every unique visitor. This technique of dynamic content customization would allow sites to reach a much larger set of the global community in a much more personalized

manner.

2.1.2 Geo-targeted Sales and Advertising

The dynamic, location-based customization of web content could also allow advertisers to be more specific in their marketing techniques. Often times, online advertisements are too general to attract the attention of the online community. For example, a person from Boston, Massachusetts may not be attracted to a simple airline advertisement for *Fare Sales from "Select Cities" to the Caribbean.* On the other hand, an advertisement that specifically mentions *Fare Sales from "New England" to the Caribbean* or even more specifically from *"Boston" to the Caribbean*, might be just enough to catch the attention of the visitor. Not only would these geo-targeted ads increase revenue by achieving better results, they also make the advertisements more tolerable for the customers.

The location of the web visitor can also help to filter the various products that on online retailer may want to highlight on their site. For example, a large clothing company that sells its products online may want to vary the front page of its site based on the current weather in the customer's region. It is most likely that a person in southern Florida has no interest in the winter jacket specials that are important to customers from Wisconsin. By dynamically adjusting the site's content to be more relevant to individual customer desires, a company can increase sales and improve customer satisfaction.

2.2 Regulatory Compliance

Just as important as improving customer satisfaction and increasing corporate revenue, is ensuring that web sites comply with local laws and regulations. It is often the case that different countries, and even different states, have various rules that govern online commerce. In order for corporations to continue to do successful business in various parts of the world, it is critical that they do everything possible to comply with each of the specific local regulations. Another important consideration is the

opportunity for companies to use geographic information to hinder online fraud and to allow for location-based selective services.

2.2.1 Location-Restricted Sales and Services

In a recent law suit, the French court ordered Internet powerhouse Yahoo! to bar French users from visiting Yahoo! auction sites where Nazi memorabilia is sold[9]. According to the judge, selling or displaying any items that incite racism is illegal in France and therefore these sites are against the law. While the executives at Yahoo! argue that they should not have to comply with a non-US law, the court has remained unwilling to reverse its decision. This ruling leaves a company such as Yahoo! with only two choices: either remove the auctions for this memorabilia from everyone's view, or find a way to determine which customers are visiting the site from France.

While the Yahoo! court case has been the spark around the issues of solving the location mapping problem, it is not the only case in which this technology could prove useful. In another case, internationally recognized casinos have questioned the ability to provide access to online gambling and wagering sites while adhering to regional, national, and international laws[17]. If it is possible to determine the geographic location of web customers, these types of services could be selectively distributed to customers who live in legal regions.

2.2.2 Fraud Detection

While restricting product sales and services based on geographic location can prove useful, it is also important to attempt to limit the thousands of cases of online fraud that happen every year. In 2001, the amount of reported money lost from Internet fraud was $4,371,724, up from $3,387,530 in 2000. The average loss per person had also risen to $636 from $427[33]. With scams ranging from abuse of online auctions to scams involving Nigerian money offers, the large economic impact they have on our community make it apparent that something must be done to stop these crimes.

There are certain regions of the world that are well known for credit card and other

types of financial fraud. With the ability to determine the location of online clients, location mapping technology can help to indicate possible problems and therefore limit the damage done by online fraud. Similarly, simply matching the credit card billing address with the location of the computer presenting it can help to signal possible fraudulent situations. Although the location mapping technology cannot completely stop online fraud in itself, it could provide helpful services in alleviating some of the damages it causes every year.

2.2.3 Digital Rights Management

In a different kind of online fraud, users do not attempt to steal money from others, but rather obtain electronic information that is illegal for them to have in their possession. For example, strict US law disallows some advanced cryptographic software from being exported to locations outside the United States. Currently, many sites simply ask the user to input their address in order to check if it is legal for them to download the application. Unfortunately, a dishonest person can easily lie about their location in order to get hold of software that the law restricts them from obtaining. By using the location mapping technology to identify the location of the client, it would be possible to enforce digital rights management by restricting these fraudulent situations.

In another example, several music companies have been required by US law to not allow certain songs to be downloaded in the United States. Unfortunately, there is not always a way to enforce these laws in other countries. In order to limit the damage caused by this blatant disregard of copyright policy, companies can at least ensure that users in the fifty states can no longer obtain illegal copies of these songs.

2.3 Business Intelligence

Another important application of the location mapping technology can be to improve sales and services through increased business intelligence. Before the concept of an e-business became a reality, it was always relatively easy to determine the geographic

spread of a customer base because you always knew exactly who your customers were. This is not the case in the Internet world. Unless a company is product-oriented and has the ability to track all customers through detailed shipping information, they become a victim to the location transparency of the web. A company that can determine the approximate location of its customers can improve its services, increase marketing in necessary geographic regions, and create an overall better business.

Recent increases in the traffic on the Internet has led to network congestion during peak hours of usage. In order to combat this undesirable network latency, companies such as Akamai[24] provide services referred to as "content distribution networks" for reducing the network complexities of the Internet. In these systems, a company replicates its web content onto many servers dispersed across the world. The goal of this replication is to reduce the network latency the users experience by increasing a server's proximity to the customer. If a server is physically located close to a customer, then the customer is less likely to be effected by network traffic and will therefore experience less delay[19, 20]. Many companies have seen the advantages of these content distribution networks and have replicated their sites across the world. Unfortunately, many of these servers frequently go unused.

The use of location mapping technology to provide more useful web analytics, can help companies to identify the regions of the world from which most of their hits are coming. This information is vital to creating a useful content distribution network. Instead of arbitrarily placing servers around the globe, location mapping technology can help companies target high-load geographic areas in order to create a more useful and less wasteful network of content replications.

To further the use of these advanced web analytics, companies can identify regions of the world where their marketing may be weak and need improvements. For example, if a company that wants to target the entire United States determines that they are not receiving many online visitors from the Midwest, they can change their marketing strategy to reach these potential customers that are not currently visiting the site. With this useful resource, a company can expand its customer base to include those customers in a previously unreached geographic area.

2.4 Summary

This chapter presented several real-world applications in which the location mapping technology could provide invaluable services. Identifying the geographic location of online customers not only provides web developers with the ability to dynamically customize the appearance of their site for a more personalized interaction, it also introduces many economic benefits through location-specific sales and advertising techniques. To further extend the use of this technology, ideas were presented which demonstrate the ability to improve regulatory compliance with the help of accurate geo-targeting. Finally, this chapter presented methods to improve overall business intelligence and customer service by identifying the geographical dispersion of customers across the globe. As seen in the above sections, the ability to resolve the geographic location of online clients solely through IP address information could change the way business is done on the Internet.

Chapter 3

Theory/Rationale

This chapter presents the underlying motivation for this research, examines some unavoidable limitations, and describes other systems inspiring this project.

3.1 Motivation

As discussed in Chapter 2, location mapping technology could have many important applications in today's Internet economy. From creating more customer-friendly web sites to reducing Internet fraud, an application that provides accurate geo-targeting results could change the way business is done on the Internet.

While there has been research done in this field over the last several years, there has been a limited amount of system evaluation to demonstrate the accuracy of each implementation. Except for the brief analysis presented in [20], no researchers have analyzed the multiple techniques that are used to map IP addresses to geographic locations. Therefore, there is no way to comprehend the extent of their usability for the various potential applications. This work intends to extend previous research to provide a reasonable understanding of the strengths and weaknesses of the various strategies that are currently being used in solving the location mapping problem.

In addition to the research that is available, several corporations have been created that offer solutions to the location mapping problem. With proprietary corporate technologies, these companies claim to provide fast and accurate services that pinpoint

the geographic location of any unique IP address. Unfortunately, the high costs of these systems and the lack of their ability to prove results have led to a limited spread of their usage.

For these reasons, the goals of this research are to determine if it is possible to create a non-invasive solution to the location mapping problem. By systematically developing and evaluating the strengths and weaknesses of various component designs, this research will also present a new method to achieve a mapping from IP address to geographic location whose accuracy surpasses that of current research technology.

3.2 Limitations

Trying to resolve an IP address to a given location is a complex task. Even though proposals have been made[5, 6], there is currently no protocol that indicates a guaranteed geographic location of a host when given only the machine's IP address. The lack of this useful resource has led many people in search of new techniques that may achieve this mapping from numbers to location. As this chapter will describe, the possible dispersion of hosts within a given domain, the wide variance in naming schemes for hosts and corporate routers, and the non-uniformity in network delay have made this task very difficult.

Even if a technique were developed that correctly mapped a given IP address to its respective geographic location, it would still not give a guarantee that the machine making a request is located at that site. In order to target the exact location of any host on the Internet, there must be only one location for every machine with the same 32-bit number. Unfortunately, the limited number of available IP addresses, the need to increase internal network security, and the desire to make network administration easier and more flexible have led to the increasing popularity in creating internal networks hidden behind machines that implement network address translation[8].

The primary purpose of a machine performing network address translation, such as a proxy, is to act as the middle-man between an internal network and the rest of the Internet. In this scenario, the proxy is assigned one IP address to connect

to the Internet and every request coming from the internal network appears to have come from the proxy. The proxy is then in charge of multiplexing communication between hosts in the internal network and those in the outside world. This setup allows for many computers to connect to the Internet with only one IP address and is advantageous in terms of security and network administration.

While this use of address translation in proxies and firewalls is advantageous in many ways, it places restrictions on the ability to determine the geographic location of a host from a given IP address. Unfortunately, with the use of network address translation, the outside world is only presented with the IP address of the proxy, and not of the host actually making the request. In many cases, this is not an issue since proxies are typically in close proximity to those hosts connected to them. In other situations, such as with the AOL network[25], machines from all over the United States are connected to proxies that are all located in one geographic region (Virginia). Situations such as these make it not only difficult, but impossible, to create a perfectly accurate mapping from a host's request to the location from where the request was originally generated.

3.3 Previous Research

The first step of this research was to study and analyze various methodologies that have been (and currently are being) used to determine geographic location of an IP address. This section provides a brief summary of existing systems and how they attempt to solve the location mapping problem.

3.3.1 Exhaustive Tabulation of Mapping

In realizing the difficulties of successfully mapping IP addresses to geographic location with scientific techniques, several researchers have turned to a different method for solving the location mapping problem. Using brute-force methodology, these systems simply attempt to create an exhaustive tabulation of all IP addresses and their respective geographic location. The idea is that the best way to determine the geo-

graphic location of a specific IP address is to simply ask the user one time for this data, and then to store it for future reference.

The problem with this scheme is immediately obvious. In order to correctly map a host to its corresponding location, it is first necessary that the respective host manually inputs this information. This data can then either be stored on the user's machine in a cookie, or it can be entered into a database on the server. In either case, there are many problems with this strategy.

Unfortunately, this scheme can be burdensome on the user if their location is not already stored in the database. Another problem with this technique is that the user can intentionally input incorrect information in order to trick the system. For instance, if a website wants to limit its services to hosts in a particular geographic region, a malicious person could enter incorrect location information for themselves in order to gain access to these services.

Once an accurate tabulation of the IP address space to geographic location is achieved, it would be a fast and practical solution to the location mapping problem. Unfortunately, building this database with correct information is not as easy as it first appears. This scheme for location resolution would best be implemented as a caching scheme on a more scientific approach.

3.3.2 DNS-Encoded Information

Since 1996, many proposals have been made to include geographic information in the domain name system (DNS). The idea was to include a new Resource Record (RR) in the DNS (called the LOC record) that indicates the latitude, longitude, and altitude of hosts on the Internet[5]. It would then be possible to solve the location mapping problem by simply performing a DNS lookup for a given IP address and parsing the LOC resource record.

At first glance, this seems like a useful feature to add to the domain name system. Unfortunately, it is not as easily implemented as it first appears. In order for this protocol to be successful, every network administrator would have to manually update the DNS information for every host in their network to include this geographic

information. Even if the information was finally added to the DNS, it would require updates every time the location of a host changed. With the rapid increase in mobile computing and wireless devices, this would be a difficult task that would lead to stale data in the domain name system.

Another problematic drawback with adding location information to the domain name system is the load that would be added to the DNS root nameservers. Each location resolution would require a query to the nameserver in order to determine the location of the host. If a popular website, such as The Weather Channel[35] or the New York Times[34], wanted to use the LOC record of the domain name system for every request to their website, the nameservers could quickly become overloaded and network delays could rapidly increase.

Currently, only a small percentage of hosts on the Internet have the geographic location information entered in the domain name system[6]. If trends continue in the current direction, it is safe to assume that DNS-encoded information will not be a reliable source for solving the location mapping problem anytime in the near future.

3.3.3 Ping Timing Information

A relatively coarse-grained method for location resolution has been presented by researchers at the Microsoft Research Center. In this solution to the location mapping scheme, which they refer to as GeoPing[20], network delay is used as an indicator of a host's geographic location. In this technique, the `ping`[18] utility is used to calculate network delay times from various locations around the United States. These `ping` probes are located at predetermined sites in which the geographic location of each machine is known.

In order for the GeoPing technique to work, a set of training data is needed to generate a delay map. This training data consists of a list of IP addresses and their known geographic location. Once this data set is obtained, the ping probes are used to generate the delay map which indicates each location in the training data and the corresponding network delay from each probe machine to that location.

Following from previous research in the area of locating hosts in wireless LANs[2],

GeoPing determines that hosts with similar network delays, with respect to other fixed hosts, tend to be located near each other. In this *nearest neighbor in delay space (NNDS)* technique, the location of a given host is then inferred by determining which host from the training set most closely resembles the delay of the host being tested.

Contrary to prior beliefs[3], this technique demonstrated a correlation between network delay and geographic distance. Due to the rapid growth of the Internet and the presence of much better connectivity, the effects of router bottlenecks and circuitous routing can be overcome by observing multiple delay times from each ping probe.

Even though the GeoPing research has demonstrated that network delay can give an estimate of geographic location, the experimental results have shown that the median error distance in this scheme was around 382 kilometers. Furthermore, the training data and test data in this study were all hosts that are located on well-connected university campuses. Since these hosts are all part of the Internet2 project[29], the network congestion between each host should be minimal. Unfortunately, the GeoPing technique would not be as effective when used in situations in which timings span both high bandwidth and low bandwidth links in the Internet.

3.3.4 Whois

Another method used to determine the location of hosts on the Internet is done by querying WHOIS servers and determining location information from the response. A WHOIS server is a tool that allows users to query for information about domain name registration records for a specified domain. A user can ask a server for information about a domain (such as `MIT.EDU`), and in return they will get a listing of information such as the company who registered the domain (and its address), the technical contact and billing contact for the domain, the domain nameservers, and the dates the record was last updated. From the various addresses returned from the query, these systems infer that any host in that domain is located near the location specified by these addresses.

This method was first introduced by researchers at the University of Illinois as part of Avatar[15], a virtual reality system for analysis and mapping of WWW server requests to location. The system cached frequent WHOIS queries in order to limit the load on the remote servers.

Another similar system was developed by the Cooperative Association for Internet Data Analysis (CAIDA). In the NetGeo project[10], a collection of perl scripts are used to query remote WHOIS servers in order to determine the registrant's zip code for the domain of the IP address. Once this information is parsed from the query, NetGeo uses a local database to map the zip code to its respective latitude and longitude. As in the Avatar system, NetGeo caches its WHOIS lookups in order to reduce the load of the remote servers and the unnecessary network traffic.

In many cases, the location of the domain's registrant gives a good indication of the locations of the hosts within that domain. For universities and small corporations, the hosts are typically all geographically clustered in one centralized location. Unfortunately, this scheme does not always work. A company whose headquarters is located in New York City might generate an inaccurate prediction that all hosts in their domain are located in New York, when in fact they may be located all over the globe. Also, the large growth in residential broadband connectivity has given homes across the country IP addresses in the domain of large corporations, such as AT&T, therefore making it appear that every host is located in Bridgewater, NJ. This fundamental limitation of the WHOIS querying technique has proven that it can only be successful for a subset of the possible hosts connected to the Internet, and therefore does not provide satisfactory results in solving the location mapping problem.

3.3.5 Traceroute

Many recent systems have been developed in an attempt to solve the location mapping problem by analyzing hostnames of routers close to the machine that is being located[11, 21, 37]. Often times, these backbone routers are named with hints as to their geographic location. For example, a router located in Boston, Massachusetts may have the name:

```
181.ATM6-0.XR1.BOS1.ALTER.NET
```

Upon looking at this host name, the `BOS` code makes it pretty clear that this router is most likely located in the Boston area. As in the above case, many companies use the airport codes of the closest city or a city abbreviation in their router naming schemes in order to facilitate network administration. These embedded codes and keywords are the essential element needed to determine a host's location in this location mapping scheme.

In order to locate nearby routers, these systems use the `traceroute`[13] utility. By running a traceroute from a given server, these applications find the last few hops to the designated host. Once these last hops are identified, the systems attempt to parse these router names and hope to find a keyword identifying its location.

Unfortunately, different companies have different naming conventions for their routers. Therefore, one company may refer to a router in Chicago, IL by using the airport code for O'Hare International Airport (`ORD`), another may refer to it by the code for Midway International Airport (`MDW`), another may use the generic `CHI` code, and a different company may use a code such as `chgil` to refer to its location. In order to overcome the vast differences in the naming schemes, these systems require special parse files that regulate how parsing is done depending on the domain in which the router exists. Once the code or keyword is extracted from the router name, a database is used to map the codes to their respective latitude and longitude.

Although these systems often produce better results than the WHOIS querying technique for large, dispersed corporations, they are also plagued with many weaknesses. The most obvious is the necessity to keep accurate parse files for every possible backbone domain. The rapid growth of the Internet has led to an increase in the number of companies who maintain these routers, and therefore a large number of different naming conventions. Even if the system correctly parses the name of the closest router to the host, this router may not be in close proximity to the machine. Often times, these backbone routers are only located in large cities, so they may incorrectly infer that host is located in Boston when it really may be a few hundred miles away in Maine.

Another inherent weakness of this system is the time it takes to geo-locate a host. The `traceroute` utility requires several messages to be sent out in order to calculate the route a packet takes from one server to another. Therefore, these systems are limited by network delay and waiting times to complete the traceroutes. In order to decrease the delay time, a caching scheme can be implemented. Unfortunately, the large number of hosts on the Internet (as seen in Figure A-2) makes it nearly impossible to cache data for every unique `traceroute`.

3.3.6 Cluster Analysis

Within the same Microsoft research effort as the ping timing analysis, a technique was created that is referred to as GeoCluster[20]. In the GeoCluster solution, network routing information (obtained from a backbone router) is combined with location information about a training set of hosts in order to infer the location of a given host. The concept involves clustering IP addresses based on a specified-length address prefix, and assuming that all IP addresses in that cluster are geographically bound within a certain area.

When a router receives a packet on the Internet, it needs to decide which physical link to use in order to transmit the message along. To do this, the router must maintain a table that maps IP addresses to the physical links that will eventually lead to the proper destination. Since the 32-bit address space of IP addresses can specify over four billion different addresses, the routers must not individually list each IP address, for the tables would be far too large.

The Border Gateway Protocol (BGP) is the solution to this problem[22]. BGP is an inter-Autonomous System routing protocol. An *autonomous system (AS)* is defined as a list of hosts that are all under the same administrative domain. By grouping hosts into these autonomous systems, the routing tables can be limited to only contain each individual AS and not every possible IP address. These systems can be specified by an *address prefix (AP)*. For example, the routes for all hosts at the Massachusetts Institute of Technology can be maintained in the Internet routing tables as an aggregate prefix such as `18.0.0.0/8`. This notation means that any IP

address whose first eight bits match the first eight bits of the IP address in the address prefix (in this case, 18), is part of this autonomous system. Therefore, `18.85.23.12` and `18.10.44.201` are both part of the same autonomous system. With this notation, one entry in the routing table can specify the direction in which to send incoming packets for over 16 million hosts.

The use of these address prefixes to topologically cluster machines was first done in [14]. In [20], efforts were made to see if these prefixes not only specified topological clusters, but also gave an indication of geographic clustering. In the case of a university campus or a local ISP, the autonomous system typically represents a geographic cluster because all hosts are in close proximity to each other. Other larger ISPs, often with nationwide coverage, may only supply a limited number of address prefixes for reasons of scalability.

The implementation of this location mapping scheme requires a large set of training data that, with a high certainty, maps IP addresses to their actual geographic location. Then, once a list of possible address prefixes is obtained from a backbone router, the system can attempt to determine if each prefix actually does represent a geographic cluster by checking the training data. Once the list of geographic clusters and their locations have been determined, any new host who has an address prefix that matches one in the cluster list can be assumed to be located close to the rest of the cluster.

The GeoCluster technique has proven to be more accurate than using `traceroute` information or using network delay measurements when supplied with a large set of training data[20]. This scheme also has the attractive feature that it does not need to generate any network traffic in order to resolve a host's location. Unfortunately, the seed data for the application can be very difficult to obtain. In order to successfully cluster the millions of possible address prefixes, it is necessary to obtain a training set that consists of several million hosts and each of their respective geographic locations.

3.4 Proprietary Corporate Strategies

Along with the research presented above, there are also many companies that have been created to provide this location mapping service. Unfortunately, it is not possible to study the techniques used in these systems, as their applications are strictly confidential and proprietary. It can be assumed that their systems most likely use a combination of the above techniques, in association with a large caching database in order to provide fast and accurate results. A list of some of the companies offering these solutions and their product names can be found below:

- Akamai Technologies, Inc. - EdgeScape DB[24]
- Digital Envoy - NetAcuity[26]
- Geo-Bytes, Inc. - NetWorldMap[27]
- Infosplit - NetLocator[28]
- Ixia Corporation - IxMapper[30]
- NetGeo, Inc. - AcuProfile[31]
- Quova, Inc. - GeoPoint 4.0[32]

Chapter 4

Design and Implementation

After careful review of several of the aforementioned techniques for solving the location mapping problem, a few methodologies that appeared to provide the best accuracy were selected to be analyzed. Separate components were created for each technique in order to evaluate their individual success rates on a testbed of information. Upon implementing these components and analyzing the results of those tests, it was possible to determine the strengths and weaknesses of each methodology and create a new system that combines the best features of each individual component. This chapter describes the design behind each of the system components and then describes the final system which combines implementation decisions drawn from each strategy.

4.1 Component Design and Implementation

The components of the system were designed to build on work mentioned in Chapter 3. By focusing on three or four individual methodologies, it was possible to analyze and understand the strengths and weaknesses of their respective designs. This section describes the underlying technologies and services used by each component in order to provide a better understanding of the design and implementation of the final system described later in the chapter.

4.1.1 Whois Parsing Component

The first component to be implemented and analyzed was based on querying WHOIS servers, as previously done in [10] and [15].

Whois Overview

The WHOIS protocol[12] specifies how TCP transaction-based query/response servers provide net-wide directory service to Internet users. The main usage of these servers has been to perform network lookups to determine the availability and ownership of a second-level domain name (e.g. `MIT.EDU`). A second-level domain name is simply a string of characters, followed by a period, which is then followed by a top-level domain. These domains are usually purchased by organizations or individuals to provide a human-readable hostname for their individual web servers.

In order to determine the availability of a second-level domain, the client can query either a registry or a registrar. A registry is an organization that has been given the exclusive right to manage and distribute domain names within a given top-level domain. A registrar, on the other hand, is any organization that manages the distribution of domain names to customers. Therefore, registrars are typically responsible for managing a subset of a given registry's domain names and ensuring that customers keep this directory information up to date. A registry often distributes the domain names within its specified top-level domain across several registrars.

Unless the specific registrar responsible for the requested domain name is known, WHOIS queries must be directed to a universal registry in a order to obtain registrar ownership information regarding the requested second-level domain name. The registry will then respond with information regarding what registrar is responsible for managing and maintaining that domain name. A client to registry interaction can be seen in Figure 4-1.

As seen in the figure, the response from the registry indicates which registrar is responsible for the directory information for the specified second-level domain. Also provided is the server that the client should contact in order to retrieve this directory

```
% whois -h whois.nsiregistry.com villascaribe.com
[whois.nsiregistry.com]

Whois Server Version 1.3

Domain names in the .com, .net, and .org domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: VILLASCARIBE.COM
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NS1.ICCAMERICAN.COM
   Name Server: NS2.ICCAMERICAN.COM
   Updated Date: 03-dec-2001


>>> Last update of whois database: Sat, 27 Apr 2002 16:50:06 EDT <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and
Registrars.
```

Figure 4-1: A Sample Interaction Between a Client and the VeriSign Registry

information. Once the client has received this response from the registry, it can forward the request to the registrar's server in the same manner it contacted the registry. The registrar will continue the query/response protocol by replying to the client with domain registrant information as seen in Figure 4-2.

Useful Information

The response from the registrar in Figure 4-2 shows a typical server response to a WHOIS query. The server tells the client the contact information for the domain, the date on which the record was last updated, and the primary nameservers responsible for the requested domain. For the implementation of the WHOIS component, the most important information in this response comes from the address of the registrant for the domain.

The theory behind the design of the WHOIS component of the location mapping application is that all clients in a specified domain are likely to be located in close proximity to the address at which the domain is registered. Therefore, obtaining the address and zip code of the registrant for any domain should give a good indication of the location of any IP address under that domain. An example of this theory is the assumption that the machines with hostnames `bitsy.mit.edu` and `www.media.mit.edu`

```
% whois -h whois.networksolutions.com villascaribe.com
[whois.networksolutions.com]

Registrant:
Villas Caribe (VILLASCARIBE-DOM)
   5656 S. Waco Ct.
   Aurora, CO 80015
   US

   Domain Name: VILLASCARIBE.COM

   Administrative Contact:
      Gibson, Janis  (JG17611)          janis@VILLASCARIBE.COM
      Villas Caribe
      5656 S. Waco Ct.
      Aurora , CO 80015
      303-680-3100 (FAX) 303-680-3900
   Technical Contact:
      Dingess, Richard  (RD46)          rkd@RMI.NET
      Rocky Mountain Internet Inc
      2860 South Circle Drive, Ste. 2202
      Colorado Springs, CO 80906
      (719) 576-6845

   Record expires on 02-May-2006.
   Record created on 01-May-1996.
   Database last updated on 28-Apr-2002 12:29:18 EDT.

   Domain servers in listed order:

   NS1.ICCAMERICAN.COM          66.118.148.2
   NS2.ICCAMERICAN.COM          66.118.148.3
```

Figure 4-2: A Sample Interaction Between a Client and a Registrar

are both concluded to be located in Cambridge, Massachusetts (and in the 02139 zip code) because that is where the `MIT.EDU` domain is registered.

From this reasoning, determining the geographic location of a given IP address, can be done with the following steps:

1. Convert the IP address into its respective hostname
2. Determine the host's second-level domain by taking the last two tokens of the hostname when tokenized by periods
3. Query a global registry to determine the registrar responsible for that domain
4. Query the given registrar to obtain the domain registrant's address information
5. Assume the location of the machine with the given IP address is in close proximity to the address of the registrant determined in Step 4

Limitations

The WHOIS component of the location mapping application provides a relatively easy and straightforward method for solving the location mapping problem. With a few simple network queries, it is possible to map any IP address to its respective geographic location. Unfortunately, this methodology can yield inaccurate results in certain situations.

While it is often the case that universities and smaller corporations are all geographically clustered in one central location, this is not the case for every second-level domain name. For example, a large company (such as the Microsoft Corporation) may be headquartered in one city (Redmond, WA), and therefore the domain is registered to that location. Unfortunately, Microsoft has offices all over the globe. Therefore, a Microsoft IP address that is really located in New York City may inaccurately be determined to be located in the State of Washington. Even worse, a machine in the `MICROSOFT.COM` domain that is located in Asia will also appear to be on the other side of the globe in Redmond, WA.

In addition to the inaccurate results produced for geographically dispersed entities, the WHOIS component is also plagued by the weakness of a strong dependence on a sometimes unreliable network service. The success of this component relies on the stability of the global registry server and also on the respective registrar's server. Not only is it important that these machines be reachable to the application, but they must also provide data that is not stale and inaccurate. This reliance on a network service also introduces delay to the performance of the application based on network latency between the client and the servers.

4.1.2 Traceroute Hostname Analysis

The next component in the location mapping application is based on parsing the hostnames of nearby network routers as done in [11], [20], [21], and [37]. The underlying assumption in this technique is that routers that are located near a host in terms of network hops are also located geographically close to the host.

Traceroute Overview

The `traceroute` tool[13] is a system that was built upon the technology of the `ping` utility[18]. The main purpose of this application is to determine the steps that an Internet packet takes to get from a source to a specified destination. Therefore, a `traceroute` between two machines (A and B) will produce the IP addresses and the hostnames of the machines that the packet visited on the route from A to B over the Internet. An example of an execution of the `traceroute` utility from a machine at MIT to Columbia University's web server can be seen in Figure 4-3.

As shown in the figure, the `traceroute` output lists each step the packet takes to get from the MIT machine to the machine at Columbia. Each step in this process is referred to as a *hop*. For each hop, the utility gives the hostname of the router it reached, the router's IP address, and information regarding the time it took to reach that hop in the route from the source to the destination.

The `traceroute` tool has proven to have many uses for system administrators

```
% traceroute -s a-bomb.media.mit.edu www.columbia.edu
traceroute to www.columbia.edu (128.59.59.84) from a-bomb.media.mit.edu
 1  passport-4-3 (18.85.23.1)  0.997 ms  0.926 ms  0.909 ms
 2  lexus-4-1 (18.85.3.97)  0.324 ms  0.285 ms  0.293 ms
 3  amtgw (18.85.0.1)  2.617 ms  1.851 ms  4.243 ms
 4  EXTERNAL-RTR-2-BACKBONE.MIT.EDU (18.168.0.27)  2.648 ms  6.244 ms  1.939 ms
 5  192.5.89.89 (192.5.89.89)  5.667 ms  3.737 ms  8.206 ms
 6  ABILENE-GIGAPOPNE.NOX.ORG (192.5.89.102)  7.768 ms  7.118 ms  13.573 ms
 7  nyc-m20-abilene-nycm.nysernet.net (199.109.5.1)  11.166 ms  23.432 ms  25.042 ms
 8  columbia-nyc-m20.nysernet.net (199.109.5.5)  22.969 ms  29.062 ms  14.553 ms
 9  cc-edge-3.net.columbia.edu (128.59.1.71)  23.031 ms *  9.441 ms
```

Figure 4-3: A Simple Traceroute Execution

over the years. By analyzing the routes that Internet packets take from one machine to another, it is possible to find network bottlenecks, circuitous routing, and even possible broken links in network paths. More recently, this useful network tool has been used to help visualize geographic Internet routing and to help minimize the location transparency of the World Wide Web.

Analyzing Nearby Router Names

The traceroute component of the location mapping application only makes use of the router hostnames provided by the `traceroute` application. Even though there is currently no standard naming convention for Internet routers, companies often embed location information in router hostnames in order to simplify network administration. As seen in Figure 4-3, the code `nyc` is used in a few of the router names to indicate that these machines are located in New York City.

The traceroute component takes advantage of these common naming schemes in order to determine the geographic location of routers near the destination IP address. Starting with the hostname of the IP address it is given, and working in the reverse direction of the `traceroute`, it iteratively attempts to extract location information from each of these router names until it finds a match. Once the location of a router is determined, the traceroute component assumes that the specified IP address is located in close proximity to that router. In the example in Figure 4-3, the application would parse the router named `columbia-nyc-m20.nysernet.net` and correctly identify that it is located in the city of New York. Since this router is only one hop away from the destination IP address, it would then correctly deduce that Columbia University's

Table 4.1: Router Names with Common Location Codes

Router Name	Code	Location
wdc-core-01.inet.qwest.net	wdc	Washington, DC
so-4-2-0.bstnma1-nbr1.bbnplanet.net	bstnma	Boston, MA
dist-01-so-0-0-0.hsto.twtelecom.net	hsto	Houston, TX
so-0-1-0.mp1.Orlando1.Level3.net	Orlando	Orlando, FL
so-7-0-0.sttlwa2-br1.bbnplanet.net	sttlwa	Seattle, WA

web server is also located in the same city.

Samples of other router names with common location information embedded in them can be seen in Table 4.1. As shown in the table, companies often use airport codes, city abbreviations, or even the entire city name as an indication of the geographic location of the router.

Parse Files

Unfortunately, without a standard naming convention, different companies use their own unique codes to indicate the geographic location of routers. Therefore, while one company may use the three-letter airport code of the closest airport to the router, another company may use a code that has no standard meaning. Therefore, in order to parse location information from router hostnames, parsing rules need to be generated to specify how location information is extracted from each unique router name.

As done in [21], routers are classified based on their second-level domain name. Since second-level domain names typically represent one unique corporation, it is most often the case that two routers within a specific domain have similar location codes embedded in their names. As seen in Table 4.1, Genuity (second-level domain BBNPLANET.NET) uses a common strategy for labeling each of their routers. In their naming scheme, the second token of the router name contains a six-letter city abbreviation (followed by a router number for that city) and then additional information is separated by a hyphen. These company-specific naming conventions make it necessary to extract location-specific details from hostnames in a different manner for each unique second-level domain.

Fortunately, much of the routing on the Internet is handled by a small subset of the total number of domains that exist. Since the Internet is essentially a network of networks, these larger corporations and Internet Service Providers are responsible for linking the individual networks of the Internet together. Therefore, in order to obtain location information from `traceroute` executions, only analyzing the router names of machines owned by large companies and major Internet Service Providers still has proven to provide highly accurate results[21].

For each second-level domain that is responsible for a large share of the Internet's routing, a *parse file* was created to specify how to extract location information from their router names. In these files, a series of regular expressions determine which tokens of the hostnames to analyze for city codes, and then specify how to handle the codes that are extracted. If it is known that a particular company uses airport codes to indicate the geographic location of their routers, then the regular expression will indicate that the code found should be looked up in a database keyed on airport codes. On the other hand, if the company uses its own unique city abbreviations, the file will contain other information mapping the city codes to their actual location.

An example of a typical parse file can be seen in Figure 4-4. Any host or router name whose second-level domain is `AMERITECH.NET` while be referred to this file for parsing instructions. The regular expression indicates that the third-to-last token of the hostname should be used as the location code (as indicated by the parentheses). The end of this regular expression then tells the application what to do with the code that has been extracted. The first database token, *this*, tells the application to look at the end of the parse file for codes that are unique to this specific ISP. Therefore, if the hostname is `so-7-0-1.chcgil.ameritech.net`, then the parse file tells the traceroute component that this router is located in Chicago, IL. If the code that is extracted does not match any of the exceptions in the parse file, it is then instructed to search the next database token to locate the code. In the case of this specific parse file, the next database token (*cities.db*) is a reference to a database containing common city abbreviation codes, that are used by many ISPs, and their respective geographic locations.

```
#Regular Expressions
s/.*\.(.+)\.ameritech\.net/$1/this,cities.db

#Data --- Do Not Remove this line
chcgil=chicago,il,us
```

Figure 4-4: A Sample Parse File for the Ameritech Network

Often times, companies adopt several different naming conventions based on the type of the particular router that is being added to the network. These various nomenclatures can lead to parse files that are larger and more complex than the sample in Figure 4-4. A example of a detailed parse file for Genuity routers can be seen in Figure A-3. As seen in this figure, there are several different regular expressions that govern the parsing of the router names. The traceroute component simply attempts to use each of these unique expressions to determine the method for parsing each specific router name. Once this location code is extracted, the database lookups are done in the same manner as the more simple parse files.

Limitations

While the traceroute component often proves to be more accurate than the WHOIS component for larger, dispersed corporations, it also has its own inherent weaknesses. While a longer length `traceroute` will often generate several router names with helpful location codes, this may not always be the case. For example, packets in a `traceroute` whose source and destination are in the same city do not necessarily traverse routers owned by the larger corporations with specific parse files. In this case, the traceroute component will be unable to determine any location information from the router names, and will therefore be unable to determine the location of the host.

Another problem with the traceroute strategy is that the larger, backbone routers are traditionally only placed in large metropolitan areas. Therefore, the accuracy of the geo-targeting results will in turn be dependent on the client's geographic distance from a major city. A host which actually resides in western Massachusetts will often times be determined to be in the Boston area, on the far east side of the state. While this is an important aspect to consider in the selected usage of the traceroute

technique, it is important to realize that it does not have a large impact on the majority of the location mapping applications mentioned in Chapter 2. Since the majority of these systems rely on regional geo-targeting capabilities, a system that can identify hosts to within a few hundred miles would suffice in many applications of this technology.

Upon analyzing the success of the traceroute strategy on a small subset of hosts, it was also determined that the median error distance was relatively large when analyzing hosts in the educational top-level domain (`.EDU` domain). Further analysis revealed that the reason for this decrease in accuracy was that these hosts are often connected to the Internet2 Abilene network[29]. This backbone network was developed by a consortium of universities around the United States to help foster richer connectivity for the academic community. The desire for faster connections led to a decrease in the network hops to cross the country, and therefore a decrease in the number of backbone routers in the network. The fewer number of routers limits the accuracy of the traceroute strategy because the backbone router hostnames that are analyzed are often a few hundred miles away from the actual host.

One of the major limitations of the traceroute component is the need to keep an updated set of domain-specific parse files. While companies do not often change the names of routers in their networks, they do constantly add new ones. In order for the `traceroute` parsing to be successful, it requires the companies to follow their standard naming conventions when adding these new routers to the Internet. A minor change in the method a router is named can lead to the inability to parse its location information. In order to ensure accuracy of the traceroute technique, the application would most likely need a system engineer to constantly analyze the results of the `traceroute` analyses in order to keep the parse files up to date with the current Internet naming conventions.

A final consideration for the usefulness of the traceroute component is focused on its performance, and not the accuracy of its results. Since this component is based on the results of active network measurements, the system must wait for the termination of the `traceroute` in order to parse its hop information. Unfortunately, this process

may take anywhere from five to thirty seconds to return a result. While this delay does not effect the accuracy of the location returned, it must be considered for usage in a high-traffic application. While a caching scheme could help reduce repetitive delay, it is highly impractical to cache the millions of `traceroute` executions that would be necessary in a real-world application.

4.1.3 Determining Geographic Network Clusters

The final major component in the location mapping application is based on determining the existence of geographic network clusters in a similar manner to the work done in [20].

Network Topology and Router Tables

The standard IPv4 32-bit address space can specify unique addresses for over four billion Internet hosts. To make routing matters worse, when a machine attempts to contact another IP address on the Internet, there is no predetermined physical location of where that IP address is located. Therefore, in order to correctly forward these packets over the Internet, routers maintain internal tables that indicate which of its physical outgoing links will eventually get the packet to its correct destination.

Unfortunately, with over four billion possible IP addresses, these tables would quickly become unmanageable if each IP address was mapped to a physical link. In order to reduce the size of these tables, the routers instead only store a mapping from *autonomous systems (AS)* to physical links. IP addresses are assigned by giving commercial or administrative entities control over all IP addresses that begin with a specified-length bitstring. The autonomous systems, therefore, usually correspond to those entities which are responsible for a certain subset of the Internet's IP addresses.

For example, the Massachusetts Institute of Technology has been assigned the Class A network prefix of 18.x.x.x, in which the x's represent any number between 0 and 255. Therefore, any IP address that begins with 18 can automatically be determined to belong to MIT. The university, in this case, represents the autonomous

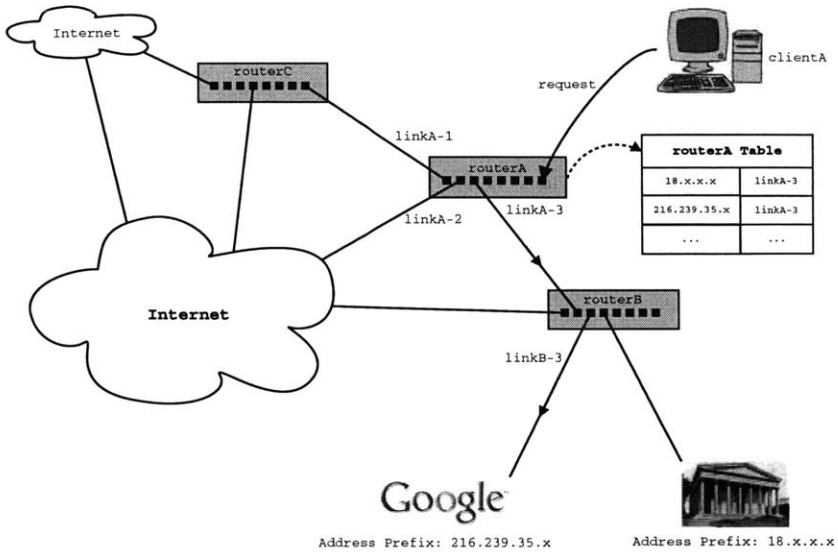

Figure 4-5: A Visualization of Common Internet Routing

system in charge of all IP addresses whose first eight bits are 18 (00010010). Google (another autonomous system), on the other hand, has been assigned the network prefix of 216.239.35.x, and therefore is responsible for any IP address whose first twenty-four bits begin with the first twenty-four bits in their prefix.

This classification of IP addresses into autonomous systems helps to limit the routing knowledge that needs to be maintained in each Internet router. The routers will therefore forward all packets within a specified address prefix in the direction of the autonomous system that is responsible for that prefix.

A high-level visualization of how routing works on the Internet can be seen in Figure 4-5. In this picture, clientA is making a request to retrieve a web page from the Google webserver (216.239.35.100). Since the client does not have an direct connection to this machine, it sends its request to routerA. Upon receiving this packet, routerA must make a decision about the direction it should forward the message in order for it to eventually end up at the Google webserver. The internal table at routerA would correctly maintain information that tells it that any incoming packets destined for the autonomous system with address prefix 216.239.35.x should be sent out along linkA-3. The request is then properly forwarded to routerB, who repeats the process of determining which link should be used to forward the packet to Google. Once

Table 4.2: The Various Network Classes and Their Respective Slashbits

Network	First 8 Bits, b	Slashbits
Class A	$b < 128$	8
Class B	$b \geq 128$ and $b < 192$	16
Class C	$b \geq 192$ and $b < 224$	24
Class D/E	$b > 224$	Not Used

routerB correctly chooses linkB-3, the message is sent to the main Google router which in turn forwards the packet to the webserver's IP address.

Identifying Clusters

The cluster analysis component of the location mapping application makes use of these entity-based routing tables in order to identify geographic clusters in the Internet. Since each router must maintain a table that contains every unique autonomous system, a router table dump from a major backbone router provides a list of every possible address prefix (AP) that make up the networks of the Internet.

Address prefixes in routers are maintained in the following format:

```
188.101.128.0/18
```

In this format, the number following the slash signifies the significant number of bits, called *slashbits*, in the given address prefix. Therefore, the above AP indicates that any IP address whose first eighteen bits match the first eighteen bits of the given prefix, is part of this particular autonomous system. If no slashbits are specified, the number of significant bits is determined by the class of the network address. Network address classes and the length of their specified slashbits can be seen in Table 4.2. As shown in the table, the class of the network is determined by the first eight bits of the IP address.

Once the large set of possible address prefixes is compiled from the router dump, the next task is to attempt to determine the geographic location of each of these autonomous systems, if they are indeed geographically clustered. The concept behind

this technique is to use a similar strategy as the WHOIS parsing component by assuming an IP address that belongs to a particular entity is located geographically close to that entity. The difference between this technique and that of the WHOIS strategy is that the clustering method will correctly identify if a company is not geographically clustered and will not return inaccurate geo-targeting information.

In order to correctly cluster the address prefixes, training data must be gathered that maps IP addresses to their known geographic locations. For the clustering component of this system, data was collected from a popular online weather reporting site. On this site, users enter their zip code in a web form in order to obtain weather information for that location. Since the site is constantly collecting this input data, as well as the IP address of the machine that makes the request, a set of almost two million IP to location data points was collected.

While most people lookup the weather conditions for their current location, it is important to realize that some of these given IP addresses may not map to the correct geographic region (if a user searches for weather in other parts of the world). Therefore, the following algorithm was used to cluster the training data in a way to ensure a correct mapping:

1. Determine the longest-length address prefix for each IP address (in the training data set) by comparing them with the APs gathered from the router dump

2. For each address prefix determined in Step 1, find every IP address from the training data that is part of that AP

3. Check if the set of data points from Step 2 is geographically clustered[1]. If so, add to a database the mapping from the given address prefix to the location in which these elements are clustered

[1]The determination of whether or not a specific set of data points is geographically clustered is application dependent. A coarse-grained use of the location mapping technology may consider IP addresses to be clustered if they are all located in the same country. On the other hand, a more specific application may require all the points to be in the same metropolitan area. For this reason, determination of geographic clustering is not described in detail in this document.

Once the database that maps address prefixes to their respective geographic location has been created, solving the location mapping problem becomes a two-step process. First, determine the IP address's address prefix. Once the AP for the given host is calculated, the machine can be geo-targeted by quickly looking up the location of this prefix in the database created in Step 3.

It is immediately apparent that this technique has definite advantages over the strategy used in the WHOIS parsing component. While they both rely on relating an IP address to the location of the corporate or administrative entity of which it belongs, the clustering technique will correctly identify which of these entities are geographically disperse. Therefore, it will never give the incorrect assumption that an IP address is located far from its actual location.

In addition to this advantage, the clustering technique is also superior to the WHOIS and `traceroute` parsing techniques because it does not rely on any active network measurements. Therefore, the time required to geo-locate a host is only dependent on the brief time it takes to perform the address prefix database query. Unfortunately, as shown on page 56, the clustering strategy also has its limitations.

Subclustering

While identifying geographic network clusters eliminates many of the previous components' inaccuracies, it fails to correctly locate a large set of IP addresses in the Internet domain. Since sizeable corporate entities are often given a large set of IP addresses, and therefore a less-specific address prefix, it is often hard to determine a specific geographic location for the entire autonomous system.

For example, while MIT may be located uniquely in Cambridge, Massachusetts, a company such as General Electric has offices worldwide. In this case, the large company is typically assigned a Class A network address in which only the first eight bits of the IP address are specified. Therefore, an application that attempts to geographically cluster the entire General Electric address prefix will most likely determine that it is not clustered because it has data points all over the globe.

As this realization of non-clustered data points helps to limit inaccurate geo-

targeting, it does no good in helping to solve the location mapping problem. An observation made in [20] proved that large corporations choose to split their corporate networks, in a manner similar to the Internet, in order to simplify internal routing. Therefore, while backbone routers in the outside world view General Electric as 4.x.x.x, the corporate network may partition the address space based on geographic location. While 4.0.0.0/16 may be located at the corporate headquarters in Princeton, New Jersey, another subset of the IP addresses (such as 4.142.11.0/24) may be uniquely located in Los Angeles, California.

In order to take advantage of this corporate partitioning of large address prefixes, the clustering algorithm was changed to detect such possibilities. In this new algorithm, if the system is unable to successfully cluster a specific address prefix, it splits the prefix into two halves and recursively tries again. The new cluster algorithm (with subclustering) is described as follows:

1. Determine the longest-length address prefix for each IP address (in the training data set) by comparing them with the APs gathered from the router dump.

2. For each address prefix determined in Step 1, find every IP address from the training data that is part of that AP.

3. Check if the set of data points from Step 2 is geographically clustered. If the data points are geographically clustered, add the AP and its respective location to the database and go to Step 2 for the next address prefix. If they are not clustered, go to Step 4.

4. Split the address prefix from Step 3 into two parts. This is done by increasing the number of slashbits by one and making each partition have opposite values for that additional bit.

5. Recursively check if each partition created in Step 4 is geographically clustered (Add each partition to the list of address prefixes and continue with Step 2).

The subclustering technique dramatically increases the number of address prefixes that are able to be clustered. Therefore, in addition to limiting the inaccurate location

mappings given by the WHOIS component, the cluster technique has the ability to geo-locate a large set of the IP address space without requiring network activity.

Limitations

While the higher accuracy rate and faster geo-location times of the network clustering component make it attractive, it too is plagued with weaknesses. The most significant of the problems associated with this strategy is the need for a very large set of training data that maps IP addresses to (possibly inaccurate) locations. Even with the millions of mappings that were used in the subcluster application, the cluster component was still only able to recognize around 25,000 unique address prefixes, which represents slightly less than twenty percent of the total number of autonomous systems connected to the Internet. In addition, less than 10,000 of the unique APs were determined to be geographically clustered. Therefore, while this technique provides a high certainty that the results returned are accurate, it is often the case that the location of a specific IP address cannot be determined with this technique alone.

4.2 The Final System

As mentioned throughout the chapter, each individual component of the location mapping application is hindered by flaws in its design. This section describes the design of a new technique that utilizes the best features of each individual component in order to achieve location mapping results superior to the results of each respective application.

In order to better understand the decisions in this section, it is important to re-emphasize the key strengths and weaknesses of each component that was implemented. Table 4.3 lists each of these components and gives a brief highlight of the pros and cons of their respective designs.

Table 4.3: The Strengths and Weaknesses of Each Component Design

Component	Strengths	Weaknesses
WHOIS	Small, Geographically Clustered APs	Large, Disperse APs
Traceroute	Large, Disperse APs near Cities	University APs - Slow
Clustering	All APs - Fast	Limited Results

4.2.1 Domain-Based Component Selection

From Table 4.3, it is apparent that no one technique will provide satisfactory results for every possible unique IP address. The WHOIS parsing strategy is ideal for small, geographically clustered address prefixes, such as a university, but its results for large corporations can be extremely inaccurate. The traceroute component appears to be slightly better than the WHOIS method because it can handle the dispersed entities for those IP addresses that are often failures in the WHOIS strategy. On the downside, this component tends to fail to accurately locate IP addresses that exist in the educational domain and can often take quite long to geo-locate a host. Finally, the geographic clustering of network prefixes appears to solve all these problems with its improved performance and accuracy. Unfortunately, this methodology often fails to correctly cluster a large set of the Internet's available IP address space.

A combination of these techniques is hypothesized to provide overall better results to the location mapping problem. The most important aspect gained from the component analysis was the revelation that *the success of each technique uniquely relies on the domain in which the targeted IP address exists.* In realizing this, the first step in designing the final system was to choose which component to use based on the top-level domain name of the given IP address.

Educational Domains

A brief analysis quickly revealed the strength of the WHOIS component. As will be shown in Chapter 5, the error in accuracy for IP addresses in the educational top-level domain is often less than a few miles when using this technique. Since universities are typically geographically clustered in one central location, and that location is where

the second-level domain name is registered, a quick WHOIS server lookup will most often return the actual geographic address of any IP address in that domain.

This realization determined the first rule of the final system design:

Rule 1: *If the IP address is part of the educational top-level domain, use the WHOIS component to solve the location mapping problem*

International IP Addresses

Due to the limited amount of time to complete the project, it was unfeasible to develop parse files for every international Internet Service Provider. Therefore, the final system design does not attempt to use the traceroute component outside of the fifty states. Fortunately, since the geographic network clustering component does not rely on country-specific information, it is the first step used for international IP addresses:

Rule 2a: *If the IP address is part of a top-level domain outside the United States, use the network clustering component to solve the location mapping problem*

Unfortunately, the training data gathered for the network clustering component did not contain many IP address to location pairs for non-US IP addresses. Therefore, the total number of address prefixes that were able to be clustered outside the United States was very small. When the network clustering failed for international IP addresses, the final system just retrieves the host's country information. This leads to Rule 2b of the final system design:

Rule 2b: *Upon failure of Rule 2a for international IP addresses, simply convert the international country code top-level domain*[2] *into a country name, and return this country as the location of the host.*

[2] *The country code top-level domain (ccTLD) is simply the two-letter country abbreviation that has been assigned to various countries to use as their top-level domain. For example, .jp for Japan and .uk for England*

Non-Educational Domains Within the United States

If the IP address is located inside the United States and is not in the educational domain, the final system again uses a combination of component techniques in order to determine the client's geographic location. Since the network clustering algorithm provides higher accuracy and faster performance, it is the first component used for these US-based, non-educational IP addresses.

Rule 3a: *If the IP address is located inside the United States and is not in the educational top-level domain, use the network clustering algorithm to solve the location mapping problem.*

Again, due to the difficulties in successfully subclustering every possible address prefix, this technique still fails for a subset of the IP addresses based in the United States. Therefore, when network clustering fails, the final system resorts to the `traceroute` parsing algorithm to determine the geographic location of the client machine.

Rule 3b: *Upon failure of Rule 3a, use the traceroute component to determine the geographic location of a client machine.*

A minor modification was made to the traceroute component to increase the accuracy of geolocation. If the `traceroute` arrives at a router name in the educational domain, the application exits the traceroute component and uses the WHOIS strategy to geo-locate that specific router. Since educational router names are often difficult to parse, the system will instead use a more accurate method of determining the correct location of that academic autonomous system with a simple WHOIS server request.

4.2.2 Detecting National Proxies

After the domain-based component selection was in place, there was only one more consideration to improve the accuracy of the location mapping application. As discussed in Section 3.2, some Internet Service Providers still mask all of their customers' IP addresses behind national proxies at one geographic location. Therefore,

while the clustering algorithm will not return an incorrect location for these hosts, the `traceroute` performed in Rule 3b may try to inaccurately geo-locate the proxy address.

In order to combat this misguided geo-targeting information, the final system design correctly identifies these proxy hostnames and does not return the address of the proxy. In order to do this, the system maintains a database of current national proxy addresses for the United States, and the application will not proceed into the traceroute component of the system if a proxy address is found. In this case, the system will simply return empty location information, an indication that the IP address could not be accurately targeted.

4.3 Summary

This chapter described the design and implementation of three separate techniques for determining the geographic location of an online client solely from the host's IP address. A brief analysis of the pros and cons of each system revealed that the success of each strategy is strongly dependent on the host's top-level domain. Therefore, in order to combine the strengths of each design, a rule-based system was implemented to selectively choose the method for geo-location based on the domain of each unique host. The next chapter will present an evaluation of each component implementation and compare these techniques to the final domain-based component selection design.

Chapter 5

Evaluation

This chapter discusses the methods used to evaluate the designs of each system mentioned in Chapter 4. First, it presents the data used to test the accuracy of each component, and the way it was obtained. A discussion of how system performance is determined immediately follows. Then, an analysis of each individual component is given to justify the design decisions made in implementing the final system. To end the chapter, the performance of each component is compared to the final system's performance in order to demonstrate the importance of the domain-based component selection methodology.

5.1 Data Collection

A large set of data, that maps IP addresses to their known geographic location, is needed to analyze the accuracy of each location mapping technique. While it is often possible to estimate the location of a client, it is essential that these data points give extremely accurate indications of the correct locations of these hosts. This section describes the techniques used to gather these data sets in a manner that successfully fulfills these strict requirements.

5.1.1 Research Community Input

The first method of collecting test data was done by polling the research community for their input. Several electronic mailing lists, which often receive traffic regarding the location mapping problem, were sent requests for assistance in order to collect this data. The mailing discussed a summary of the work being done and the reason for needing the help of the research community, in hopes of accumulating several hundred data points.

A website was built to allow users to input the zip code of their current computer's location to assist in the collection of this data. A script then stored the IP address of the user's machine and the zip code which the visitor entered in the web form. Unfortunately, after several months of data collection, only fifty data points were compiled from this technique.

5.1.2 University Server Mapping

Additional data points were obtained by collecting information regarding university web servers. According to the research done in [20], there is a close correlation between the location of a university web server and the location of the university itself. Therefore, data points were next added to the system by mapping the IP address of several university web servers with the corresponding geographic location (zip code) of the university. The information regarding the location of the educational institutions was obtained from the university addresses given on U.S.News & World Report Online[36]. Eighty-five different university web servers from across the country were added to the set of test data with this technique.

5.1.3 Commercial Website Collection

The last technique used to collect accurate location mapping information was based on an analysis of data gathered from the online weather reporting site mentioned in Section 4.1.3. This data, used for the test points, was gathered a month after the data upon which the network clustering component was built in order to limit the

advantages it gives to this component in the system evaluation.

This method produced over two million data points that map IP addresses to possibly inaccurate location information. An analysis was done on this collection to extract those IP addresses whose requests were repeatedly for the same geographic location. These repetitive requests were assumed to be an indication of the likely geographic location of the corresponding host. This data analysis and extraction was done with the following three steps:

1. Extract every IP address from the data set (of possibly inaccurate mappings) that appears at least ten times.

2. From the list of IP addresses obtained in Step 1, identify those which are mapped to the same geographic location at least ninety-nine percent of the time.

3. For every IP address that is ninety-nine percent accurate in Step 2, store the IP address and the corresponding zip code for that address in the database of test data.

This method of data collection provided almost 350 unique IP addresses and a highly accurate prediction of their respective geographic locations. This data, in combination with the other two data collection methods, produced a set of 483 data points that map IP addresses to correct geographic locations. These different collection techniques provided a diverse set of data elements with clients from several different top-level domains. Table 5.1 lists the various top-level domains represented in this data set and the number of hosts from these different TLDs.

5.2 Evaluation Criteria

Once the data to test the system components had been gathered, a well-defined structure for determining accuracy needed to be developed. Since the location mapping technology can be used in a wide range of applications, a generalized form of evaluation was created to analyze the success of these systems. This section describes

Table 5.1: The Top-Level Domains Represented in the Cumulative Test Data

Top-Level Domain	Number of Hosts
.COM	176
.EDU	151
.NET	141
.ORG	9
.GOV	4
.US	2
Total # of Hosts	483

the parameters used to measure each application's performance and the reasons why these factors were chosen.

5.2.1 Determining Performance

The success of an application is often referred to as "system performance". In the context of the location mapping techniques presented, there is no clear-cut definition of what makes one system better than another. While one strategy may provide more accurate location identification for all Internet hosts, it may also be plagued by slow execution times. On the other hand, a system that is able to return a location resolution in a short amount of time may unfortunately return results that are very inaccurate. System dependence, the introduction of network activity, and many other factors can play a significant role in the "performance" of these applications.

While the traceroute component has continuously proven to have slower running times than the other components, caching schemes beyond the scope of this paper could be used to improve its performance. These advanced caches would also eliminate much of the component's unnecessary dependences and need for real-time network activity. Therefore, in the following analysis of the solutions for the location mapping problem, system performance is solely measured by the geo-targeting accuracy of the various techniques.

For this location mapping analysis, accuracy is measured as the distance from the actual host's location to the estimated location returned from the location mapping

solution. This distance is referred to as the *error distance* in the rest of the chapter and will serve as the governing factor in comparing the performance of each location mapping technique.

5.2.2 Limiting Incorrect Targeting

In certain applications, returning an incorrect location can be more harmful than returning no location at all. Unfortunately, the caching schemes mentioned in the previous section do not assist in overcoming this predicament. Therefore, in addition to the accuracy comparisons, this chapter will also present brief performance evaluations of the number of incorrect targetings that are successfully avoided by each component.

5.3 Component Analysis

This section presents an analysis of each individual system component in order to justify the design decisions made when implementing the domain-based component selection of the final system.

5.3.1 Whois Lookup Analysis

As originally predicted, Figure 5-1 demonstrates the superior ability of the whois component when geo-targeting clients in the educational second-level domain. In this analysis, the test data was obtained from the university server mapping discussed in Section 5.1.2. For each web server in the test data, the individual components attempted to locate these clients, and the error distance for each resolution was calculated. The graph in Figure 5-1 displays the percentage of hosts that were accurately located within the given range of error distances.

As seen in the graph, the whois component correctly locates eighty-three percent of the web servers to their exact zip code and ninety-nine percent of them in under a fifteen mile radius. The centralized clustering of these educational institutions

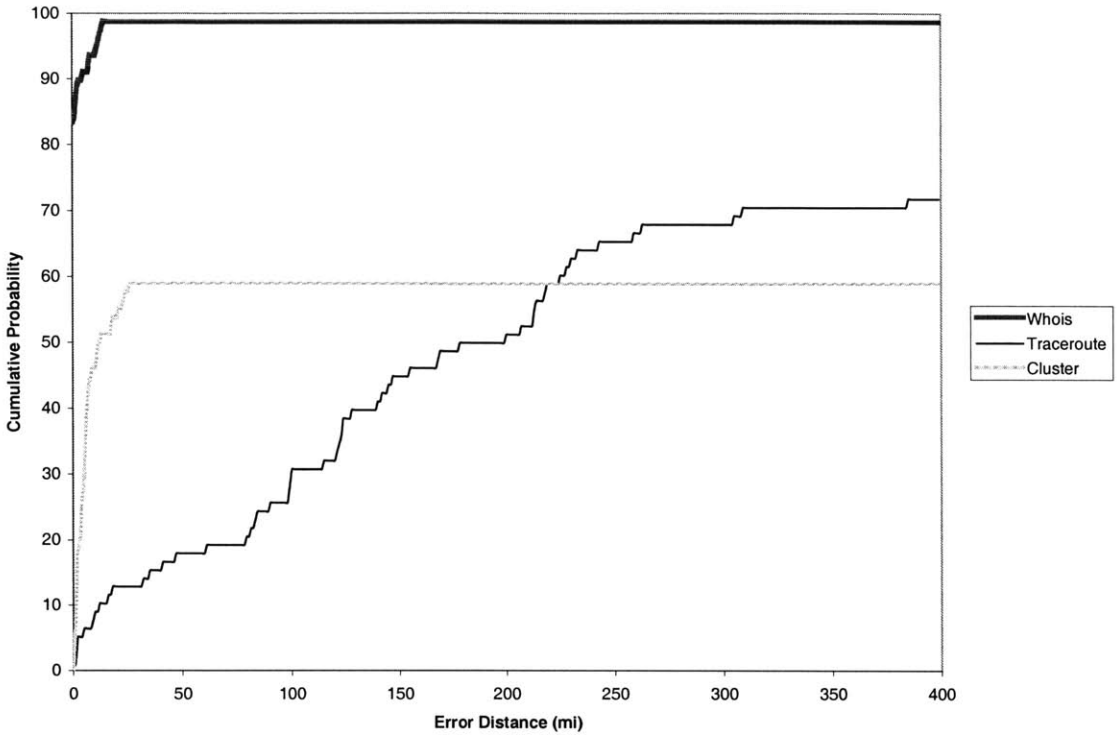

Figure 5-1: The CDF of Each Component for the Set of University Web Servers

demonstrates the advantages of this component in targeting non-disperse entities such as a university or small company.

Unfortunately, the whois component fails to achieve these incredible results for clients which are located in geographically disperse second-level domains. The analysis of the whois component on a larger, less-specific set of data led to the statistical results presented in Table 5.2. This data indicates that while the median error distance for this technique is still relatively low, it can also give extremely inaccurate predictions that are highly undesirable in certain applications. In addition, the system comparisons later in the chapter will show that while this low median error distance is impressive, it can also be a misleading indication of the accuracy of the whois component.

Table 5.2: The Statistical Analysis of Each Component for All Test Data

	Whois	Traceroute	Cluster
IP's Able to Locate:	456	434	357
IP's Unable to Locate:	27	49	126
Best Error Distance (mi):	0	0.19	0
Median Error Distance (mi):	31.19	121.37	4.03
Worst Error Distance (mi):	5004.39	594.77	253.25

5.3.2 Traceroute Parsing Analysis

For the next evaluation, the traceroute component was selected to be analyzed. As seen in Figure 5-1, parsing the results of a `traceroute` does not give a desirable success rate for clients in the educational domain. As previously discussed, the reason for this is that many, if not all, of these universities are connected to the Internet2 Abilene network. In this network, the longer-length Internet links often lead to nearby network routers that are several hundred miles away from the client.

Further statistical analysis revealed that while the median error distance of this technique is higher than that of the whois component, it does not give the terribly inaccurate results that are often possible with WHOIS parsing. Table 5.2 indicates that the traceroute component often does not attempt to locate a client in a situation that would give extremely inaccurate geo-targeting information. With a worst-case error distance of under six hundred miles, the traceroute component has obvious advantages over the whois component in certain applications.

5.3.3 Network Cluster Analysis

The last component analysis of the evaluation revealed the most accurate technique in the system. Unfortunately, while the network cluster component produces highly accurate results for both educational domains and geographically disperse corporations, it often cannot determine a location at all. As seem in Figure 5-1, the cluster component is only able to determine the location of nearly sixty percent of the web servers in the test data. The graph also indicates that this technique quickly levels off

and does not resolve any more hosts after a certain error distance. This indicates that while network clustering is an extremely accurate solution to the location mapping problem, it often is unsuccessful in geo-targeting a client.

The data in Table 5.2 confirms this hypothesis. The statistical analysis on the entire set of test data revealed that while the median error distance is less than five miles for the network cluster component, this technique is unable to give any indication of geographic location for over twenty-five percent of the clients being tested. Therefore, determining geographic clusters in network addresses may provide good location mapping if successful, but is not an overall solution for the location mapping problem.

5.4 Final System Performance

The goal of the final system design was to combine the strengths of each individual system component from Section 5.3. As seen in this section, each technique has the ability to contribute to the success of the final system implementation. While the whois component can be extremely inaccurate for large, disperse corporations, its accuracy is unsurpassed in identifying the location of clients in the educational domain. The analysis also revealed that while network clustering is very accurate for all other domains, it often returns no location at all. Fortunately, the traceroute component is the final piece of the puzzle that pulls the system together. If the client is realized to exist outside of an educational institution, and the network clustering component is unable to determine its location, the traceroute component provides a useful indication of the most likely location of the host.

The domain-based component selection of the final system design creates an application that has a lower median error distance and also resolves more hosts than each of the individual components. Table 5.3 presents the statistical analysis that was generated when using the final system on all test data that was collected. As seen in the table, half of the clients in the test data could be correctly geo-located within three miles of their actual location. The reliance on the whois component

Table 5.3: The Statistical Analysis of the Final System for All Test Data

IP's Able to Locate:	475
IP's Unable to Locate:	8
Best Error Distance (mi):	0
Median Error Distance (mi):	3.18
Worst Error Distance (mi):	594.77

for educational domains and the application of the network clustering component in other cases, leads to the much improved system performance. The worst-case error distance also indicates the possibility of relying on the traceroute component for a last-resort resolution.

In the next section, data is presented that demonstrates how this final system design is superior to each individual component in every way.

5.5 System Comparisons

While the statistical analysis presented in the previous sections gives a relatively good indication of the performance of each implementation in numbers, it is important to visualize these results. Figure 5-2 shows the graph of the cumulative distribution function for each component (and the final system) when run on the entire set of test data. The "Combo" technique represents the design of the final system which utilizes the strengths of each individual location mapping technique.

In the graph, the deception of the median error distance of the whois component is immediately visible. Even though half of the clients are accurately located within a short error distance, the graph quickly levels off after the fifty percent mark. This leveling is due to the inability for this component to correctly locate a large set of the hosts which are part of geographically disperse domains. Therefore, the visualization indicates that the data set most likely consists of a good mix of clients in both small, clustered domains and also those in large, disperse corporate entities.

The cumulative distribution function of the traceroute component indicates the

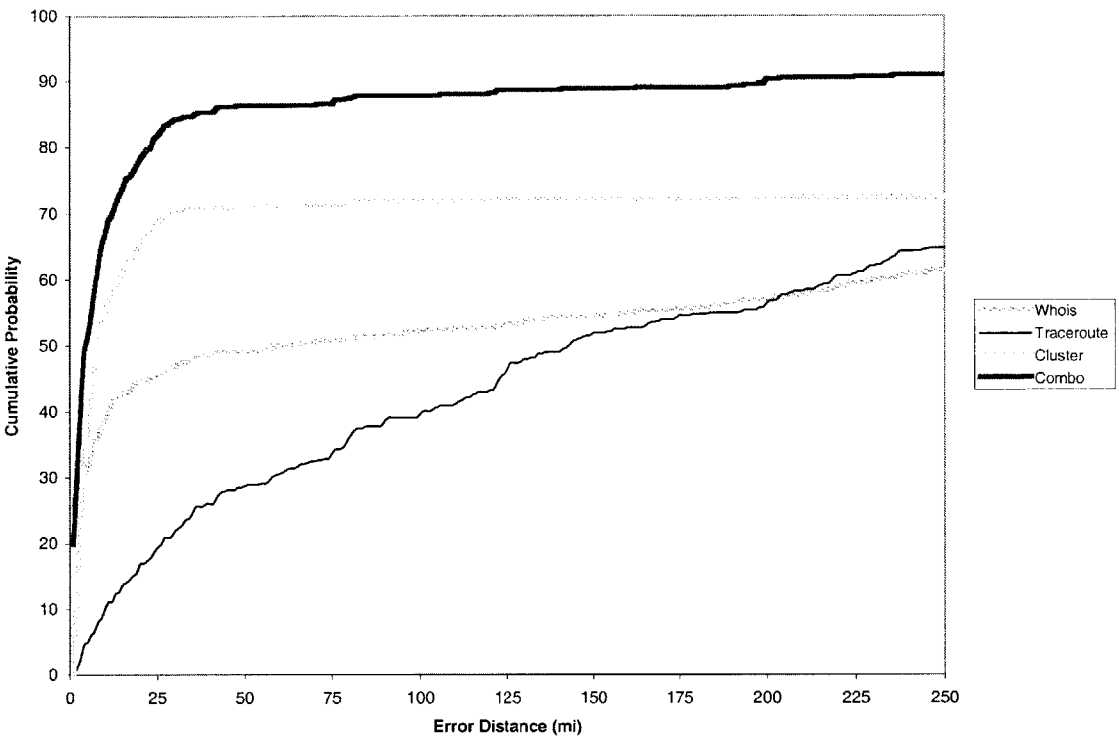

Figure 5-2: The CDF of Each Design for the Entire Set of Test Data

steady climb associated with parsing router names from `traceroute` results. While a large set of Internet hosts cannot be correctly located within a short error distance, the component demonstrates a steady rise as the distance increases. As the other strategies tend to level off at a certain error distance, the traceroute methodology is not flawed with this weakness.

As seen in the figure, the network clustering component can only identify about seventy percent of the hosts in the test data. The astonishing accuracy of this technique can be seen for these identifiable clients. Unfortunately, after these hosts have been identified, it is apparent that the cluster look up becomes unsuccessful. The quick plateau in the graph for this strategy indicates the strengths and weaknesses of this methodology. If the cluster component is able to determine a location for any IP address, it is very likely that the location is accurate. Unfortunately, as seen in the figure, not every host can be resolved with this technique.

To combine the strengths of each individual technique, the final system relies on domain-based component selection to improve its geo-location accuracy. While

the whois and network clustering graphs quickly flatten out, the final system design continues to climb. This gradual increase in host resolutions is due to the assistance added from the traceroute methodology. As seen in the figure, the final system is able to locate over eighty-five percent of the hosts in under a thirty-mile error radius.

While each component has an associated strength and weakness, the combination design of the final system is advantageous in many ways. As demonstrated, the domain-based component selection methodology produces highly accurate results for a large percentage of the hosts in the test data. Even after this large set of clients has been identified, the final system continues to geo-locate other machines with slightly less accurate results. By systematically choosing component implementations in a rule-based strategy, the final application generates geo-targeting results far superior to each individual design.

Chapter 6

Conclusion

This thesis discussed the possibility of determining the geographic location of Internet hosts solely from IP address information. The design, implementation, and evaluation of several unique components for geo-targeting led to an understanding of the strength and weakness of each strategy. Upon realization of the pros and cons of the location mapping designs, a new system was built that combines the best features of each technique. An overall evaluation of each component and the final system was then presented to demonstrate the accuracy gained with the use of a *Domain-Based Component Selection* methodology. As discussed in this paper, these findings demonstrate that location mapping can be used as a useful tool for many common Internet applications.

For the past several years, people have questioned the ability of location mapping technology. Whether the desires are to create a more personalized web experience, increase business intelligence, or to please foreign justice systems, the possibility of geo-targeting Internet clients has been under intense scrutiny. Small businesses, large online retailers, news organizations, and government agencies could all use the knowledge of the geographic location of web surfers to improve their success. As shown in Chapter 2, an accurate location mapping application could change the way business is done on the Internet.

Building on the research efforts presented in Chapter 3, the next chapter discusses the design and implementation of various components to achieve a mapping

from IP addresses to geographic location. In this study of domain registrant lookups, `traceroute` hostname analysis, and geographic network clustering, a realization of component success dependences was achieved. Upon initial evaluation of each location mapping technique, it became evident that the accuracy of their resolution varied according to the top-level domain of the client. Therefore, a new application was implemented to overcome the domain dependences present in the individual components. In this domain-based component selection application, the system dynamically chooses which strategy to use for geographic resolution based on the top-level domain of the client. To further improve the accuracy of the application, additional measures were also added to the system which help eliminate inaccurate predictions of client locations.

To provide a thorough understanding of the design choices made and system resolution accuracies, Chapter 5 evaluates the performance of each component implementation. The first analysis establishes the success of the whois component for clients in the educational top-level domain, and presents the ways in which the other components are inadequate in these resolutions. It was then shown that while the traceroute component is not desirable for fine-grained resolution applications, it does not provide the highly inaccurate results often present in using the whois component. Finally, the network clustering component was realized to be the most accurate strategy of the system, but unfortunately it lacks to ability to resolve a large set of the Internet's hosts.

These evaluations led to the design of a new application which utilizes a Domain-Based Component Selection methodology to provide more accurate system performance. In this strategy, clients in the educational domain are chosen to be resolved with the use of the whois component discussed in Chapter 4. For all hosts in other top-level domains, the system uses a predefined set of rules to correctly choose the strategy most likely to determine the correct location of the client machine. The analysis of this final system design helped visualize the importance of the domain-based component selection technique. While the other system components successfully solved the location mapping problem for a small subset of the Internet's hosts, the final system

provided much better accuracy for a larger number of the clients in the test data.

An evaluation of this system for nearly five hundred client IP addresses and their known geographic locations led to the graph presented in Figure 5-2. In this graph, it can be seen that the domain-based component selection application correctly geo-targets nearly ninety percent of the hosts in the test data within a twenty-five mile radius of their actual location. In addition, this multi-component system correctly determines many national proxies that can lead to highly inaccurate location resolutions. While each individual component provided useful solutions to the location mapping problem, the final system design combined the strength of each technique in order to obtain far superior results.

The location transparency of the Internet has been a significant factor in the world-wide expansion of its networks. Unfortunately, this transparency has left web developers and service providers with difficult challenges in the presentation and distribution of online information. This thesis demonstrated that while creating a perfect mapping between IP addresses and geographic location is not possible, systems can be developed to provide highly accurate results for a large set of the Internet's hosts. In addition, further expansion of the domain-based component selection methodology would allow for the creation of geographic resolution implementations that are satisfactory for many of the suggested Internet applications.

Appendix A

Figures

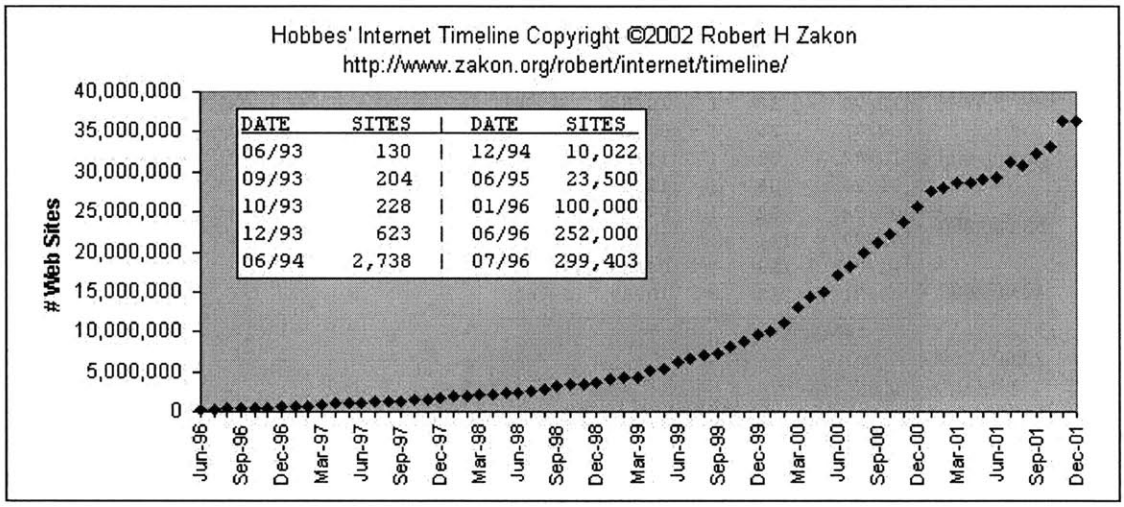

Figure A-1: The Growth of Websites on the Internet[23]

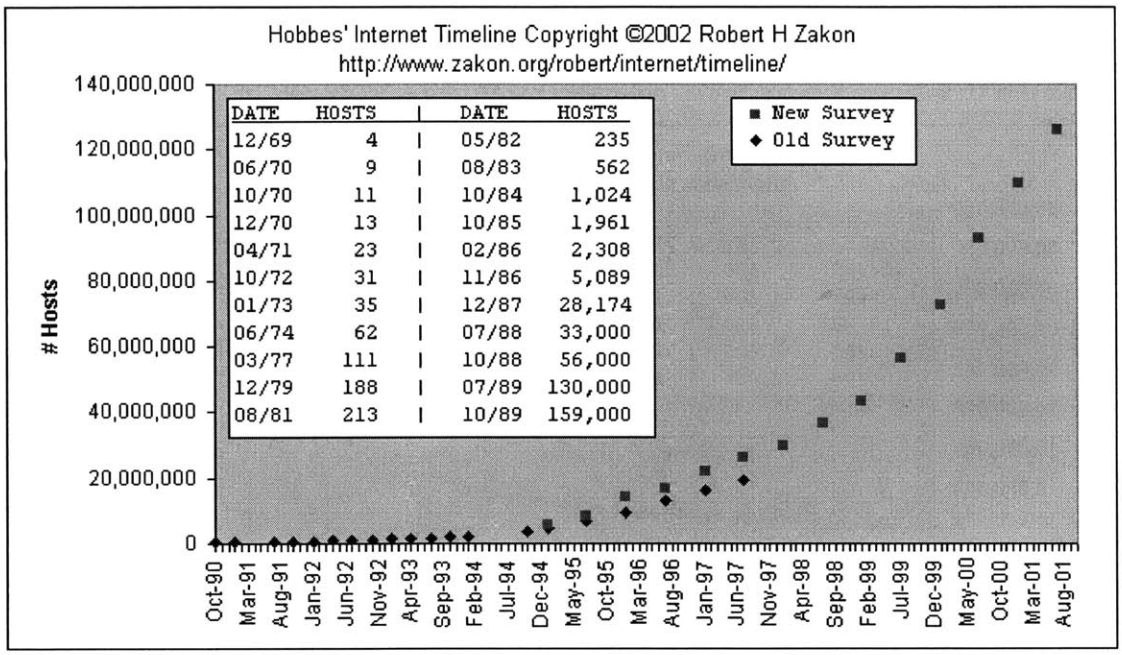

Figure A-2: The Rapid Expansion of the Online Community[23]

```
#Regular Expressions
s/^.*?\.(.*?)\d-.*?\.bbnplanet\.net/$1/this,cities.db
s/^.*?\.(.*?)-.*?\.bbnplanet\.net/$1/this,cities.db
s/^.*?\.(.*?)\.bbnplanet\.net/$1/cities.db
s/(.*?)\d-.*?\.bbnplanet\.net/$1/cities.db
s/^.*?\.(.*?)-.*?\.bbnplanet\.net/$1/cities.db

#Data --- Do Not Remove this line
nyc=new$york,ny,us
washdc=washington,dc,us
sajca=san$jose,ca,us
lsajca=san$jose,ca,us
philadelp=philadelphia,pa,us
bstnma=boston,ma,us
vsanfran=san$francisco,ca,us
sttlwa=seattle,wa,us
phlapa=philadelphia,pa,us
nycmny=new$york,ny,us
lsanca=los$angeles,ca,us
ontrca=ontario,ca,us
crtntx=carrollton,tx,us
cambridge=cambridge,ma,us
chcgil=chicago,il,us
paloalto=palo$alto,ca,us
dllstx=dallas,tx,us
atlnga=atlanta,ga,us
```

Figure A-3: A More Complex Parse File Used for Genuity Routers

Bibliography

[1] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile context-aware tour guide. In *ACM Wireless Networks*, number 3 in 1, pages 421–433, 1997.

[2] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM (2)*, pages 775–784, 2000.

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

[4] Orkut Buyukkokten, Junghoo Cho, Hector Garcia-Molina, Luis Gravano, and Narayanan Shivakumar. Exploiting Geographical Location Information of Web Pages. In *WebDB (Informal Proceedings)*, pages 91–96, 1999.

[5] C. Davis, P. Dixie, T. Goodwin, and I. Dickinson. A Means for Expressing Location Information in the Domain Name System, January 1996. RFC 1876.

[6] Christopher Davis. DNS LOC: Geo-enabling the Domain Name System. http://www.ckdhr.com/dns-loc/index.html, March 2001.

[7] Junyan Ding, Luis Gravano, and Narayanan Shivakumar. Computing Geographical Scopes of Web Resources. In *26th International Conference on Very Large Databases, VLDB 2000*, Cairo, Egypt, September 10–14 2000.

[8] K. Egevang and P. Francis. The IP Network Address Translator (NAT), May 1994. RFC 1631.

[9] Lori Enos. Yahoo! Ordered to Bar French from Nazi Auctions. *E-Commerce Times*, November 20 2000.

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

[11] Sarang Gupta. Sarangworld Traceroute Project. http://www.sarangworld.com/TRACEROUTE/, June 2001.

[12] K. Harrenstien, M. Stahl, and E. Feinler. NICNAME/WHOIS, October 1985. RFC 954.

[13] Van Jacobson. The traceroute utility, June 1999. Available as ftp://ftp.ee.lbl.gov/traceroute.tar.Z.

[14] Balachander Krishnamurthy and Jia Wang. On Network-Aware Clustering of Web Clients. In *SIGCOMM*, pages 97–110, 2000.

[15] Stephen E. Lamm, Daniel A. Reed, and Will H. Scullin. Real-time Geographic Visualization of World Wide Web Traffic. *Computer Networks and ISDN Systems*, 28(7–11):1457–1468, 1996.

[16] Kevin S. McCurley. Geospatial Mapping and Navigation of the Web. In *World Wide Web*, pages 221–229, 2001.

[17] Jan Moller. Online gambling deal tempts city. *Las Vegas Review Journal*, December 12 2001.

[18] Mike Muuss. The ping utility, December 1983. http://www.ping127001.com/pingpage.htm.

[19] K. Obraczka and F. Silva. Looking at network latency for server proximity. Technical Report 99-714, USC/Information Science Institute, 1999.

[20] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proc. of ACM SIGCOMM 2001*, San Diego, CA, USA, August 2001.

[21] Ram Periakaruppan and Evi Nemeth. GTrace: A Graphical Traceroute Tool. In *13th Systems Administration Conference - LISA '99*, pages 69–78, Seattle, WA, USA, November 1999.

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

[23] Robert H'obbes' Zakon. Hobbes' Internet Timeline v5.5. http://www.zakon.org/robert/internet/timeline/, 2002.

[24] Akamai Technologies, Inc. http://www.akamai.com.

[25] America Online, Inc. (AOL). http://www.aol.com.

[26] Digital Envoy. http://www.digitalenvoy.net.

[27] GeoBytes, Inc. http://www.geobytes.com.

[28] Infosplit. http://www.infosplit.com.

[29] Internet2. http://www.internet2.org.

[30] Ixia Corporation. http://www.ixiacom.com.

[31] NetGeo, Inc. http://www.netgeo.com.

[32] Quova, Inc. http://www.quova.com.

[33] The National Fraud Information Center. http://www.fraud.org.

[34] The New York Times on the Web. http://www.nytimes.com.

[35] The Weather Channel. http://www.weather.com.

[36] U.S.News & World Report Online. http://www.usnews.com.

[37] VisualRoute - Visual Traceroute Utility. http://www.visualware.com/visualroute/index.html.