GEOGRAPHIC LOCATION OF A COMPUTER NODE EXAMINING A TIME-TO-LOCATION ALGORITHM AND MULTIPLE AUTONOMOUS SYSTEM NETWORKS

THESIS

Duane C. Sorgaard, Master Sergeant, USAF

AFIT/GCS/ENG/04-17

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCS/ENG/04-17

GEOGRAPHIC LOCATION OF A COMPUTER NODE EXAMINING A TIME-TO-LOCATION ALGORITHM AND MULTIPLE AUTONOMOUS SYSTEM NETWORKS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Duane C. Sorgaard, BS

Master Sergeant, USAF

March 2004

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GCS/ENG/04-17

GEOGRAPHIC LOCATION OF A COMPUTER NODE EXAMINING A TIME-TO-LOCATION ALGORITHM AND MULTIPLE AUTONOMOUS SYSTEM NETWORKS

Duane C. Sorgaard, BS
Master Sergeant, USAF

Approved:

_________//signed//_____________________	4 March 2004
Major Rusty O. Baldwin (Chairman)	date
_________//signed//_____________________	4 March 2004
Dr. Richard A. Raines (Member)	date
_________//signed//_____________________	5 March 2004
Dr. John F. Raquet (Member)	date

To my patient and loving wife.

Acknowledgments

I would like to thank my thesis advisor Major Baldwin, for the time, guidance, support, and direction he provided through the proof of one theory to the adaptation and discovery of another idea into a complete thesis. I would also like to thank my thesis committee members, Dr. Raines and Dr. Raquet, for their guidance and support throughout the thesis process. Finally, I would like to thank my wife, for her undying support and understanding through the entire AFIT experience.

Duane C. Sorgaard

Table of Contents

List of Figures

Figure

Page

List of Tables

Table

Page

AFIT/GCS/ENG/04-17

Abstract

To determine the location of a computer on the Internet without resorting to outside information or databases would greatly increase the security abilities of the US Air Force and the Department of Defense. The geographic location of a computer node has been demonstrated on an autonomous system (AS) network, or a network with one system administration focal point. The work shows that a similar technique will work on networks comprised of a multiple AS network. A time-to-location algorithm can successfully resolve a geographic location of a computer node using only latency information from known sites and mathematically calculating the Euclidean distance to those sites from an unknown location on a single AS network.

The time-to-location algorithm on a multiple AS network successfully resolves a geographic location 71.4% of the time. Packets are subject to arbitrary delays in the network; and inconsistencies in latency measurements are discovered when attempting to use a time-to location algorithm on a multiple AS network. To improve accuracy in a multiple AS network, a time-to-location algorithm needs to calculate the link bandwidth when attempting to geographically locate a computer node on a multiple AS network.

GEOGRAPHIC LOCATION OF A COMPUTER NODE EXAMINING A TIME-TO-LOCATION ALGORITHM AND MULTIPLE AUTONOMOUS SYSTEM NETWORKS

I. Introduction

"The Air Force believes that dominating the information spectrum is as critical to conflict now as controlling air and space or occupying land was in the past and is seen as an indispensable and synergistic component of aerospace power" [AFD98]. These systems therefore, must be protected to the level required of any weapons asset. An enemy has the capability to exploit these assets either by hacking into them and gaining access to DOD information or disrupting access by authorized users to this information through denial of service attacks.

To establish the location of an enemy attacking an information system, the Air Force needs the capability to geographically locate a node on the Internet via its logical address consistently and reliably. The problems associated with this requirement include interference from background network traffic, packet routing, and packet time of flight. These interference sources introduce unpredictable latencies, which make it impossible to establish a relationship between packet round trip time and distance to the location.

1.1 Background

The NSA developed a time-to-location algorithm which uses mathematical calculations to eliminate the effects of line speed, queue size, switch speed and geographical physical separation of computer nodes in latency measurements [NSA02]. This method appears to be quite reliable within a single autonomous system (AS). An autonomous network is a network that is owned and operated by one vendor in contrast to the multiple AS network consisting of multiple autonomous networks that make up the Internet. Establishing a relationship between round trip time (RTT) and location on a single AS produces fairly consistent results; however, moving to an environment that has a multiple AS network introduces unpredictable delays more difficult to eliminate.

1.2 Problem Definition

The goal of this research is to determine the geographic location of a node using only packet latency measurements to establish time-to-location "markers". One might assume this is a trivial task limited to finding the RTT of a packet from one computer node to the distant computer node, then designating the RTT as a finite measurement to be divided by the line speed over the given medium. This approach however, does not take into account various latencies introduced due to the particular route the packet travels, queuing delays, switch speeds, and physical distances. In fact these latencies make establishing a time-to-distance relationship impossible.

Baseline physical distances between destination city centers are used to establish a reference minimum time from city to city. This minimum time, or t(min), is the shortest time a packet takes to travel from city to city and establishes a parameter that remains a constant. Solving a linear slope formula for $y = mx + b$, where m is the slope of the line, x is the size of the packet transmitted and b is the y intercept. Using this formula the y intercept or the round trip time (RTT) for a theoretical "zero byte" packet can be determined. It is expected that line speed will converge to a linear slope and provide the time to transmit a theoretic "zero byte" packet (which is independent of line speed) leaving only packet size as a factor. A hypothesis is made that latencies in the network will affect the slope of the line that produces a theoretical "zero byte" packet, but that the packet sizes themselves will not affect the slope. Thus, the slope of the line that produces a theoretical "zero byte" packet will only rotate around b, the point of intercept on the y axis and will provide a consistent and reliable time-to-location algorithm output.

1.3 Summary of Current Knowledge

Latency measurement tools that are currently in use include Ping, Whois, Traceroute, GTrace, Pathping, and Skitter [HSF85, SUN99, PeN99, Mic03, HPMC02]. These tools all use a latency measurement and some even attempt to establish the path the packet takes on its round trip journey between source and destination. None can be used to reliably achieve a geographic time-to-location value to a computer node within a single or multiple AS network. This research effort begins by validating the NSA time-to-location algorithm in a controlled laboratory environment. After this time-to-location algorithm has been validated, a baseline of latency calculations are available to assist in identifying latency introduced when moving to a multiple AS network environment.

1.4 Assumptions

Several assumptions are made to meet the goals of this research. A simulation model using OPNET version 10.0 modeling and simulation software is developed using AS network information that can be obtained from the Internet [OPNET03]. An OPNET network model testing environment is used so interference can be controlled, and to demonstrate that the original NSA time-to-location data results are repeatable in a laboratory environment [NSA02]. The time-to-location algorithm uses the round trip time from a polling network node to multiple distant nodes on the network. The Euclidean distance is then determined from the unknown polling location to all known distant nodes. Using this data, a reliable time-to-location correlation is established for an AS network using latency measurements. The calculated linear slope is assumed to be identical for the single AS network as it is for the multiple AS network.

It is assumed that all network traffic is carried over fiber optic cables that travel along the main Highways traversing the United States. This baseline physical distance between cities on the network is established using city center to city center driving distances between the poling node and destination cities obtained from the Mapquest website [Map03]. An analytic model using Euclidean distance measurements is established based on physical distances establishing a minimum time or t(min) baseline between cities using the city to city driving distances.

The first AS network simulation uses the AT&T IP network model latency measurements as calculated from the baseline driving distances taken from the calculation of mileage divided by the speed of light in glass to account for the fiber optic cable latency. Based on this information AS network simulations collect latency measurements used to develop a geographic location baseline from city to city and identify latency "invariants" within the network simulation for the multiple AS network simulation testing results. The AT&T simulation model uses Asynchronous Transfer Mode (ATM) switches as a baseline for long haul communications. The ATM network will provide a bandwidth constant for research simulations. The MCI simulation model uses Fast Ethernet as a baseline for the network links which is used to provide a contrast in topologies for the simulations.

The AT&T and MCI network node locations are obtained from the public web-sites of the companies, although most interconnections between the cities are assumed to exist for purposes of this research [ATT03, MCI03]. They merely establish a baseline to demonstrate the time-to-location algorithm and Highlight differences in routing paths of the two networks within the continental United States boundaries.

1.5 Scope

The scope of this research is limited to two distinct problems, the first to demonstrate that the previous NSA research can be duplicated in a controlled laboratory environment. The second, to demonstrate and identify issues with packets crossing over from one commercial vendor network to another vendor network. The multiple AS network is the latest research area in the time-to-location algorithm and sources of latency inconsistency need to be demonstrated and identified. The network under study is limited to the continental bounds of the Lower 48 states of the United States.

1.6 Document Overview

This chapter provides an overview of various aspects of the Internet, such as IP addressing, and the way information is transferred throughout the Internet. Additionally, this chapter introduces the hypothesis, summary of some current location methods and the scope of the research. Chapter II is the literature review providing background information on the time-to-location algorithm and network models that serve as a foundation for the research. Chapter III introduces the methodology used to attain the goal of the research. Chapter IV provides the implementation of the methodology and the analysis of the results. Chapter V contains the conclusions of this research and discusses future work related to the research.

II. Literature Review

2.1 Introduction

This chapter discusses Internet Protocol (IP) packet and traffic characteristics in a number of situations and topologies, such as ATM, Multicast traffic, and fragmentation of IP packets traveling across networks with different Maximum Transfer Unit (MTU) sizes. A number of tools or methods available for use to locate a computer node address on the Internet are also discussed. These methods may use an Internet database, such as Domain Name Services (DNS), which may or may not be up to date. Other methods may physically trace the route from source to destination to find an Internet Protocol (IP) address logically across the Internet.

Cyberspace geography provides background information on how we as humans relate the physical three dimensions of our world in a logical two dimensional interpretation of cyberspace. Points of Presence (POP) or the physical location of a commercial vendor access into the Internet backbone can also cause concerns for geographically locating a node across a network. A POP provides a central access point for multiple sub-network connections into the backbone, creating an invariant that may be hundreds of miles from the geographical location of the computer node. NSA research demonstrates the concept of a time-to-location algorithm which works to geographically locate a node on an AS network [NSA02].

2.2 Internet Protocol Characteristics

IPv4 addresses are 32 bit numbers, which consist of 4 octets that range from 0 to 255, separated by a period. Each IP address identifies an addressable node on the Internet or a subnet. An example is 24.209.66.18. Collecting Internet traffic characteristics has become more difficult due to the segmentation of the Internet into multiple commercial vendors, each competing for the others economic e-business.

2.2.1 IP Traffic Patterns Latency can be increased with a large traffic background load on a network, so traffic patterns must be analyzed to allow interpretation of round trip time results. Traces collected from one of four OC-3 ATM links at the NASA Ames Internet exchange (AIX) determined that scheduling was accomplished at the packet level within the queues [McC00]. The distribution of the packets was not completely uniform in the four OC-3 links; two of the links carried double the traffic of the other two links.

The distributions measured were from packet sizes less than 1600 bytes and were built from approximately two, one week periods in the study. One collection time was towards the beginning of the study period and one towards the end for the period of May 1999 through March 2000. The collections contain traffic from different times of day and different workloads of the network, so it is believed that an "average" picture of the packet size distribution is obtained [McC00].

Approximately 85% of the traffic is TCP, with a large proportion of that traffic being HTTP and FTP bulk transfers. The majority of the packets were of four sizes: the TCP minimum size of 40 bytes (TCP acknowledgements without a payload); Ethernet maximum payload size of 1500 bytes using TCP and Maximum Transfer Unit (MTU) path discovery; lastly 556 and 576 bytes packets from TCP implementations that don't use MTU path discovery. The two trace distribution results were similar despite the nine month separation period between collections [McC00]. No significant long term trends were found in overall packet size distributions, but some short term trends were identified including an increase in the volume of e-mail traffic during holiday shopping season and a difference in online gaming volume on weekday versus weekend [McC00].

2.2.2 IP Multicast Traffic Multicast traffic applications typically consist of satellite broadcast replacement, audio and video distribution, multimedia conferencing and other distributed simulations [BeC02]. This background traffic will affect time-to-location latencies because of the time it takes to process at individual routers. All multicast traffic is monitored not just explicit sessions, IP traffic flow patterns and characteristics which include packet distributions, duplication and fragmentation. This assists in helping to gain an understanding of true traffic patterns on the Internet.

In a recent study, Point of Presence OC-12c Asynchronous Transfer Mode (ATM) links were monitored at four different sites; Chicago, Houston, Washington and New York City [BeC02]. Router performance is typically bounded by packet rates and not by bit rates because multicast traffic puts the burden of packet replication onto the router. The mean packet size in bits at all sites combined was 897 bits with a standard deviation of 567 bits [BeC02]. At each site the traffic patterns varied widely which reflects on the variety of customers and applications utilizing each individual site [BeC02].

Duplication occurs when ATM is used at the data link layer because ATM decouples itself from the IP layer. ATM uses a logical interface creating a Permanent Virtual Circuit (PVC); although the IP traffic may arrive on the same physical interface, the logical interface is treated as a separate interface and the IP traffic is duplicated creating a multicast flow in both directions on the same physical interface [BeC02]. Multicast traffic is time of day and week dependent, but exhibits a constant baseline rate. Only 0.5% of the multicast traffic is fragmented while 3.2% of the traffic is marked ‘don’t fragment’ [BeC02]. 76% of the traffic is found to be short lived and does not contribute to multicast packet volumes. This will help establish a simulation model for IP traffic patterns and help identify latencies introduced by traffic duplication and multicast traffic when packets travel amongst various commercial backbones. This may help establish some type of predication capability to a time-to-location algorithm.

2.2.3 Fragmented IP Traffic Characteristics The Internet Protocol (IP) provides a Lowest common denominator protocol which facilitates communication between multiple AS networks [SMC01]. IP fragments packets when transferring them from a large MTU network to a smaller MTU network, this can add an additional latency measurement that must be considered when calculating a time-to-location algorithm on a multiple AS network. Each fragment duplicates the original packet header for correct identification of the destination. The last fragment does not have the ‘more fragments’ bit set. This ensures the receiving host knows there are no more fragments and assists in reassembling the original packet. The size of the IP fragment is the size of the smallest MTU minus the size of the header added to each fragment.

Hosts using IP can communicate without specifying a route due to IP routing, even if the current route of the packet is different than the previous route. Common assumptions made about fragmented traffic include: (1) it is no longer prevalent; (2) it is only present in LANs; (3) it is not present on backbone links; or (4) only misconfiguration causes fragmentation. In fact, the majority of the fragmented traffic is UDP (68% by packet), but also includes TCP, IPSEC, ICMP and tunneled traffic [SMC01]. Tunneled traffic turned out to be the single largest cause of fragmentation and accounts for 16% of the packet fragmentation [SMC01]. Furthermore, fragmented traffic increases the workload on routers involved and was detrimental to wide area network performance and increases the latency measurement of the packets.

2.3. Tools to Measure Latency

A number of tools exist to discover a computer node on the Internet, although none reliably geolocate a node in its physical location or identify any latency issues. Some tools use online databases whose reliability is undetermined; while others attempt to trace each step the network packet takes along the route to its destination to logically locate the computer node. The Bellman-Ford algorithm is demonstrated as one method Internet routers use to create routing tables [CLR01].

2.3.1 Ping The ping utility sends a packet to a designated host and waits for a reply [Ker02]. The destination hosts address and round trip time is returned for each pair of packets. The total number of packets sent, received, percent of packet loss, minimum, average and maximum round trip times (RTT) are also calculated. This utility can be used to provide an initial RTT to a designated host. This utility does not count duplicate packets in the packet loss calculation, but it does use the duplicates in calculating minimum, average and maximum RTT. The minimum RTT is used as a first step in calculating the theoretical zero byte packets RTT.

2.3.2 Whois The whois database is a utility that contains administrative contact information for all domains, filled in at the time of registration [HSF85]. The whois database's reliability is largely dependent on the entity registering the domain providing reliable information and updates. Since entities are not required to provide updates, this database is often incorrect. This tool cannot provide a geographic location or reliable input to a time-to-location algorithm because of the inaccuracies of the data.

2.3.3 Traceroute One way to discover a route to a network is to use the traceroute utility. Traceroute tracks the route packets take to a network host from the requesting host using the time to live (TTL) within the IP header [Sun99]. TTL is used by each router decrementing the TTL field of an IP datagram, when TTL reaches zero, a router discards the packet and sends an error message to the originator. However, if the route includes an unreachable node, the utility exits. If the route the node uses can be determined, the route the packets have traveled may provide an indication of the geographic location of the distant end node.

2.3.4 GTrace GTrace is a graphical front end to traceroute [PeN99]. Often the name of a node in the path contains geographical information such as a city name/abbreviation or airport code, this information is entered in the DNS database. GTrace uses geographic information returned within traceroute to represent the information on a world map and provides a notional geographical path a packet took to reach the destination node. GTrace was developed at the University of Colorado at Boulder and is available for download at www.caida.org/tools/visualization/gtrace. GTrace output is notional since it uses DNS location records to obtain location information and cannot be used in reliable geographic location.

2.3.5 Third Party Addresses in Traceroute Traceroute reports the IP addresses of the routers used to a destination node [HyBC03]. Traceroute can be a very useful tool in developing source data in the study of Internet topology, performance and routing. Autonomous systems (AS) on the Internet are usually studied versus individual IP addresses. AS level analysis helps to determine the overall performance of the Internet, but is not very useful in any type of location finding, either geographically or logically.

One factor of the AS study that may be useful in geographic location evaluation is the ability to avoid errors due to midpath routing. That is the path segment variations can be seen in the unique segments during individual hops. A stable route needs to be established before any kind of geographic location attempt can be made. This stability can help in eliminating ambiguous routes to a distant end location.

2.3.6 Pathping Pathping is a route tracing tool that has features of both ping and traceroute. In addition to the information provided by traceroute and ping, Pathping reports the packet loss at routes along the way. This is intended to identify routers which may be causing network problems. A single latency is identified for packets traveling among commercial backbones, if the router that is causing network problems is identified.

2.3.7 Skitter A CAIDA topology probing tool is similar to traceroute and ping, except it has increased timestamp accuracy [Cla00]. A 52 bytes ICMP echo request packet is used, incrementally increasing time to live values until the target host is reached. Increased timestamp accuracy is helpful in producing more accurate time-to-location measurements. Each trace produces a record of IP addresses of responding intermediate routers on the forward path from source to destination, as well as producing the RTT.

2.3.8 Distance Metrics in the Internet Propagation time of a packet between two nodes on the Internet is a simple metric that reflects the performance as perceived by a user [Cla00]. Traversing from source to destination packets cross many links each having independent and unpredictable delays that include queuing delay, Low bandwidth, propagation latencies, and packet loss. Each of these latencies makes a contribution towards the overall end to end delay. IP path length, autonomous system (AS) path length, geographical distance, and round trip time (RTT) all have some correlation to the latency described below [Cla00].

IP path length is the number of hops traversed by a packet from a source to a destination. An Autonomous System (AS) is a network or networks under a single administrative domain. The AS is the domain that determines the reachability of the IP address; it is the home of the assigned IP address. Since the AS is the home network for an IP address, if you have found the originating AS, you have narrowed the search for the latency measurement of the destination IP. The numbers of AS transitions are counted from the source to destination path and the total number of autonomous systems traversed is tracked.

Geographic distance is defined as the distance between two hosts using the length of the earth's surface between the hosts. Geographic distance is a significant factor in the measuring RTT. RTT is the time a packet takes to traverse the network from source to destination and back to the originating host. RTT provides a correlation of the distance between the two hosts and will produce better results for a time-to-location algorithm to determine the Euclidean distance.

2.3.9 Bellman-Ford Algorithm OPNET's Routing Information Protocol (RIP) uses the Bellman-Ford algorithm to create routing tables in network simulations [OPNET03]. This algorithm is the original single-source shortest-path problem. Given a weighted, directed graph $G = (V, E)$ with source s and weight function $w : E \rightarrow R$, the Bellman-Ford algorithm returns a Boolean value indicating whether or not there is a negative-weight cycle that is reachable from the source [CLR01]. If a cycle with a negative value exists, then no solution exists and the algorithm returns that result. If a non negative cycle exists, the shortest path and weight is returned. The Bellman-Ford algorithm returns TRUE if and only if the graph does not contain a negative-weight cycle. The pseudo-code [CLR01] is as follows:

```
BELLMAN-FORD(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
for i ← 1 to |V[G]| - 1
    do for each edge (u, v) ∈ E[G]
        do RELAX(u, v, w)
for each edge (u, v) ∈ E[G]
    do if d[v] > d[u] + w(u, v)
        then return FALSE
return TRUE
```

Figure 2.1 shows how execution of the Bellman-Ford algorithm on a graph with 5 vertices. The source of the search is z, the weights of the vertices are shown and in this particular example, each pass relaxes the edges in lexicographic order: (u, v),(u, x),(u, y),(v, u),(x, v),(x, y),(y, v),(y, z),(z, u),(z, x) [CLR01]. The algorithm returns a TRUE. The algorithm computes shortest-path for all vertices reachable from the source [CLR01].

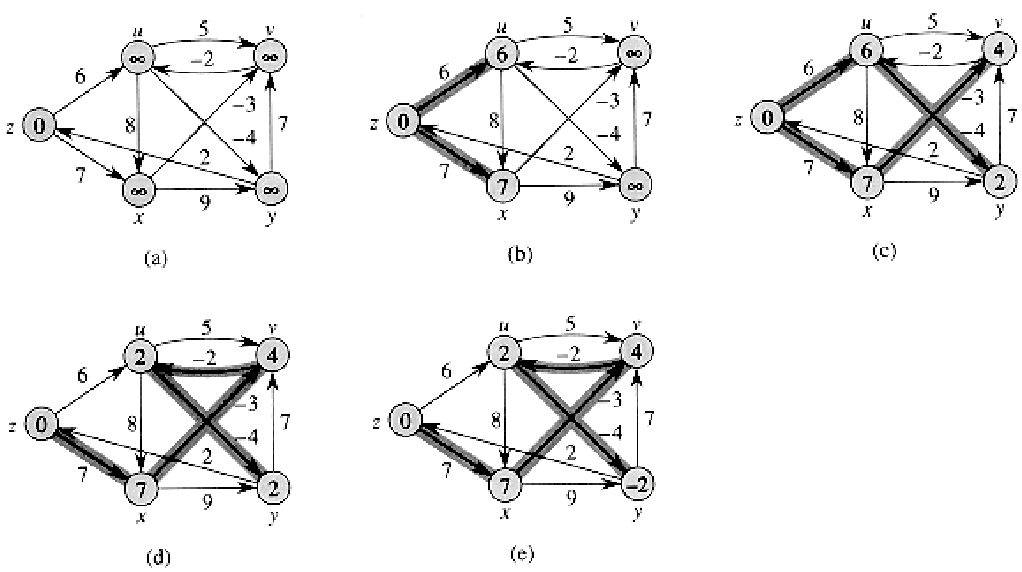

Figure 2.1 The Bellman-Ford algorithm

2.4 Cyberspace Geography

2.4.1 Naive Geography Naive Geography is the body of knowledge that people have about the surrounding geographic world [EgM95]. Geography is a scientific study of relationships, patterns, and processes of our world. Geographic Information Systems (GIS) are based in the definition and design of the underlying Naive Geography. Naive Geography is distinct from related topic areas such as spatial information theory, geographic information science, and Naive Physics. Central to the theory of Naive Geography is temporal and spatial reasoning [EgM95]. It employs qualitative reasoning methods characterized by variables that can only take on small predictive values. It uses qualitative spatial reasoning to be able to separate out numerical analysis from the magnitude of events, which deal with possibly undetermined values. The values are within a range, one of which is the correct result.

Qualitative information and reasoning is a complementary method, not a substitution to quantitative approaches. Qualitative approaches allow a user to combine a wide range of details and correlate a solution based on established landmarks. Naive geographic reasoning may actually contain “errors” and will occasionally be inconsistent [EgM95]. These theories do not hold to the belief that information systems should have only one solution. To develop a geographic location system, we need to relate Naive geography to geographic reasoning and how people think of the geographic world around them, whether it be in the Cartesian coordinate frame, cognitive mapping or even topological in nature. It is a need to develop a relationship that is intuitive, so no explanation is required for it to make sense to people observing the system.

Naive Geography is being developed as a two dimensional geographic space. It eliminates the horizontal and vertical coupling of dimensions in geographic space and is interpreted as a two-dimensional space with a third dimension becoming an attribute of position rather than an equal dimension in space. It does however couple time and space tightly to each other, which links geographic time to geographic space, such as how far an army can walk in a day. The mental map a person creates for themselves is generally biased to North-South, East-West configurations. This is an over simplification that can create problems for interpretation of geographic reality. Naive Geography links the way people think about geography and the models that can incorporate the thinking into information systems and geolocation.

2.4.2 Geographically Speaking All cyberspace geography needs to be addressed as both background information and how people interpret results; it provides background information for identifying latency based issues. If the packet travels from one country to another, does it pass through one centralized location prior to reaching its destination? Traces are analyzed centering on the following questions [CCAS01]: 1) Geographically what countries/states/cities are the biggest sources and destinations of IP traffic? 2) Where is the traffic from/to a particular geographic source/destination flowing? 3) How far does the IP traffic travel in relation to the actual distances between source and destination? Over a one hour time period traffic was collected from NASA AMES Internet exchange (AIX) containing 3.6 million IP flow traces [CCAS01].

The U.S. accounts for 92% of all the source bytes traffic. In the remaining 8%; Japan accounts for 2%, Canada, China, Korea and the Philippines accounted for the remaining 6%. The U.S. accounts for 69% of the destination bytes traffic showing that more requests are being made to hosts within U.S. borders, than are being made to the rest of the world [CCAS01]. Japan came in second again with 7% of the destination traffic, the rest of the destinations were scattered throughout the world. Breaking the traffic down to state and city levels; California, Washington, and Colorado lead the top 20 states listed accounting for ˜ 61.1% of the source traffic [CCAS01]. From NON-US destinations, Virginia, and California lead the top 20 destinations and account for ˜ 65.4% of the destination traffic [CCAS01]. Santa Clara, NON-US sources, Redmond, Louisville, and Seattle led the top 20 source cities listed accounting for ˜ 39.4% of the source traffic [CCAS01]. NON-US destinations, Fairfax, and San Jose led off the top 20 destination cities and accounted for ˜ 57.2% of the destination traffic [CCAS01].

AT&T's 80/20 research claims that 80% of the traffic originated from an AS stays within that boundary and does not cross over the AS boundary [CCAS01]. This tends to reinforce the demonstrated consistent results of the NSA time-to-location algorithm for geographically locating a computer node on an AS. Two points need to be made about the AT&T study, like the NSA study; (1) AT&T's tests were conducted on a single network to analyze how their network AS behaved and (2) this study looked at the geographic source and destination of IP traffic, not the IP source and destination of traffic. Unlike the NSA study, DNS registries were used to determine the geographical location of the traffic.

2.4.3 Geolocation Technologies Geolocation technologies for wireless applications can be divided into four categories; Mobile Station (MS) Based, Network Based, Network/MS Based, and Hybrid Type solutions [DjR01]. Of interest to this research is a MS Based geolocation called Assisted-Global Positioning System (A-GPS), as well as some network based geolocation technologies. Network Based geolocation technologies include Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), and Timing Advance (TA) [DjR01]. A-GPS consists of using a partial GPS receiver in a mobile station and predicted information obtained from the base network station [DjR01]. The base network station uses GPS predicted coordinate data based on the 10 – 15 square kilometer cell that the mobile station is located in [DjR01]. This is important to account for wireless connections into the Internet and the ability to geolocate wireless devices.

TOA is produced by three known base stations receiving a signal from the mobile station. The independently received arrival times are computed at a separate location, which produces a mobile station location. TDOA is produced by three known base stations receiving a signal from the mobile station and the difference of the received arrival times are computed, which provides the mobile location. AOA requires a special set of antennas to determine the angle of arrival by the location receivers. Base stations compute the intersection of arrival directions, thus providing a mobile location. Timing Advance uses frame/slot times at link establishment with the base station to determine the distance to the base station. Network hand-offs enforce the need for three known base stations, which are used to triangulate the mobile location. A mobile station moving in a predominately straight line makes this method unreliable.

2.4.4 Location-based Authentication Computer and network security can be improved through authentication based on geodetic location [DeM96]. Location based authentication techniques are used to secure networks accessed by remote users. The effect of the location based authentication is to physically locate cyberspace in the physical world. A users location can be used to validate a user helping to prevent unauthorized personnel from accessing the network from an unauthorized location. While this helps defend a network from attack, it does not geolocate an attacker. This is one of the reasons that reliable network geolocation abilities need to be developed.

2.5 Points of Presence

One of the places that latency is introduced in multiple AS networks Internet environment is when traffic moves from one vendor network to another. All commercial vendors researched use fiber optic cable networks within the continental United Sates, which is the basis for the assumption of all long haul communications being carried over fiber optic cables in the simulations created for this research.

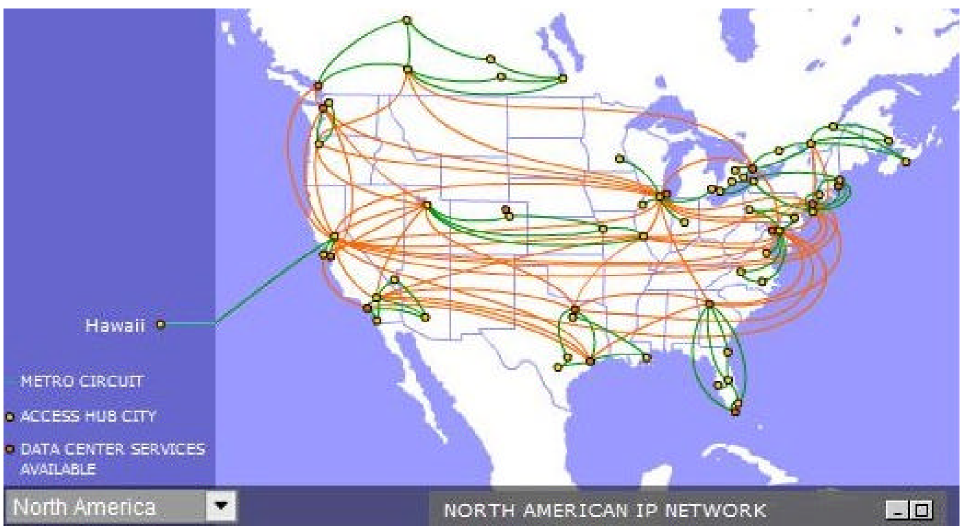

Figure 2.2 MCI North America Intra-Continental Presence

2.5.1 MCI MCI is one commercial vendor providing Points of Presence (POP) throughout the world [MCI03]. MCI network facilities of interest to this research are located in North America, since the geographical limits of this research is the continental United Sates. MCI maintains a very large fiber optic network, which validates the assumption made in this research that long haul network traffic is carried over fiber optic cable. The destination cities for the MCI simulation network are derived form the information gathered at MCI's Internet site. An example of the North American hub network for MCI is displayed in Figure 2.2.

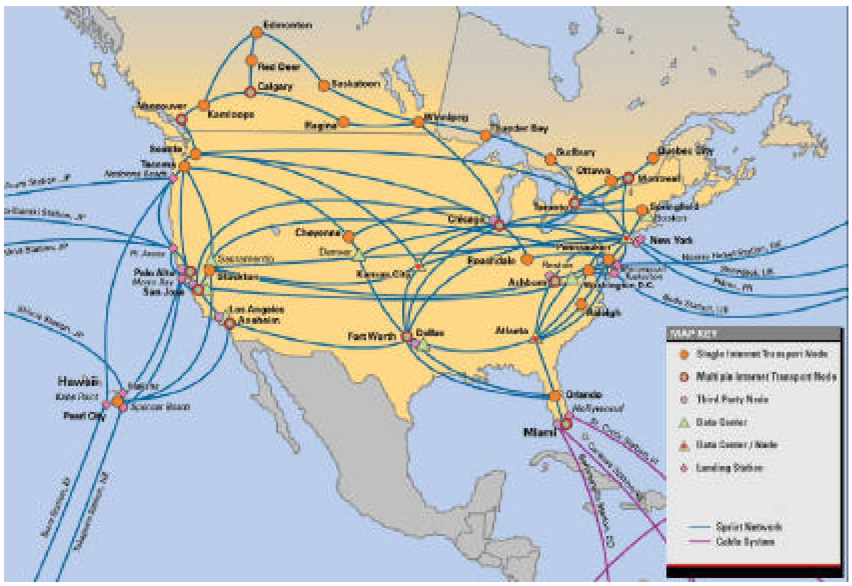

Figure 2.3 Sprint North American IP Network

2.5.2 Sprint Sprint is another POP provider with network facilities that maintain a very large digital network, validating the assumption made in this research that long haul network traffic is carried over fiber optic cable. An example of the North American Sprint network is shown in Figure 2.3.

2.5.3 AT&T AT&T is a provider of IP services, within the United States it is built of AT&T facilities consisting of OC-48 (2.5 Gbps) and OC-192 (10 Gbps) trunk facilities, which validate and serve as the baseline topology for the AT&T ATM simulation network built within OPNET for this research [ATT03]. The AT&T network that is of interest to this research is the continental United States and the destination cities are derived from the information obtained from all three POP providers. Figure 2.4 shows the AT&T IP network present in North America and the World. AT&T RTT between US networked cities in Figure 2.5 serves as a validation of the data received from the simulation data collected. Figure 2.5 shows AT&T IP network delay statistics as of 22 May 2003.

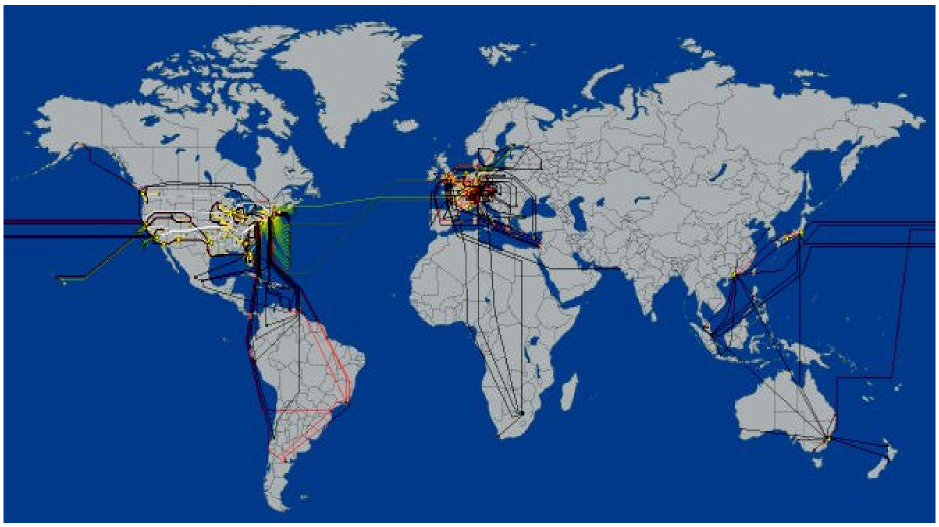

Figure 2.4 AT&T IP Global Network Map

Results for Selected City

City of Origin Chicago   *Round trip delay in ms

Destination	Delay* ms	Loss %	Destination	Delay* ms	Loss %
Atlanta	35	0.0	New York	20	0.0
Austin	25	0.0	Orlando	46	0.0
Cambridge	23	0.0	Philadelphia	22	0.0
Chicago			Phoenix	41	0.0
Dallas	21	0.0	San Diego	50	0.0
Denver	17	0.0	San Francisco	41	0.0
Detroit	6	0.0	St. Louis	7	0.0
Houston	26	0.0	Seattle	45	0.0
Los Angeles	47	0.0	Washington	24	0.0

Thresholds are distance sensitive

Figure 2.5 AT&T IP Network Delay Statistics

2.5.4 Point of Presence Issues Some of the issues that arise when attempting to geolocate a node across the Internet is the location of the servicing POP. Often a servicing POP is located miles away from the node of interest. For example, the POP for an ISP servicing Beavercreek, OH is located in Chicago, IL [Car03]. This makes it difficult to track a node in another part of the country from Beavercreek, since the latency produced will always be biased due to the POP in Chicago, IL. The distance from Chicago to other major cities can be determined, but a bottleneck exists and eliminating the bottleneck latency from a node to the POP is problematic, Figure 2.6 is an example.

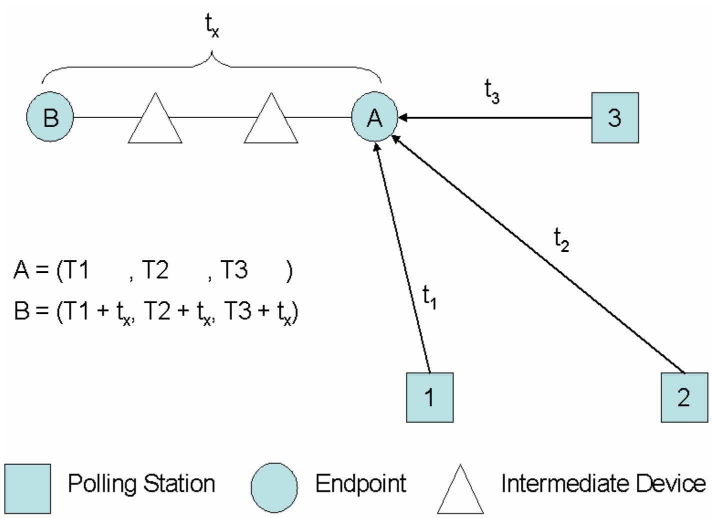

Figure 2.6 Bottleneck Example

2.6 Previous Research

2.6.1 National Security Agency (NSA) Network Geolocation Network Geolocation Technology is 'the ability to physically geolocate a logical network address across the net [NSA02]. Latency data from a private network that spanned the continental U.S. was obtained to perform geolocation analysis. Nodes used for the latency measurements were located within a single AS, so there were no latency effects due to crossing AS boundaries. Latency is determined by calculating a RTT from the source to the destination node. The four sources of network latency are line speed, queue size, switching speed, and physical separation. These latencies are not calculated, but compensated for. After they are compensated for a time-to-location relation can be established.

Latency due to line speed was compensated for using a slope-intercept graph as shown in Figure 2.7. The dots in the figure represent a single RTT of a packet produced by the ping utility sent repeatedly for a given packet size. Pings are sent using increasing packet sizes and the data is recorded as shown in Figure 2.7. The slope-intercept formula for a line is y = mx + b, where y is the latency, m is the slope of the line shown in Figure 2.7 below, x is the packet size and b is the theoretic latency of a zero byte packet. Using the minimum latency for a given packet size, a straight line can be drawn that intercepts the y-axis at b. The slope m is inversely proportional to the packet size, that is, if the packet size increased so did the latency. Due to the inverse relationship of the packet size to the delay and utilizing a static bandwidth; the "latency" of a zero byte size packet can be estimated. Thus, latency due to line speed has been compensated for.

It was determined through empirical measurements that the probability of a packet traveling through a switch in two milliseconds is 0.95 [NSA02]. This figure is used to compensate for queuing delay. Thus, only city level resolution can be achieved using this method.

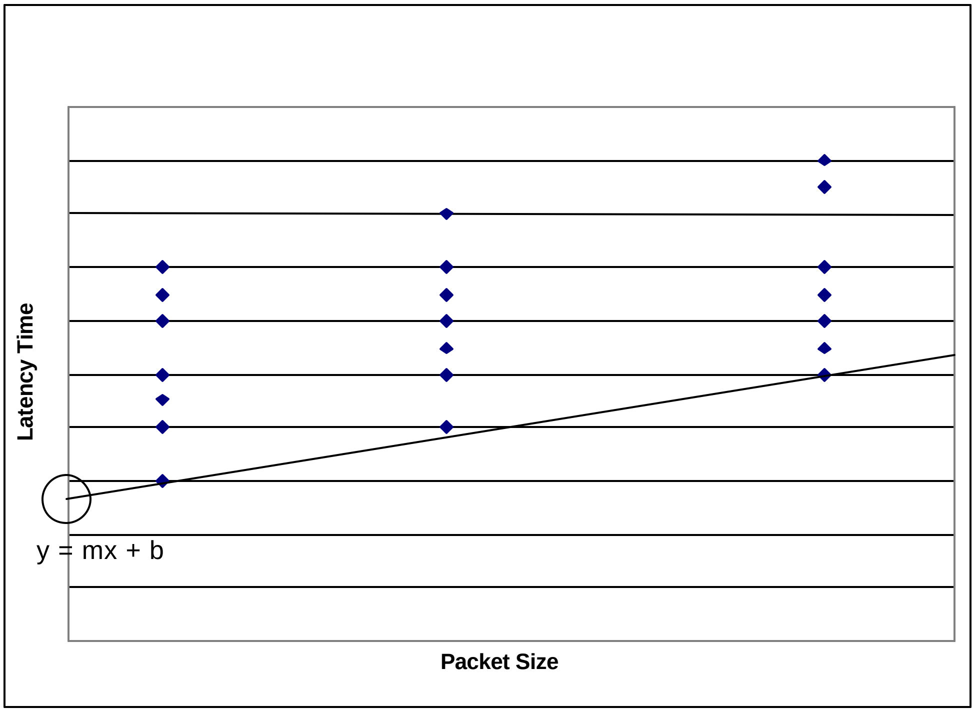

Figure 2.7 Latency and Line Speed Slope Intercept Graph

There is no RTT to distance correlation for a packet traveling on the Internet based on data collected from over 200 nodes worldwide [NSA02]. There are however, some things that can be inferred about distance using RTT. Lines can be used to exclude areas of the globe the node could not have been reached in the time frame given. In Figure 2.8, the sloped small dashed line is the shortest RTT required for light to travel x degrees along a great arc route. RTTs less than 100ms can be used to exclude certain areas of the earth (those greater than x degrees distance from the source) as possible locations of a node. The level large dashed line at 134 ms is the time it takes to encircle the globe at the equator traveling at the speed of light. The top line (dot-dot-dash) at 478 ms, is the time a packet requires to make a geo-synchronous satellite hop. RTT measurements below that line means a packet did not traverse a satellite hop.

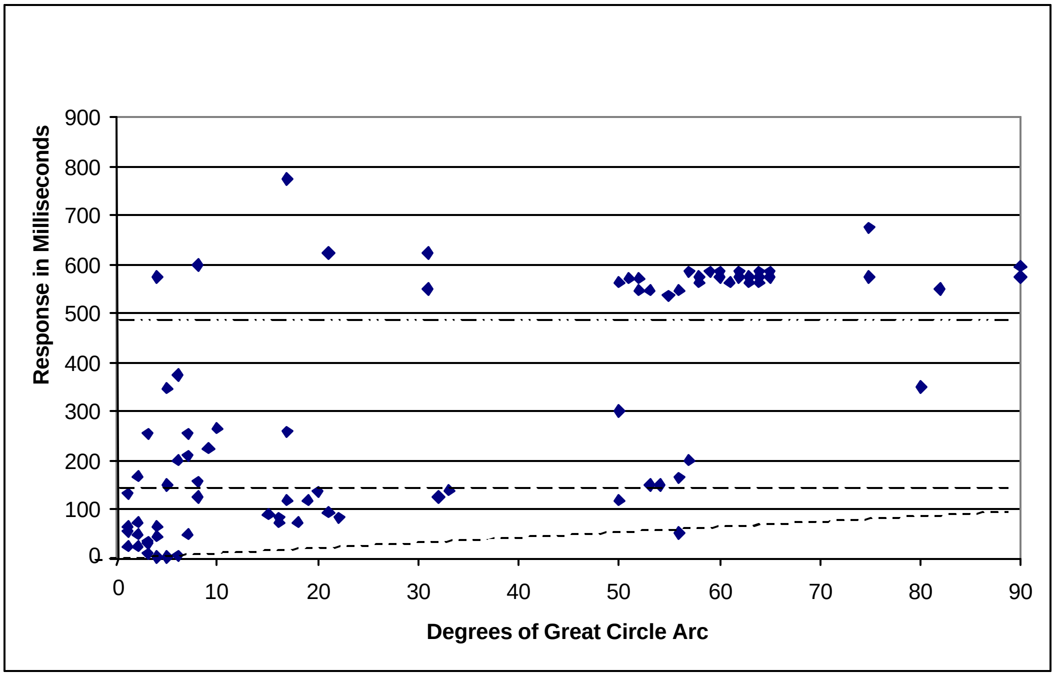

Figure 2.8 Why Time to Distance Does Not Work

Due to the arbitrarily long delays a packet may suffer, no time to distance correlation can be established. However it may be possible to establish a correlation between time and a location because latency to a particular node, while not corresponding to a distance, may be consistent enough to serve as a "marker" to that location. In Table 2.1 a Euclidean distance is calculated using RTT. The first column lists the endpoints or destination locations targeted by each of the stations in column 2, Cambridge and column 3, Palo Alto. The Euclidean distance shown in column 4 was calculated by taking the squared difference between column 2 values and polling station, Cambridge; adding the squared difference between column 3 values and polling station, Palo Alto then taking the square root of that sum.

Table 2.1 Time-to-location

End Points	Polling Stations		Euclidean Distance
	Cambridge	Palo Alto	
Cambridge	**3.47**	**79.05**	**107.54**
NYC	9.31	76.95	101.97
Washington DC	15.28	81.39	101.37
Atlanta	31.76	68.98	81.49
Denver	43.84	34.92	47.85
Dallas	49.04	43.27	50.54
Los Angeles	68.10	12.67	14.94
Oakland	72.38	5.62	7.48
Palo Alto	**79.31**	**2.80**	**0.00**
San Jose	81.47	4.61	2.82

The following example will clarify the Euclidean distance formula by finding the distance between Palo Alto and Cambridge:

$$(3.47 - 79.31)^2 = 5751.71$$

$$(79.05 - 2.80)^2 = 5814.06$$

$$\sqrt{(5553.23 + 5856.84)} = 107.54$$

The Palo Alto reference point is used for all Euclidian distances, in Table 2.1, Cambridge and Palo Alto are used as a polling station reference points. The Chicago reference point is actually physically located in Palo Alto according to the Euclidean distance. NSA confirmed this data labeling mistake with the network data owner and demonstrated the Euclidean distance between Palo Alto and New York City is 101.97.

2.6.2 Reverse Geographic Location of a Computer Node Fundamental issues for network geolocation have been identified [Car03]. Network routing issues were identified as a factor for proving time to distance not working. The routes that physical networks take from city to city differ greatly from network vendor to network vendor. A bottleneck of network latency traveling between the actual computer nodes location and entrance into the Internet at the POP is also a problematic latency effect. The time-to-location based on RTT is based on a relationship of temporary signature delay times. These signature times and the slope-intercept method for determining the zero byte packet travel time are thought to hold the key to reaching a time-to-location solution for reverse geographic location methods. These are the areas of interest to this research in demonstrating the NSA time-to-location algorithm. Identifying the sources of variability for a packet traveling across multiple vendor networks is identified as a future research area and this research will demonstrate that link bandwidth must be taken into account for crossing over a multiple AS network.

2.7 Summary

This chapter discussed various Internet Protocol characteristics and utilities. IP Multicast traffic and fragmented IP traffic characteristics were discussed, followed by multiple utilities including Ping, Whois, Traceroute, GTrace, Pathping and Skitter. Third Party addresses and Distance Metrics in the Internet were then examined along with the function of the Bellman-Ford Algorithm. After that Cyberspace Geography was discussed to include Naive and Social geography. Then the geographic origin of IP traffic was examined and techniques to visualize the Internet along with geolocation and Location-based Authentication techniques. Points of presence of three commercial vendors, MCI, Sprint and AT&T were discussed along with issues in dealing with POP. Previous research on the subject of geolocation was discussed last and included National

Security Agency (NSA) Geolocation and Reverse Geographic Location of a Computer Node research conducted at the Air Force Institute of Technology.

III Methodology

3.1 Background

In past conflicts military commanders at all levels were taught to dominate the air, sea, and land. A new spectrum has become highly important in the past few years, the information systems that we use to send our general orders and other highly valuable information to and from command centers around the world. In order to protect these systems tight security measures must be taken. One way to assist in increasing the security of these systems is to be able to locate a hacker, no matter where they are attacking from. An enemy has the capability to exploit these assets either by hacking into them and gaining access to DOD information or disrupting access by authorized users to this information through denial of service attacks. The first step in being able to do this is to geographically locate an attacking computer node from a distant location. The ability to do this rapidly and reliably is a good first step to a strong deterrent from being hacked in the future.

3.2 Problem Definition

3.2.1 Goals and Hypothesis The goal of this research is to determine the geographic location of a node using only packet latency measurements. Baseline physical distances between major cities are used to ensure that a time minimum or t(min) is established from city to city in which a packet round trip time measured cannot be less than and to establish a parameter that remains a constant. It is expected that line speed will converge to a linear slope and provide the time to transmit a theoretic “zero byte” packet (which is independent of line speed) leaving only packet size to use as a factor. Latencies in the network will affect the slope of the line that produces a theoretical “zero byte” packet, but the packet sizes will not. The slope of the line that produces a theoretical "zero byte" packet will only rotate around the point of intercept on the y axis and always provide a consistent and reliable time-to-location algorithm output. A hypothesis is made that using correlations developed on multiple, AS networks, a time-to-location will also effectively correlate geographic locations across a multiple AS network.

3.2.2 Approach To meet the goal established in this research of a time-to-location algorithm; a strategy of identifying and characterizing latency sources using a simulation model is developed. OPNET, version 10.0 modeling and simulation software is used to develop the network models. Only information that can be obtained from the Internet is used to determine the configuration and setup the OPNET model for the network simulation.

An OPNET network model will be used to control and verify the original NSA time-to-location results [NSA02]. The time-to-location algorithm uses the round trip time from a polling node to multiple distant nodes on the network. The Euclidean distance is determined from an unknown node to all known polling nodes. Using this data, a time-to-location correlation will be established for a single AS network using latency measurements.

The baseline physical distance between cities on the networks are established using driving distances between cities obtained using city center to city center distances as destinations [Map03]. The AT&T simulation model uses ATM switches as a baseline for long haul communications. ATM will provide a constant bandwidth for the simulations. MCI provides bandwidth information for their network, but a constant Fast Ethernet bandwidth is used in the MCI model to provide a separate contrasting topology for simulation result comparison. An analytic model using Euclidean distance measurements is established based on physical distances establishing a minimum time or t(min) baseline between cities.

The first AS network simulation is based on an AT&T IP network model. Latency measurements from AT&T's network establish a set of "true" Euclidean distances between the same cities as the analytic model [ATT03]. Based on this information AS network simulations collect metrics used to develop a geographic location baseline from city to city and identify latency "invariants" within the network simulation for comparison to multiple AS network simulation testing results.

MCI only provides country to country or continent to continent latency statistics, so distance latency measurements for MCI simulations are based on AT&T statistics under the assumption that commercial vendor networks have similar latencies. The MCI network model is based on a MCI IP network model. Both the AT&T and MCI network model setups are obtained from the public web-sites of the aforementioned companies [ATT03, MCI03]. This model is used to identify the source and nature of these latencies and establishing a time-to-location correlation.

3.3 System Boundaries

The system under test (SUT) is illustrated in Figure 3.1. The system includes all latencies associated with the Internet and packets traveling across the Internet. The scope of the simulations is the continental U.S. boundaries for networked cities. The component under test (CUT) is the time-to-location algorithm, which is developed with data captured from the simulations. The algorithm uses all latency sources to include: queuing delays, switching speeds, line speeds and physical distances. The CUT includes the identified latency components that effect RTT measurements and the Euclidean distances. The polling node geographic locations are known to allow validation with the Euclidean distances obtained from the OPNET simulation data. The time of day workload is limited to the normal daytime working hours and nighttime hours of the continental U.S.

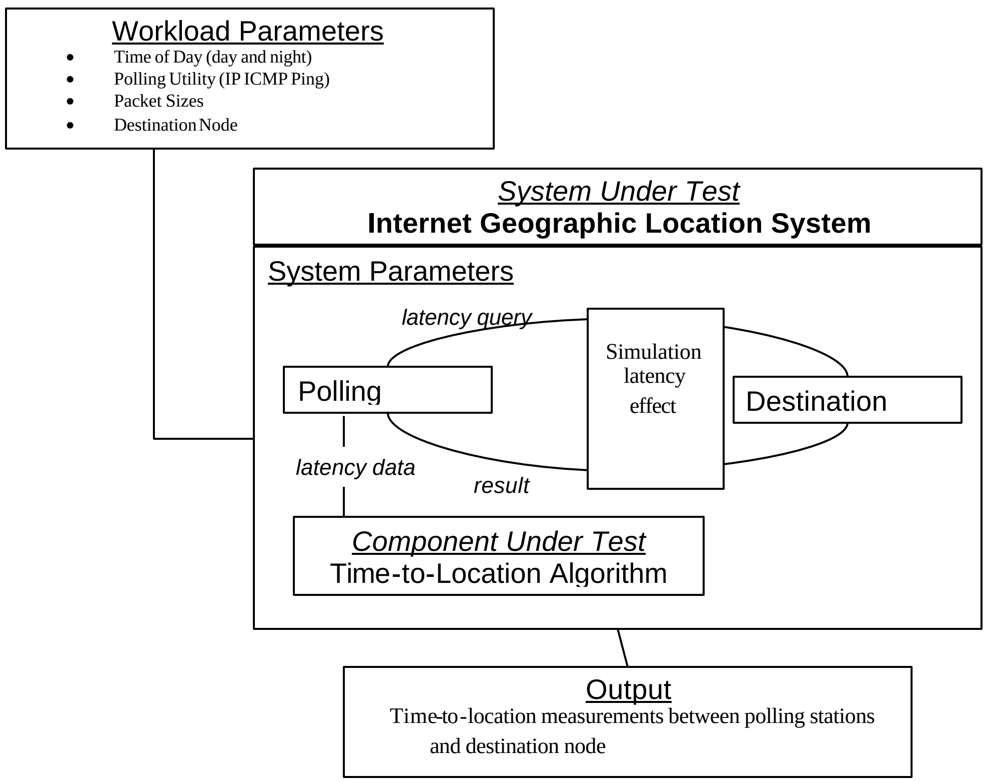

Figure 3.1 System Under Test

3.4 System Services

This system provides an Internet geographic location of a computer node. The basic service offered is the identification of the geographic location of a computer node using a time-to-location algorithm. It provides this service by identifying and measuring varying packet latencies and establishing variable and constant latencies in the time-to-location algorithm. A constant minimum baseline Euclidean distance is established to show correlation in the geographic location of the cities involved.

The results of the system services are limited to success or failure. The success or failure of the algorithm is based on the precision to which a city can be geographically located using packet latency measurements over a single AS network with a 95% probability of being within a 2ms mean of the cities geographic location and thus matching results achieved in the previous NSA study. The second success or failure of the algorithm is based on how precisely a city can be geographically located using packet latency measurements over a multiple AS network.

3.5 Performance Metrics

The performance metrics are the correct identification of polling node geographic location and accuracy. The polling node geographic location is based on the results of the time-to-location algorithm returning a result from at least two known distant locations. The resolution of this measurement is city to city resolution. The accuracy of the algorithm is based on a 95% probability of the latency measurements being within a 2 ms mean difference of each other and how close the results actually are to the known location of the originating computer node.

3.6 Parameters

3.6.1 System Parameters The system parameters are the polling node, the polling node distance to the Internet backbone location, the topology of the Internet, the background load, simulation latency effect (the latency measurements of queuing delay and switch speed), and finally the time of day workload associated with the Internet. The system

Table 3.1 System Parameters

Distance to the Internet
Internet Topology
Background Load
Simulation latency effect
Polling node

parameters are displayed in Table 3.1. The number of source nodes will be fixed within each simulation and is only changed to validate the accuracy of a given time-to-location algorithm. The polling node is the originating node of a ping packet, which is sent out to multiple destination cities. The polling node distance to the Internet backbone location is based on the location of the polling node being used in the simulation. The OPNET standard wide area network (WAN) model is used to establish a city wide network for every simulation city created. The computer lab within the WAN is used as the polling node for each simulation and the WAN router is used as the destination of each city to city ping packet. The topology of the Internet is based on the AT&T and MCI simulation models produced from the Internet web sites. The arbitrary background load is fixed based the topology of the network. The AT&T ATM network uses a Paretto distribution of a 25% arrival rate bandwidth load for nighttime hours and 100% arrival rate bandwidth load for daytime hours. The MCI Fast Ethernet network uses a Paretto distribution of a 25% arrival rate bandwidth load for nighttime hours and 80% arrival rate bandwidth load for daytime hours.

3.6.2 Workload parameters Latency measurements are affected by queue delays and switch speeds, along with physical distances and line speeds. The time of day workload varies emulating people arriving at work. There is a High workload for normal working hours and a Low workload at night or off work hours. Packet sizes are varied, but will never exceed a 53 bytes size packet (ATM packet size) to ensure no segmentation is induced. The line slope equation used is y = mx + b; m is the slope of the line and is equal to the formula:

$$m = \frac{\sum xy - n\overline{xy}}{\sum x^2 - n\overline{x}^2} \qquad (3.1)$$

Where n is the number of packet sizes, $\overline{x}$ is the mean of the packet sizes and $\overline{y}$ is the mean of the RTTs. The minimum time for each size packet establishes the slope of the line, as shown in Figure 3.2. Using this slope, the y intercept or b, can be determined. This b is the theoretical zero-bytes size packet round trip time and is used to eliminate the effect of line speed from the time-to-location algorithm.

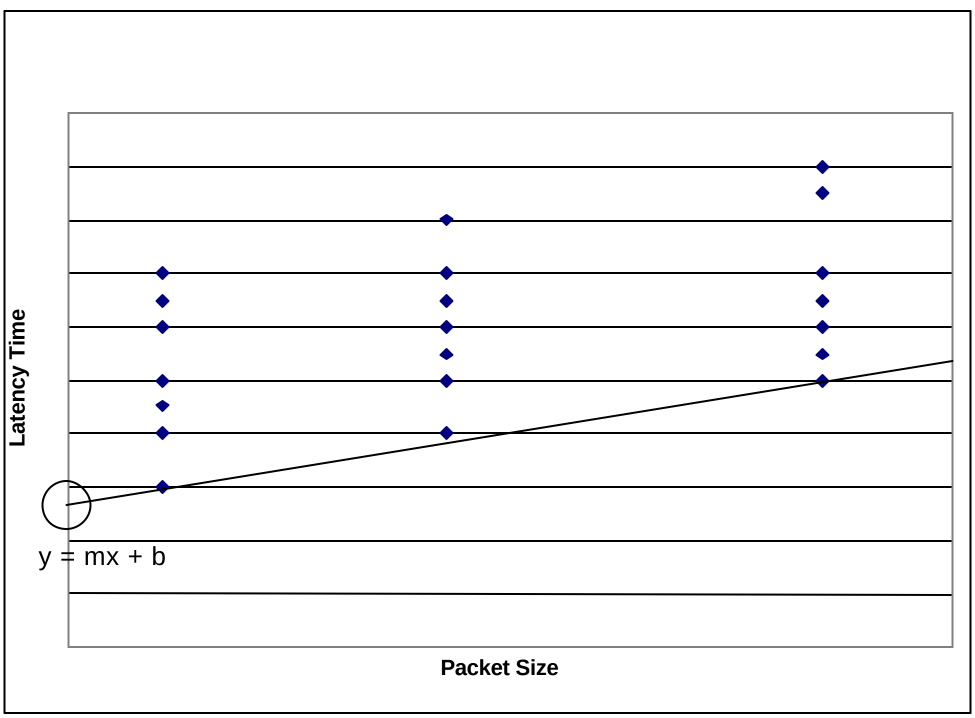

Figure 3.2 Latency and Line Speed Slope Intercept Graph

3.7 Factors

Polling node geographic locations change between simulations and multiple destination locations are used to validate RTT results. The use of queuing delay and switching speed is reduced to a single factor for the AS network model OPNET simulation; a 2 ms mean is expected to account for queue delay. The queuing delay more than accounts for the switch speed in today's switches running in the gigahertz and faster speeds [NSA02].

Thus, queuing delay and switch speed is exponentially distributed with a mean of 2 ms for the AS network models. In the multiple AS network model the bandwidth is altered in the crossover segment between commercial vendors by changing the link bandwidth between T1(1.54Mbps), OC-3(51Mbps) and OC-12(622Mbps) links passing traffic between the two modeled networks.

To verify the NSA results, two workloads are used to provide background latency. Time of day workload for the nighttime hours is set to a 25% Paretto distribution arrival rate; for the daytime business hours the workload is set to a 100% Paretto distribution arrival rate for the ATM network and a 80% Paretto distribution arrival rate for the Fast Ethernet. The final factor is the packet size, which is set such that no segmentation occurs. This provides a single minimum latency time and not a bi-modal latency due to packet segmentation on certain links. Packet sizes are set to 16 bytes, 32 bytes, and 53 bytes. Repeating the NSA data evaluation, simulations are run for 20 repetitions [NSA02].

3.8 Evaluation Technique

To evaluate the system two evaluation techniques are used. The first technique is an analytic model based on the physical distances, line speeds and major cities that have connections involving both the AT&T and MCI IP networks. An analytic study is an appropriate evaluation technique to use because of the Euclidean distance measurements generated between geographically separated cities to verify time-to-location algorithm data output.

OPNET modeling and simulation software is used to evaluate the individual factors in a completely controlled environment. The modeling and simulation software allows individual factors to be changed and the ability to evaluate the combined queuing delay and switch speed as well as the time of day workload per previous NSA research [NSA02]. These factors effect the RTT measurements and also the Euclidean distance results that actually geographically locate the computer node in the Internet. The two evaluation techniques combined provide control over factors to validate measurements and verify the results received for the metrics.

3.9 Workload

The workload is determined by the factors that are specified in Table 3.2, and Section 3.7 with the SUT. The queuing delay and the switch speed is expected to be exponentially distributed with a mean of 2 ms for the AS network and exponentially distributed in the multiple AS network models to demonstrate the crossover latencies between networks and switching between network routes and domains on the Internet. The time of day workloads are set to two levels for an Internet traffic load. The times are based on normal working hours for the different time zones, the nighttime hours with a 25% Paretto distribution arrival rate workload and a 100% / 80% Paretto distribution

Table 3.2 Factors to be varied

Factors	Variables
Time of day workload	AT&T ATM Network High(100%) for day time Low(25%) for night time MCI Fast Ethernet Network High(80%) for day time Low(25%) for night time
Packet Sizes	16 bytes 32 bytes 53 bytes
Polling Node locations	Boston / Cambridge Chicago San Francisco
Destination Node locations	Atlanta Austin Boston / Cambridge Buffalo Chicago Dallas Denver / Aurora Detroit Houston Kansas City Las Vegas Los Angeles Miami New Orleans New York Orlando Philadelphia Phoenix Pittsburgh Raleigh Salt Lake City San Diego San Francisco St Louis Tampa Washington DC

arrival rate workload for daytime hours. The packet size is varied to allow for various network hardware topology effects and to account for bandwidth and physical distances. The packet sizes are set to 16 bytes, 32 bytes, and 53 bytes to prevent any packet segmentation induced latency. NSA determined that 20 replications with a 95% confidence interval would be required for a Low and High workload to obtain consistent results [NSA02].

3.10 Experimental Design

A full factorial experiment of 120 simulations is conducted for the single AS network simulations. The first parameter has one level, queuing delay and switch speed with an expected exponential mean of 2 milliseconds. The first factor time of day has two levels: a Paretto distribution of 25% workload for Low and a Paretto distribution of 100% / 80% workload for High. The second factor packet size has three levels: 16 bytes, 32 bytes, and 53 bytes to traverse the network returning a round trip time. The third factor polling node location and the fourth factor destination node location will both change based on geographic locations of cities used.

A partial factorial experiment of 16 simulations is conducted for the multiple AS network simulations. The first parameter has one level: queuing delay and switch speed with an expected exponential mean of 2 milliseconds. The second parameter becomes the polling node location: only San Francisco is used for the multiple AS network simulations. The first factor time of day has two levels: a Paretto distribution of 25% workload for Low and a Paretto distribution of 100% / 80% workload for High. A second factor packet size has two levels: 16 bytes and 32 bytes packets on the T1 link to solely determine if the linear slope formula reacts the same as in the single AS network.

The third factor becomes the link bandwidth of T1, OC3 and OC12 to interconnect the single AS networks creating a multiple AS network. The third factor destination node location will change based on geographic locations of cities used.

3.11 Summary

In this chapter, the experimental methodology is outlined. Based on the goal to determine a time-to-location; a strategy of characterizing latency sources using simulations. The approach to achieving the goals is discussed and the system boundaries are defined in Figure 3.1 and include all latencies associated with the Internet and packets traveling across the Internet. System services and performance metrics related to the system are also described.

Based on this methodology system and workload parameters are selected to define the system in more detail. Factors selected from these parameters and workload levels are described to identify the packet latency issues. The evaluation techniques chosen are an analytic model and an OPNET simulation model. After selecting the repetitions and types of experiments to run, an analysis technique is put in place to achieve the requested confidence interval. This chapter presents the methodology and the approach of the thesis, establishing a basis to interpret the results in a meaningful way.

IV. Results and Analysis

4.1 Overview

This chapter provides an overview of the analysis methods used in the OPNET simulations to evaluate the geographic time-to-location of an Internet node algorithm. The first analysis is used to determine the minimum latency and Euclidean distance measurements between 26 cities on the AS AT&T and MCI network models built within OPNET. Tables A.1 and A.2 demonstrate the simulation model network behavior is as expected in a real world network with propagation delays and background load effecting RTT. The theoretical "zero" bytes size packet measurements are used to determine the Euclidean distances.

The second analysis is conducted to determine the sources of additional latency when attempting to geographically locate a node on a multiple AS network model, in this case combining the AT&T ATM network and the MCI Fast Ethernet network into one 73 city multiple AS network. This network model is used to analyze the response time of the original 26 destination cities from each network. In the initial analysis of a multiple AS network model, 16 bytes and 32 bytes packets are used to establish the results of the same calculations eliminating line speed as on a single AS network, which is demonstrated in Table A.3 and A.4. A comparison of minimum latency measurement is used to determine any differences in topologies or link bandwidths.

4.2 Time-to-location Algorithm for AS network

OPNET uses the Bellman-Ford algorithm to compute the shortest path routing within the simulations. This algorithm is used for dynamic routing in networks that use automatic fault recovery techniques, such as Internet service providers. The AT&T ATM simulation links are setup to demonstrate the bandwidth of commercial vendor traffic [FML03]. Ping requests are sent every 3 seconds for a total of 300 simulation seconds. The first 100 simulation seconds are used by OPNET to setup the routing on the simulation network, leaving the last 100 simulation seconds to return 67 ping RTTs for analysis. The pilot network uses OC-12 (622 Mbps) links to demonstrate ATM traffic RTT using the specified background loads.

Background traffic arrives according to a Paretto distribution with a load of 100% during business hours using a OC-12 (622MBps) traffic load and 25% for nighttime hours using an OC-1 (51Mbps) background load on the network, Figures 4.1 and 4.2 show the throughput for 25% and 100% loading, respectively.

To conduct the actual data collection, the OC-12 links were changed to OC-48 (2488Mbps) connecting all cities, except the San Francisco to Chicago direct link which is one OC-192 (9952Mbps) link to more accurately model the networks used by commercial vendors to handle Internet traffic loads [FML03]. The AT&T ATM network map connecting 29 cities throughout the continental United States is shown in Figure 4.3.

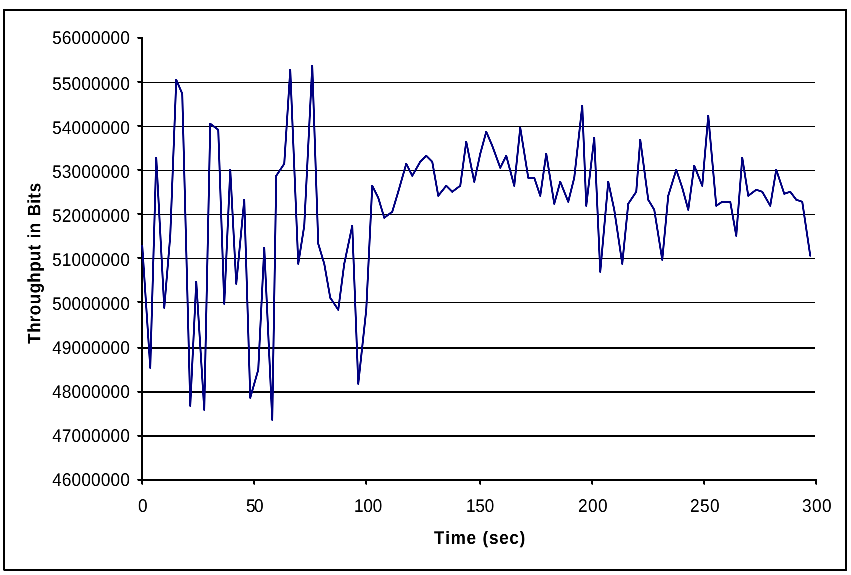

Figure 4.1 25% Paretto Distribution for AT&T Links (Low Background Load)

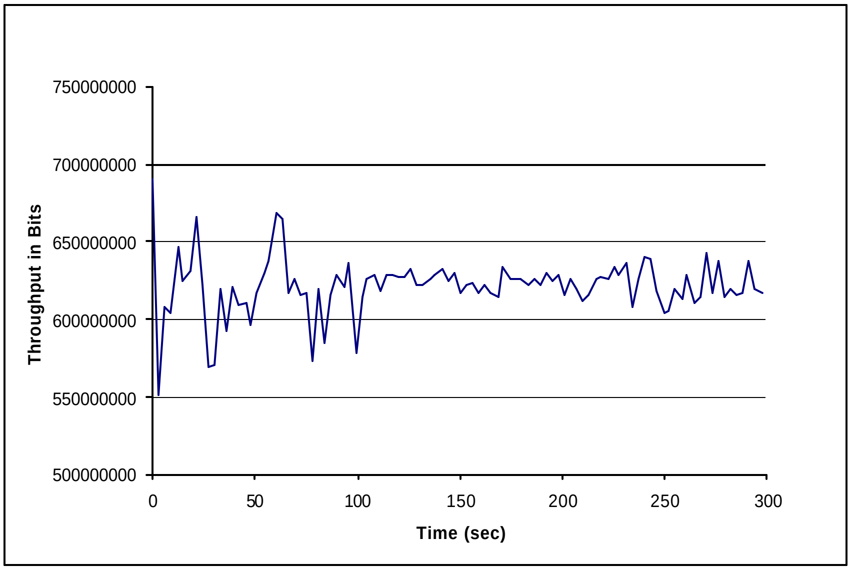

Figure 4.2 100% Paretto Distribution AT&T Links (High Background Load)

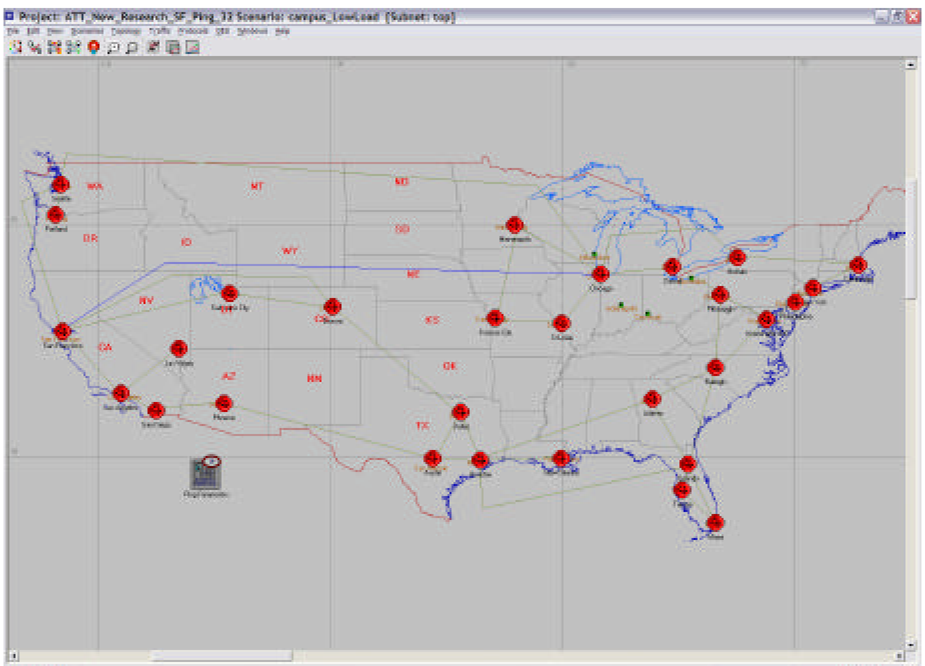
Figure 4.3 AT&T ATM OPNET Simulation Model

In the MCI simulation, a Fast Ethernet Model network is used to demonstrate the difference in topologies and network routing in contrast to the AT&T ATM network model. Ping requests are sent out every 3 seconds for a total of 300 simulation seconds. The first 100 simulation seconds is used by OPNET to setup the routing on the simulation network, leaving the last 200 simulation seconds to return 67 ping RTTs for analysis. The MCI network uses 2 - 100 Mbps bandwidth links to connect 44 cities throughout the continental United States, Figure 4.4 shows the MCI Network model.

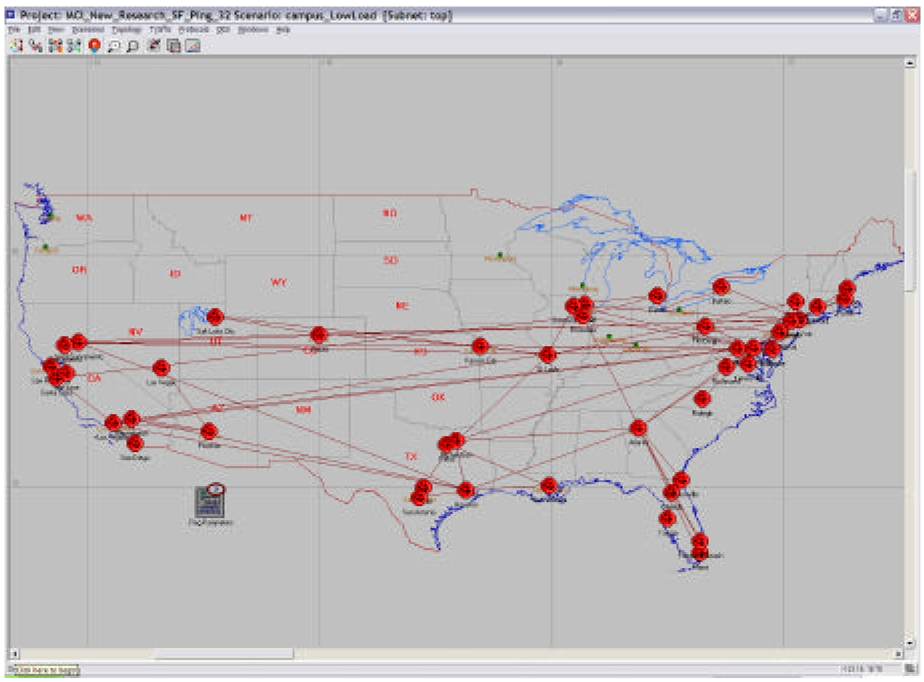

Figure 4.4 MCI Fast Ethernet OPENT Simulation Model

Background traffic arrives according to a Paretto distribution with a load of 80% during business hours using an OC-3 (155Mbps) traffic load and 25% for nighttime hours using an OC-1 (51Mbps) background load on the network, Figures 4.5 and 4.6 shows the throughput for 80% and 25% loading respectively.

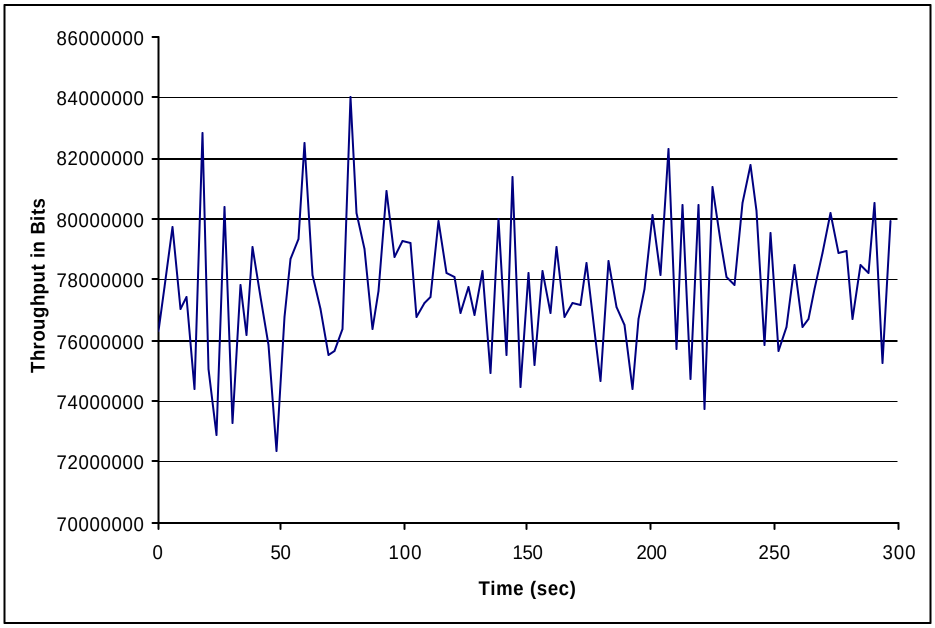

Figure 4.5 80% Paretto Distribution for MCI Network Links (High Background Load)

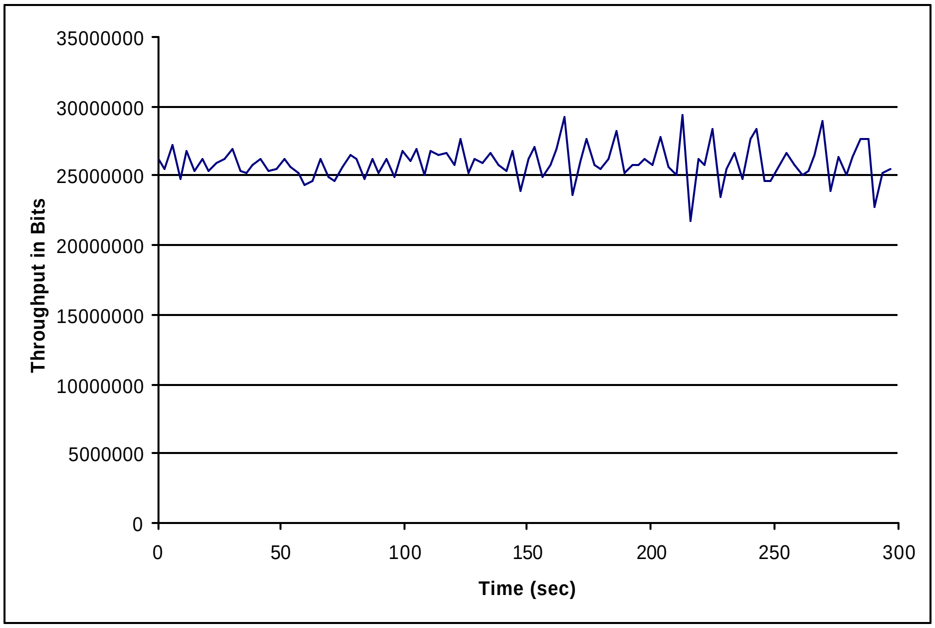

Figure 4.6 25% Paretto Distribution for MCI Network Links (Low Background Load)

4.3 Time-to-location Algorithm

To calculate a time-to-location for a single AS network node, 4 issues have to be addressed: line speed, queue size, switching speed and physical separation.

4.3.1 Line speed Line speed is addressed by using the y = mx + b line equation. Using the y = mx + b equation, b is the y intercept around which the linear slope will rotate and is the theoretical zero byte packet value. The AT&T and MCI network simulations were run with a 16, 32, and 53 bytes packet size using both High and Low background loads. This produces the minimum times for 16, 32 and 53 bytes packet RTTs to construct a linear slope regression model to produce the theoretical zero byte packet RTT for each polling node to each of the destination cities.

This "zero" bytes RTT is the data point for the time-to-location algorithm. In Figure 4.7 the AT&T network shows Chicago's theoretical zero byte packet response time and the Figure 4.8 shows Chicago's theoretical zero byte packet on the MCI network. The method is the same although results vary for both simulation networks. The standard error mean of the data results for both networks is approximately 120 nanoseconds, which is insignificant in comparison to the required 2 millisecond mean required for city to city level resolution.

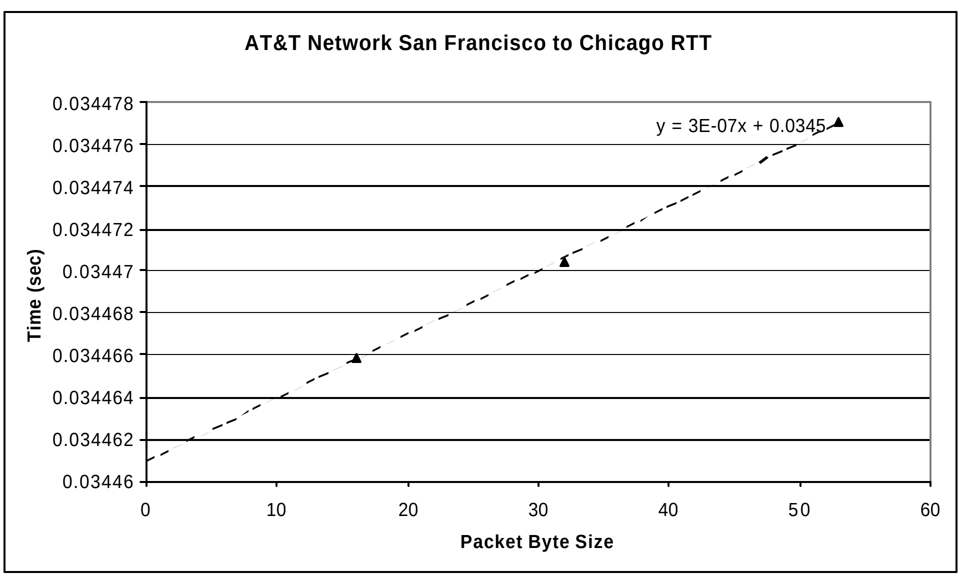

Figure 4.7 AT&T Minimum Linear Slope

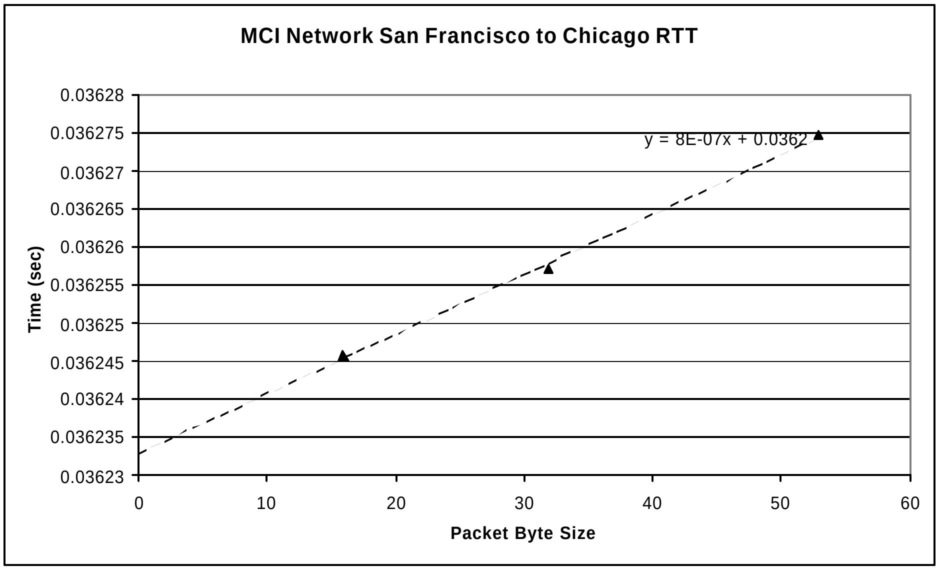

Figure 4.8 MCI Minimum Linear Slope

The linear slope is calculated by using the following formula and steps [Jai91]:

$$y = mx + b \tag{4.1}$$

$$m = \frac{\sum xy - n\overline{x}\overline{y}}{\sum x^2 - n\overline{x}^2} \tag{4.2}$$

and

$$b = \overline{y} - m\overline{x} \tag{4.3}$$

where

1) Number of packet sizes

$$n = 3 \tag{4.4}$$

2) Mean of the simulation packet sizes

$$\overline{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) \tag{4.5}$$

$$= \frac{16 + 32 + 53}{3} \quad = 33.67$$

3) Mean of the minimum RTTs

$$\bar{y} = \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right) \qquad (4.6)$$

$$= \frac{0.034466 + 0.03447 + 0.034477}{3}$$

$$= 0.034471$$

4) Sum of the packet and RTT products

$$\sum xy = \sum_{i=1}^{n} x_i y_i \qquad (4.7)$$

$$= (16 \times 0.034466) + (32 \times 0.03447) + (53 \times 0.034477)$$

$$= 3.48178$$

5) Sum of the square of each product

$$\sum x^2 = \sum_{i=1}^{n} x_i^2 \qquad (4.8)$$

$$= 16^2 + 32^2 + 53^2$$

$$= 4089$$

6) Slope of the line

$$m = \frac{\sum xy - n\overline{xy}}{\sum x^2 - n\bar{x}^2} \qquad (4.9)$$

$$= \frac{3.48178 - (3 \times 33.67 \times 0.034471)}{4089 - (3 \times 33.67^2)}$$

$$= -0.00000019725$$

7) Theoretical "zero byte" packet, the y intercept

$$b = \bar{y} - m\bar{x} \qquad (4.10)$$

$$= 0.034471 - (-0.00000019725 \times 33.67)$$

$$= 0.034478 \text{ or } 34.5ms \text{ for a theoretical zero byte packet}$$

The zero byte packet RTT for each network is shown in the Euclidean distance tables in Section 4.4 Euclidean Distance, the next issue to address is queue size and switch speed.

4.3.2 Queue Size and Switching Speed Two background loads are used to induce a latency effect of two varying topologies on the packet response times. The AT&T and MCI network models use the High load to demonstrate daytime business hours and the Low load to demonstrate non-business or nighttime hours. The AT&T network example is shown in Figure 4.9 and the MCI network example is shown in Figure 4.10, a power trend line is used to ease visual interpretation. A power trend line is a curved line that is used with data sets that compare measurements increasing at a specific rate.

The minimum sample size and calculation is based on the High and Low load calculations for the final research simulation network. NSA research found the High load to require 11 samples and the Low load to require 5 samples for a total of 16 samples to obtain a 95% probability of being within 2ms of t(min)[NSA02]. NSA theorized that 20 samples would be a general rule of thumb for sample sizes to obtain the required accuracy for city to city resolution [NSA02].

This research based the number of repetitions on the 20 sample rule of thumb. In this research the formula to determine a 95% confidence interval to be within 2ms of a minimum mean was calculated as the sample size calculation. The t(min) is used to provide a constant physical distance latency for the network links between cities.

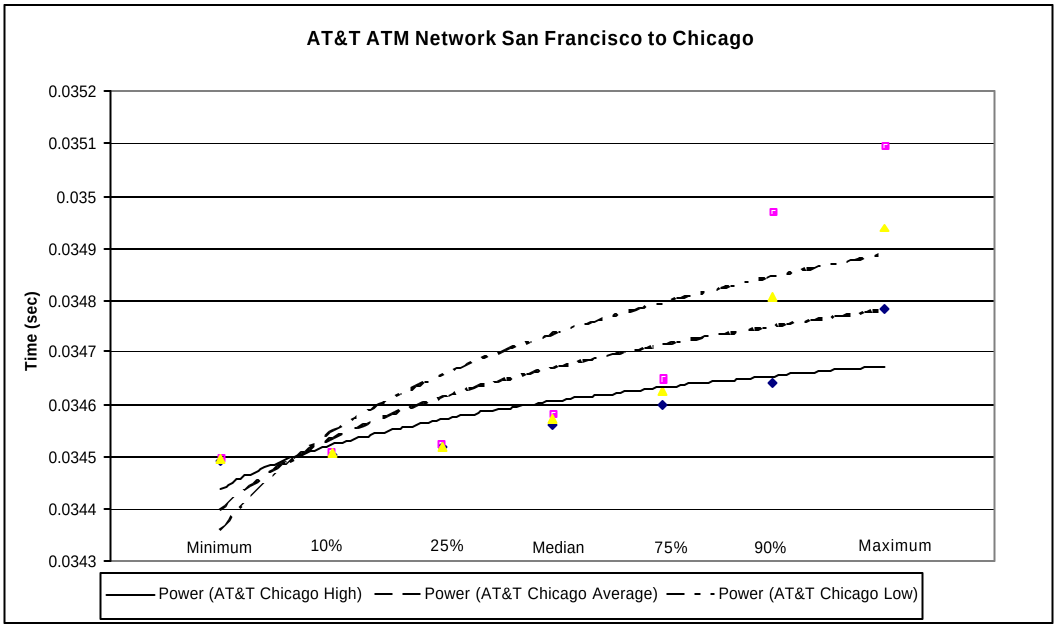

Figure 4.9 AT&T Pilot Network Load RTT

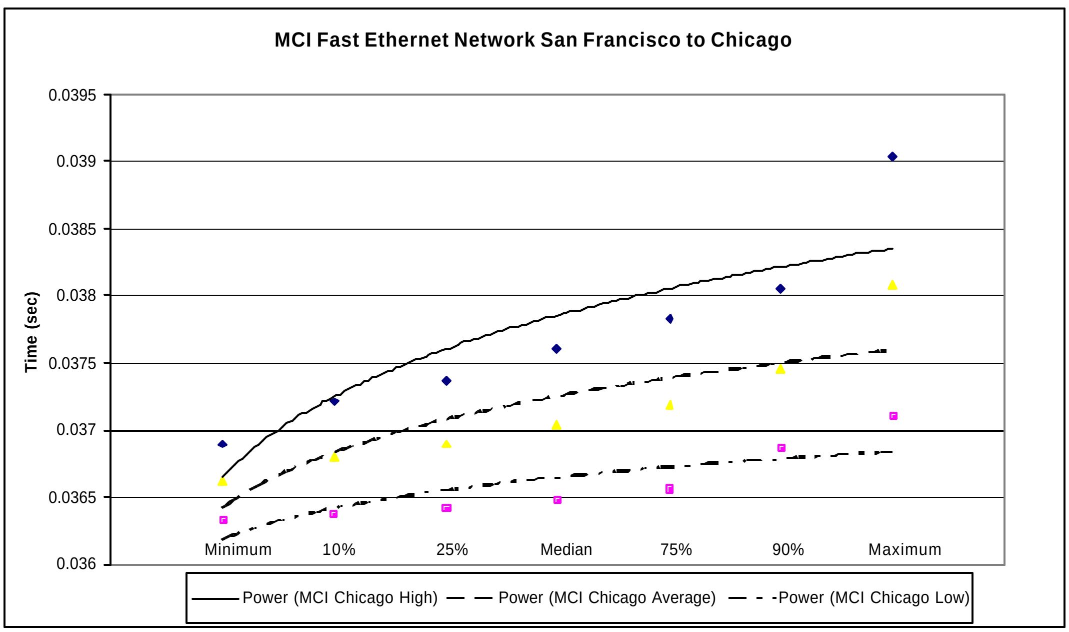

Figure 4.10 MCI Pilot Network Load RTT

The AT&T San Francisco polling station network will be used as an example. The formula and steps followed to obtain the sample size is calculated by the desired confidence interval with that found in *n* observations, we can find n, the sample size, as shown in the steps below [Jai91]:

$$x \pm z\left(\frac{s}{\sqrt{n}}\right) = x\left(1 \pm \frac{r}{100}\right) \qquad (4.11)$$

or

$$n = \left(\frac{100zs}{rx}\right)^2 \qquad (4.12)$$

First calculate the confidence interval and sample standard deviation:

1) 100 (first half of the equation 100(1 – a)% confidence interval)

2) $z = 1.960$ (95% confidence level z value from Table A.3 [Jai91])

3) $s = 0.018128$ (sample standard deviation of the minimum RTTs)

Next use the mean of the minimum RTTs from the sample data:

4) $x = 36.1ms$ (High load)

$x = 36.4ms$ (Low load)

Nest use the required accuracy of 2ms divided by the mean of the minimum RTTs:

5) $r = \frac{2ms}{36.1ms}$ (5.5% for High load)

$r = \frac{2ms}{36.4ms}$ (5.5% for Low load)

Based on all AT&T networks sample size collections the most required is 1340 samples for both High and Low load network per packet. All ping traffic examined is the product of 2 source cities, 2 networks per polling station, 2 background loads per network, 3 packet sizes per background load, 20 seeds per packet, and 67 ping latency measurements per seed for a total of 48,240 samples per destination city. There are a total of 26 destination cities for a total 1,254,240 collected pings.

Queuing delay for the final research simulations was limited due to the bandwidth of the AT&T ATM network being OC48 links between the all the destination cities, except for one San Francisco to Chicago is connected with an OC192 link. Figure 4.11 shows the convergence of the Low and High loads due to the proportionally small background load of an OC12 link. Queuing delay contrasts between the two topologies for this research are demonstrated in Figure 4.11 for the AT&T ATM Network and Figure 4.12 for the MCI Fast Ethernet Network, a power trend line is again used to ease visual interpretation.

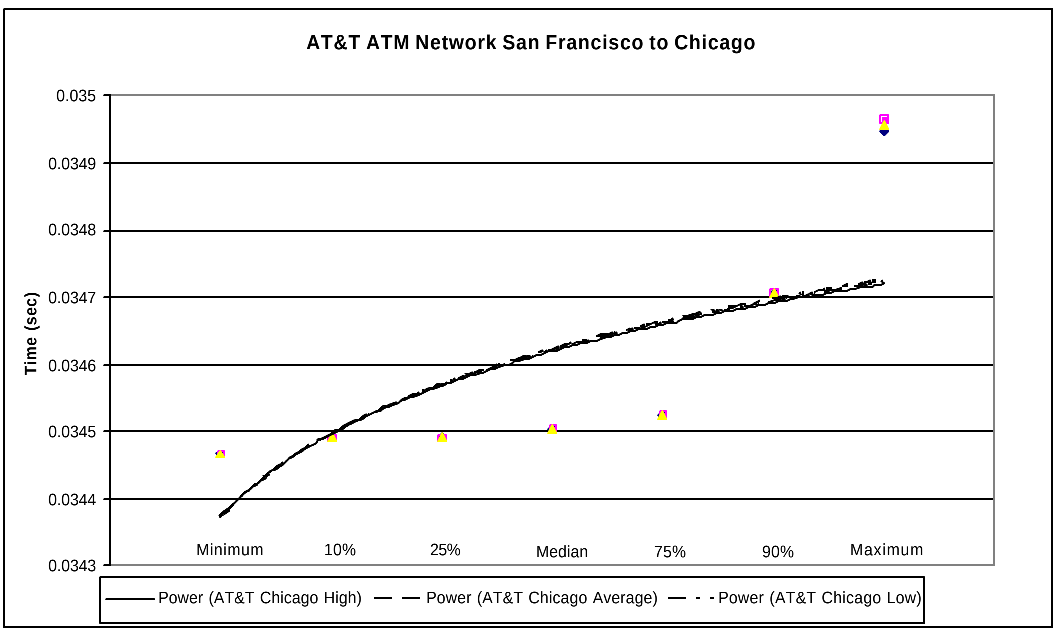

Figure 4.11 AT&T ATM Network Load RTT

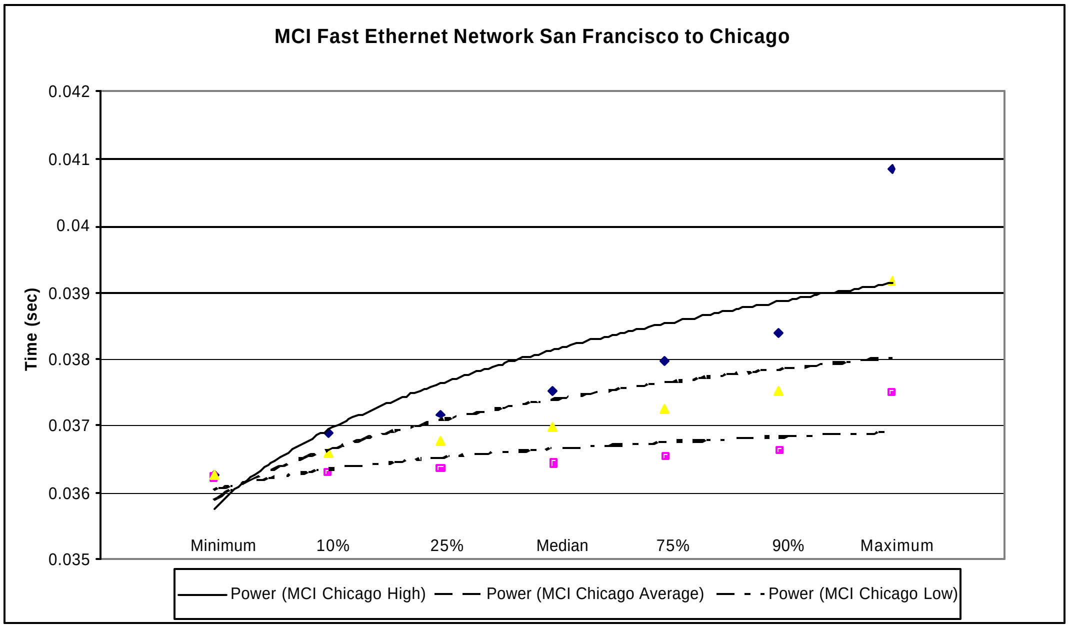

Figure 4.12 MCI Fast Ethernet Network Load RTT

4.3.3 Physical Separation Physical separation is based on polling station to destination city driving distance based on fastest driving routes from the Mapquest website [Map03]. This is not used to demonstrate a time to distance measurement for calculation latencies, but used to model fiber optic cable runs from city to city. It is assumed that fiber optic cable is buried in the Highway right of way for ease of installation, access and maintenance. This also produces a t(min) time for truth values to compare RTTs against and ensure all data collected is realistic. The mileage obtained from Mapquest was used in the OPNET simulations to determine the link delay. The link delay was calculated by taking the mileage multiplied by meters per mile and that product divided by the speed of light in glass. The link delay formula and example for Chicago to San Francisco is:

$$Delay = \frac{mileage \times meters/mile}{speed\_of\_light\_in\_glass} \qquad (4.13)$$

1) $$Delay = \frac{2134 \times 1609}{200 \times 10^6}$$

$$= 0.01716803 \text{ or } 17.17ms$$ (one way latency)

2) $$RTT = 2 \times 0.01717 = 34.34ms$$ (round trip time)

4.4 Euclidean Distance

The Euclidean distance formula can be obtained from many sources; this research uses an un-weighted Euclidean distance formula [Jai91]. The formula has one unknown location latency measurement and subtracts a known location, squaring the difference and sums the difference of a second unknown location latency measurement and a second known location, then squares the sum of the differences. The square root of this product results in the Euclidean distance, which is unitless. The formula below shows how to calculate the distances from each destination city to Chicago in Table 4.1. The distance examples of the t(min) data are listed in Table 4.1.

$$d = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \qquad (4.14)$$

1) $$x_{ik} - x_{jk} = 17.699 - 15.881$$ (Atlanta RTT – Chicago RTT

$$= 1.818$$ from Boston/Cambridge column)

2) $$x_{ik} - x_{jk} = 39.935 - 34.336$$ (Atlanta RTT – Chicago RTT

$$= 5.599$$ from San Francisco column)

3) $$\left(x_{ik} - x_{jk}\right)^2 = 1.818^2 = 3.385$$ (square the differences)

4) $$\left(x_{ik} - x_{jk}\right)^2 = 5.599^2 = 31.349$$

Table 4.1 Analytical Euclidean Distance of t(min) times

Destination City	Polling station		Euclidean Distance
	Boston/Cambridge	San Francisco	
	RTT in ms	RTT in ms	
Atlanta	**17.699**	**39.935**	**5.89**
Austin	32.566	28.978	17.52
Cambridge	0.000	49.895	22.23
Chicago	**15.881**	**34.336**	**0.00**
Dallas	29.429	28.141	14.90
Denver	31.762	20.418	21.12
Detroit	11.537	38.632	6.11
Houston	29.750	31.102	14.24
Los Angeles	48.077	6.162	42.78
New York City	3.475	46.790	17.58
Orlando	20.933	46.564	13.23
Philadelphia	4.956	46.323	16.22
Phoenix	43.491	12.100	35.45
San Diego	49.107	8.093	42.34
San Francisco	49.895	0.000	48.33
Seattle	49.235	13.001	39.59
St. Louis	19.211	33.145	3.54
Washington DC	7.064	45.390	14.14

5) $$\sum_{k=1}^{n}\left(x_{ik}-x_{jk}\right)^2 = 34.654$$ (sum the squared difference)

6) $$d = \sqrt{\sum_{k=1}^{n}\left(x_{ik}-x_{jk}\right)^2} = \sqrt{34.654} = 5.887$$

The t(min) Euclidean distances are calculated using the delay formula and locating a known city to establish the truth table for comparison against the simulation results. The theoretical zero byte packet RTT is used to create a Euclidean Distance table for the simulation results. The t(min) Euclidean distances from Table 4.1 are used in Table 4.2 then compared against the AT&T ATM network simulation results for the destination city of Chicago. In this example the city of Chicago is within a Euclidean

Table 4.2 AT&T Network Euclidean Distance Table

Destination City	Polling station Boston/Cambridge RTT in ms	San Francisco RTT in ms	Euclidean Distance
Atlanta	17.9	51.7	17.6
Austin	87.8	37.9	71.9
Cambridge FTP Server	0.2	50.4	22.6
Buffalo	7.0	43.2	12.7
Chicago	**15.6**	**34.4**	**0.4**
Dallas	84.6	34.7	68.6
Denver	70.5	20.6	56.2
Detroit	20.2	39.0	6.4
Houston	31.0	38.6	15.6
Kansas City	24.6	43.3	12.5
Las Vegas	60.7	10.7	50.5
Los Angeles	56.3	6.3	49.0
Miami	24.9	58.2	25.6
New Orleans	36.7	44.3	23.0
New York	3.1	53.8	23.5
Orlando	20.9	54.2	20.6
Philadelphia	4.7	49.9	19.3
Phoenix	64.2	14.0	52.3
Pittsburgh	11.1	43.8	10.8
Raleigh	11.3	51.8	18.2
Salt Lake City	61.9	12.0	51.0
San Diego	58.2	13.2	47.1
San Francisco	49.9	0.4	47.9
St Louis	20.4	39.2	6.7
Tampa	22.4	55.7	22.4
Washington DC	7.0	47.7	16.2
Chicago Router	**16.0**	**34.2**	

distance 0.4 of the t(min) result, thus the Chicago Router is located in Chicago. The results of this are shown in Table 4.2 for the AT&T network model.

The t(min) Euclidean distances for the MCI Fast Ethernet network are calculated using the distance formula and will be slightly different from Table 4.1 because of different routing in the network topologies. The t(min) Euclidean distances are then

Table 4.3 MCI Network Euclidean Distance Table

Destination City	Polling station		Euclidean Distance
	Boston/Cambridge RTT in ms	San Francisco RTT in ms	
Atlanta	18.2	45.6	9.1
Austin	32.1	37.1	15.0
Boston FTP Server	0.1	50.5	22.0
Buffalo	9.9	53.4	18.3
Chicago	**17.0**	**36.2**	**0.4**
Dallas	29.0	36.9	11.9
Aurora	32.5	47.0	18.6
Detroit	21.5	40.3	5.8
Houston	30.6	32.5	14.1
Kansas City	22.9	37.4	5.8
Las Vegas	57.1	17.2	44.4
Los Angeles	51.2	6.4	45.5
Miami	28.8	58.0	24.4
New Orleans	36.3	38.2	19.3
New York	3.5	47.7	17.6
Orlando	24.6	53.2	18.3
Philadelphia	9.5	47.8	13.6
Phoenix	53.0	12.5	43.2
Pittsburgh	11.3	49.5	14.2
Raleigh	11.6	49.8	14.3
Salt Lake City	40.3	54.7	29.4
San Diego	49.4	8.5	42.8
San Francisco	50.3	0.3	49.2
St Louis	18.9	33.3	3.7
Tampa	26.1	55.5	20.9
Washington DC	7.2	46.3	13.9
Chicago Router	**17.1**	**36.5**	

compared against the MCI network simulation results for the destination city of Chicago again. In this example the city of Chicago is within a Euclidean distance 0.4 of the t(min) result, thus the Chicago Router is located in Chicago. The results of this are shown in Table 4.3 for the MCI network model.

4.5 Time-to-location Algorithm for Multiple AS network

The next issue is to establish a multiple AS network time-to-location and identify unique problems associated with multiple commercial vendors passing packets to each other. To calculate a time-to-location for a multiple AS network node the same four issues of line speed, queue size, switching speed and physical separation come into play. Simulation results show the differences between T1(1.54Mbps), OC-3(51Mbps) and OC-12(622Mbps) links passing traffic between the two modeled networks. Physical separation is calculated the same way as in paragraph 4.3.3 above.

4.5.1 Linear Slope The linear slope of a multiple AS network packet behaves in much the same way as in a single AS network by comparing a T1 link with 32 and 16 bytes packets passing between the networks. San Francisco is the baseline polling station for origination of ping packets to provide a consistent starting location. Twenty-eight destinations are sent ping packets to include 26 cities on the opposing network and 2 local destinations, the outgoing router and the internal WAN FTP server. An example for the AT&T polling station passing packets to the MCI network model is shown in Figure 4.12 and for the MCI polling station to pass packets to the AT&T network model is shown in Figure 4.13.

The figures show that 32 and 16 bytes packets behave in much the same fashion on a multiple AS network as they do on a single AS network. Assuming the linear slopes are approximately equal, the linear slope for use in the y = mx + b equation is established using the mean linear slope of the AS network destinations for each source network. The average linear slope and the RTT minimums are used in the formula listed in paragraph

4.3.1 to obtain the theoretical zero byte packet RTTs from the minimum values for each multiple AS network destination city.

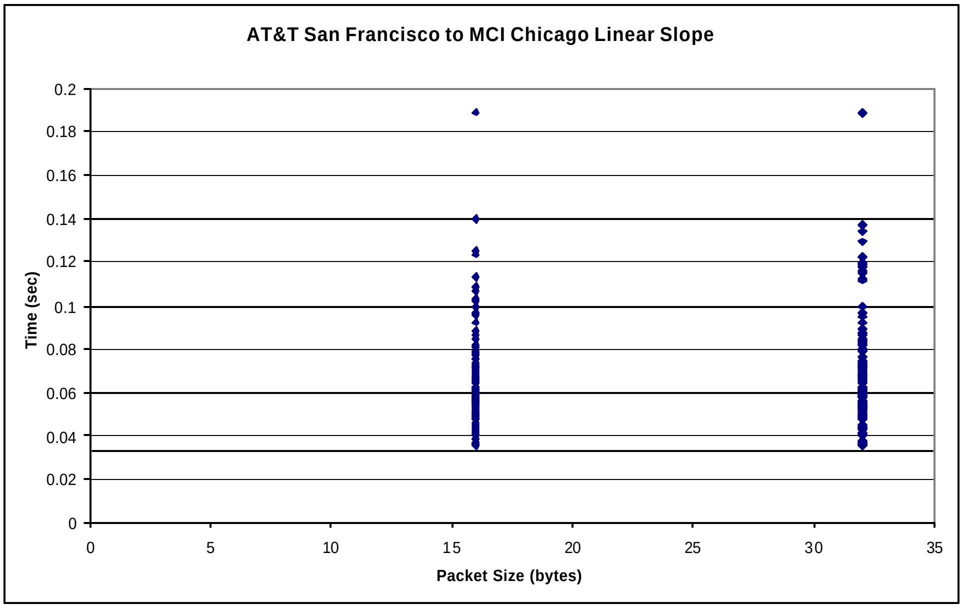

Figure 4.13 AT&T to MCI 32 and 16 bytes Combined High and Low Load

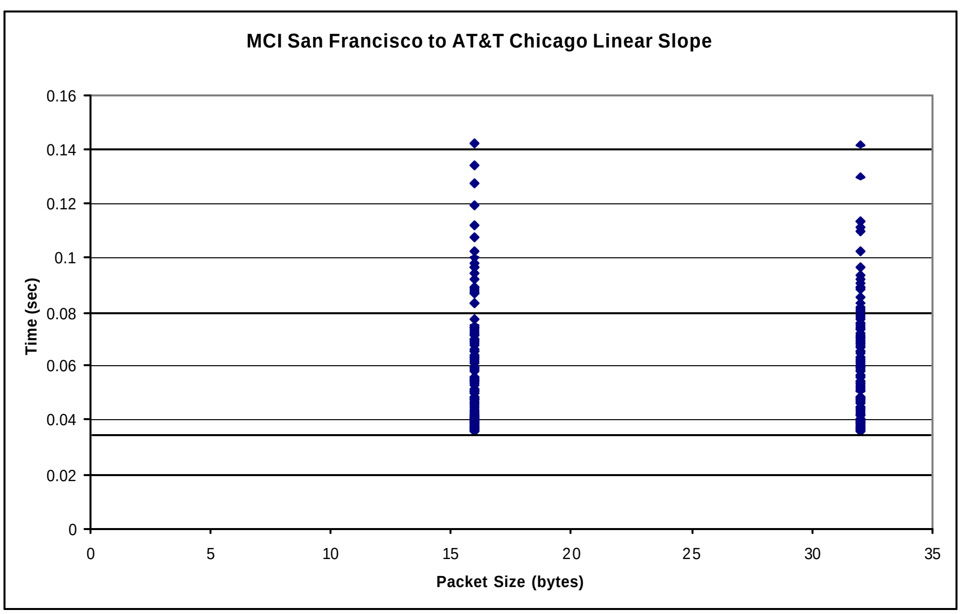

Figure 4.14 MCI to AT&T 32 and 16 bytes Combined High and Low Load

Table 4.4 MCI to AT&T Network "Zero" Bytes Packet RTT

"Zero byte" Y Intercept in milliseconds			
MCI to AT&T Network	OC12	OC3	T1
MCI_SF_FTP	1.92	1.93	1.92
MCI_SF_ROUTER	**1.77**	**1.77**	**1.77**
ATT Atlanta	49.77	47.36	49.86
ATT Austin	38.28	36.91	38.25
ATT Cambridge	52.14	52.15	53.32
ATT Buffalo	44.82	44.83	46.07
ATT Chicago	36.17	36.16	37.46
ATT Dallas	36.47	36.46	38.32
ATT Denver	22.29	22.28	24.00
ATT Detroit	40.80	40.78	42.40
ATT Houston	34.41	34.30	35.68
ATT Kansas City	39.69	39.15	40.44
ATT Las Vegas	12.45	12.43	14.73
ATT Los Angeles	8.06	8.05	10.89
ATT Miami	57.77	53.85	56.50
ATT New Orleans	40.11	39.96	41.34
ATT New York	48.79	48.72	50.12
ATT Orlando	49.95	49.86	51.10
ATT Philadelphia	49.74	49.54	50.93
ATT Phoenix	14.44	14.59	15.82
ATT Pittsburgh	45.47	45.46	47.75
ATT Raleigh	51.70	51.57	52.85
ATT Salt Lake City	13.76	13.75	16.04
ATT San Diego	10.08	10.06	13.40
ATT San Francisco	**1.92**	**1.92**	**5.32**
ATT St Louis	35.25	35.12	36.60
ATT Tampa	51.42	51.32	53.11
ATT Washington DC	47.39	47.30	48.70

The linear slope for the MCI simulation network is calculated using the formula in paragraph 4.3.1 to calculate the OC12, OC3 and T1 theoretical zero byte packet RTTs shown in Table 4.4 for the MCI to AT&T network. This table shows visually that link bandwidth is a factor in being able to successfully calculate a time-to-location algorithm. Moving within the city of San Francisco from one commercial vendor network to another, MCI to AT&T demonstrates the issue of link

Table 4.5 AT&T to MCI “Zero” Bytes Packet RTT

“Zero byte” Y Intercept in milliseconds			
AT&T to MCI Network	OC12	OC3	T1
ATT_SF_FTP	0.86	0.63	0.63
ATT_SF_ROUTER	0.86	0.63	0.63
MCI Atlanta	45.87	44.33	46.37
MCI Austin	38.22	38.00	39.24
MCI Boston	50.70	50.50	51.68
MCI Buffalo	43.37	43.16	44.40
MCI Chicago	34.73	34.51	35.77
MCI Dallas	35.03	34.81	36.19
MCI Aurora	20.85	20.63	21.87
MCI Detroit	39.35	39.13	40.38
MCI Houston	31.66	31.37	32.50
MCI Kansas City	30.49	30.27	31.67
MCI Las Vegas	11.03	10.80	12.19
MCI Los Angeles	6.63	6.41	7.84
MCI Miami	57.26	55.93	59.36
MCI New Orleans	40.55	37.78	39.14
MCI New York	47.51	47.15	48.48
MCI Orlando	53.18	53.04	55.26
MCI Philadelphia	49.20	48.40	50.17
MCI Phoenix	12.79	12.52	14.39
MCI Pittsburgh	44.04	47.07	45.30
MCI Raleigh	52.27	50.02	52.79
MCI Salt Lake City	12.35	12.15	13.53
MCI San Diego	8.66	8.46	9.77
MCI San Francisco	0.80	0.63	1.73
MCI St Louis	34.21	33.58	37.86
MCI Tampa	54.52	55.37	57.29
MCI Washington DC	46.53	45.90	47.15

bandwidth not being eliminated as a latency factor. The MCI San Francisco polling station to the AT&T Los Angeles destination city is one example of being outside of a 2ms zero byte mean. This may be caused by a large MTU network such as Fast Ethernet transferring packets to a small MTU network such as ATM, but this is not confirmed by this research.

The linear slope for the AT&T simulation network is calculated using the formula in paragraph 4.3.1 to calculate the OC12, OC3 and T1 theoretical zero byte packet RTTs shown in Table 4.5 for the AT&T to MCI network. This table shows visually that link bandwidth is a factor in being able to successfully calculate a time-to-location algorithm. The AT&T San Francisco polling station to the MCI Miami destination city is one example of being outside of a 2ms zero byte mean for city to city resolution.

4.5.2 Queue Size and Switching Speed In a single AS network, the High load is used to emulate business hours and the Low load is used to emulate non-business hours. The same criterion is used in the multiple AS network to standardize the modeling of the networks. The OPNET simulations designed for this research used a FIFO service discipline. Queuing delay for the AT&T ATM network is $2.75\times10^{-1}ms$ for the T1 link and $6.82\times10^{-4}ms$ for the OC12 link connecting AT&T San Francisco to MCI San Francisco. Both queuing delays are well within the required 2ms mean to eliminate queue delay and switch speed as a factor in a time-to-location algorithm as specified by NSA for city to city level resolution.

The result of the loads on the minimum RTT is shown in Figure 4.15 for the AT&T to MCI network and in Figure 4.16 for the MCI to AT&T network. The scatter plot lines are the link bandwidth lines from top to bottom, T1, OC3, and OC12. The minimum results visually use a power trend line to ease the visual interpretation of the data. The number of repetitions to determine a 95% confidence interval to be within 2ms of a theoretical zero byte size packet is used. The link sample sizes when calculated with the minimum RTTs for each destination city are listed in Table 4.6 for the network simulations.

Table 4.6 Network Sample Size Calculations

Network Link Bandwidth	Mean of the Minimum (3 repetitions)	Std Dev	Sample Size
AT&T to MCI OC12	29.8ms	18.1ms	317
AT&T to MCI OC3	29.8ms	18.2ms	318
AT&T to MCI T1	31.1ms	18.4ms	326
MCI to AT&T OC12	29.6ms	17.8ms	305
MCI to AT&T OC3	29.4ms	17.6ms	300
MCI to AT&T T1	31.1ms	17.6ms	296

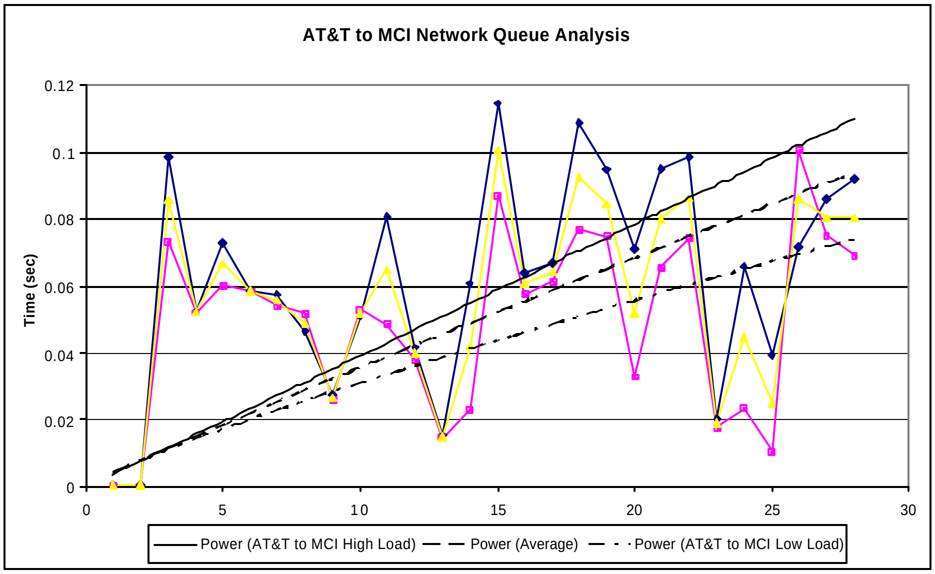

Figure 4.15 AT&T to MCI Network Load RTT

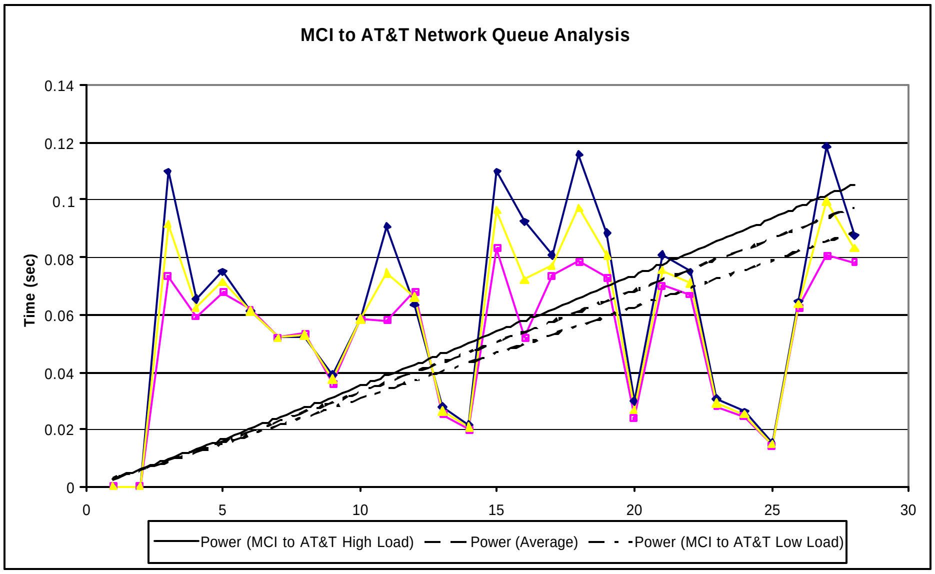

Figure 4.16 MCI to AT&T Network Load RTT

A partial factorial design is used to collect the multiple AS network simulation data results. The two network simulations were executed for 200 simulation seconds on each network, allowing for the routing tables being setup in the first 100 simulation seconds and 33 ping RTTs being calculated in the last 100 simulation seconds. All ping traffic examined is the product of 1 polling station, 2 networks per polling station, 2 background loads per network, 3 link bandwidths per network, 1 packet size per link bandwidth, 3 seeds per packet, 33 ping latency measurements per seed for a total of 1,188 samples per destination city. There are a total of 28 destinations for a total 33,264 collected pings.

4.6 Analysis of Link Bandwidth Behavior

The multiple AS network model setup is a combination of the AT&T ATM network model and the MCI Fast Ethernet network model. They were joined at 26 destination cities from WAN router to WAN router by the 3 links, OC12, OC3 and T1. An overview of the network is shown in Figure 4.17.

The theoretical zero byte packet RTTs are analyzed for variance using the statistical discovery software tool, JMP, Release 5.0.1.2. The networks are found to not behave in consistent ways, the topologies and the way they handle the packets are unique to the simulation network routing and link bandwidths. At first it appears the results in Figure 4.17 visually prove the Chicago destination link bandwidth does meet the criteria of being within 2ms for a zero byte packet minimum. In the AT&T to MCI network analysis Chicago has zero byte response times of 34.727ms for an OC12 link, 34.51ms for an OC3 link and 35.769ms for a T1 link. When transferring packets from the MCI to

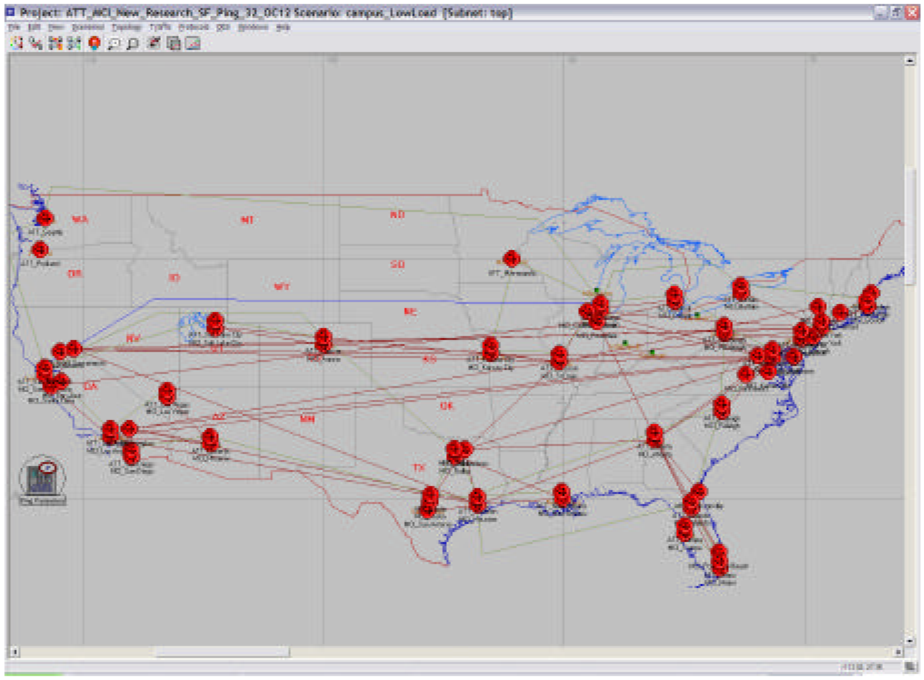

Figure 4.17 OPNET Multiple AS network Model

AT&T network Chicago has zero byte response times of 36.166ms for an OC12 link, 36.156ms for an OC3 link and 37.46ms for a T1 link. So it appears that the required city to city resolution is maintained.

In Figure 4.18 an example is shown that demonstrates how the link bandwidth cannot be eliminated, showing the MCI network simulation to have a 3ms deviation on "zero" bytes packets in the OC12, OC3 and T1 response times. The Los Angeles destination link bandwidth does not meet the criteria of being within 2ms for a zero byte packet minimum. In the AT&T to MCI network analysis Los Angeles has zero byte response times of 6.63ms for an OC12 link, 6.414ms for an OC3 link and 7.836ms for a T1 link, which meet the required city to city resolution. When transferring packets from the MCI to AT&T network Los Angeles has zero byte response times of 8.059ms for an OC12 link, 8.048ms for an OC3 link and 10.885ms for a T1 link, which does not meet the required city to city resolution.

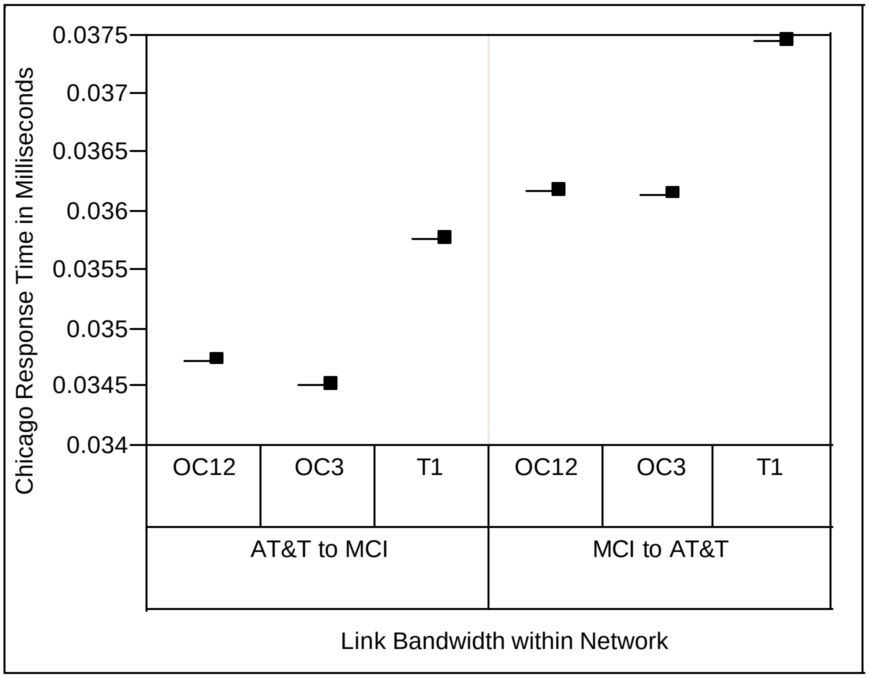

Figure 4.18 San Francisco to Chicago by Network Link Bandwidth

This is further shown in the analysis of variance in Tables 4.7 and 4.8, showing the network routing and the link bandwidth have a percentage of the variance in the latency between multiple AS networks. The percentage of network routing variance demonstrates the distance variance in the network routes for each packet, this shows the physical distance a packet travels effects the RTT. The physical distance is accounted for using the Euclidean distance tables and this also demonstrates how a time-to-distance algorithm will not work for geographic location. The link bandwidth is the size of the pipe between the networks to pass packets from one network to the other. The conclusion of this research is that in any calculations of a time-to-location algorithm on a multiple AS network are required to initiate a process for estimating link bandwidth calculations and account for multiple AS network routing to the destination city.

Table 4.7 Chicago Analysis of Variance on "Zero" bytes packets

Source	DF	SS	Mean Square	F Ratio	Prob > F
Network Routing	1	4×10^{-6}	3.8×10^{-6}	420.653	2.4×10^{-3}
Link Bandwidth	2	2×10^{-6}	1.01×10^{-6}	111.427	8.9×10^{-3}
Network*Link Bandwidth	2	1.81×10^{-8}	9.04×10^{-9}	.	.
Total	5	6×10^{-6}	1.17×10^{-6}		

Table 4.8 Chicago Variance Components

Component	Var Component	% of Total	Plot%	Sqrt(Var Comp)
Network Routing	1.26×10^{-6}	71.3		1.12×10^{-3}
Link Bandwidth	4.99×10^{-7}	28.2		7.1×10^{-4}
Network*Link Bandwidth	9.04×10^{-9}	0.5		1.0×10^{-4}
Total	1.77×10^{-6}	100.0		1.33×10^{-3}

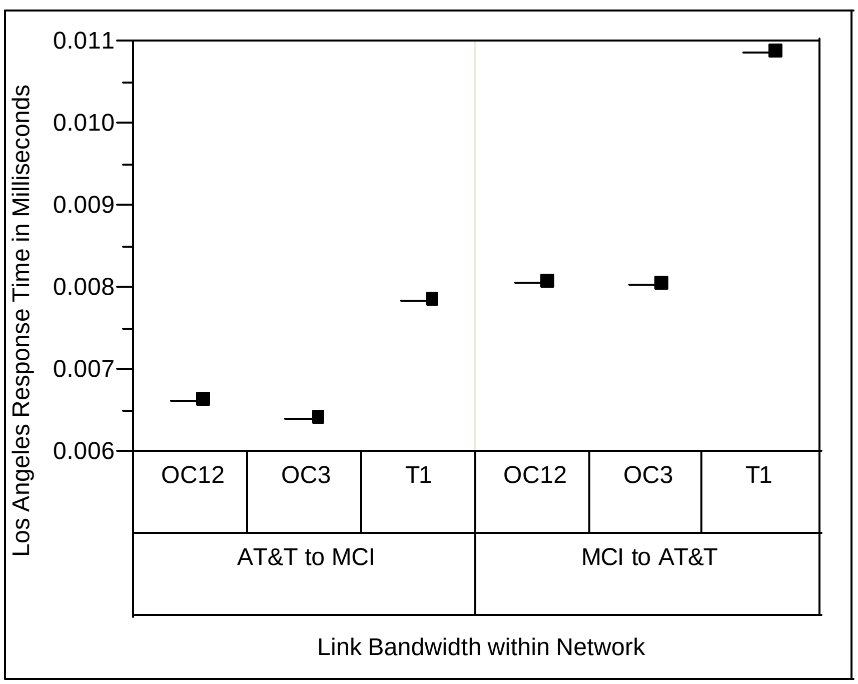

Figure 4.19 San Francisco to Los Angeles by Network Link Bandwidth

4.7 Summary

This chapter presented the implementation of the network model simulations and analysis of the results and various methods used in this research. NSA research is duplicated in a controlled laboratory environment solving for line speed, queue size, switching speed, and physical separation. A time-to-location algorithm is verified in the controlled laboratory environment and a model Euclidean distance table is created for each AS network. The mean linear slopes are used to calculate the multiple AS network "zero" bytes packet intercepts.

The analysis of the network routing and link bandwidth is shown to become a factor in a multiple AS network time-to-location algorithm. The T1 link demonstrates that the type of link bandwidth is a factor that must be calculated to start future research into a successful time-to-location algorithm for a multiple AS network geolocation resolution.

V. Conclusion and Future Work

5.1 Overview

"The Air Force believes that dominating the information spectrum is as critical to conflict now as controlling air and space or occupying land was in the past and is seen as an indispensable and synergistic component of aerospace power" [AFD98]. These systems therefore, must be protected to the level required of any weapons asset. To prevent an enemy from exploiting these assets, the Air Force and DOD require the capability to geographically locate a node on the Internet via its logical address consistently and reliably. A consistent multiple AS network time-to-location algorithm is the first step towards the goal of completely securing our information systems.

5.2 A Time-to-location Algorithm

The goal of this research was to determine the geographic location of a node using only packet latency measurements on an AS network and was a success in a controlled laboratory environment. Duplicating NSA research the line speed, queuing delay, switch speed and physical distance measurements are used as input to a time-to-location algorithm. The time-to-location algorithm was then used to establish a Euclidean distance table measurement of known locations in an autonomous system to provide known locations or markers to determine the location of unknown computer nodes at the city to city level resolution.

The time-to-location algorithm was successful 71.4% of the time as demonstrated in Tables 4.4 and 4.5 in locating a computer node in a multiple AS network. The mean of the linear slope measurement was used with the packet size to calculate the zero byte packet intercept or RTT value. Using these measurements as a baseline for conducting research into the multiple AS network environment this research identified the link bandwidth as an additional factor to introduce into the calculation of a time-to-location algorithm to resolve a multiple AS network resolution. This is only a small first step into resolving the multiple AS network time-to-location algorithm and more latency issues need to be identified as factors for this algorithm to work successfully.

5.3 Future Work

Some areas of future research are:

1) Developing a reliable mathematical calculation to estimate bandwidth sizes on the real world Internet to input the link bandwidth as a factor in the time-to-location algorithm.
2) Develop software that calculates the location of a computer node in real time using a time-to-location algorithm to identify and isolate the metropolitan area that a hacker is attacking the network from.
3) Identify more multiple AS network latency issues, ensuring that link bandwidth, queuing protocols or automatic fault recovery routing techniques are not affecting real world multiple AS networks.

Network vendors have an option to use priority queuing of their own packet traffic pushing traffic destined to a competitor's network to a Lower priority than their own internal network traffic. Another option for queuing by a network vendor is "Hot Potato" routing or passing a packet destined for another network to the competitor's network by constantly transferring the packet until it reaches its destination, potentially adding latency to the packet RTT [Web04]. Both of these issues could add inconsistent latency measurements into a time-to-location algorithm and thus cause a result to be unsuccessful in locating a hacker's location.

Appendix A: Collected Data

The first six tables listed Table A.1 – A.6 will be provided on a Compact Disc.

Table A.1 AT&T Network Model Raw Data

Request Thesis Raw Data CD for results of 53, 32, and 16 bytes packet results.

Table A.2 MCI Network Model Raw Data

Request Thesis Raw Data CD for results of 53, 32, and 16 bytes packet results.

Table A.3 AT&T to MCI Network Model Raw Data

Request Thesis Raw Data CD for results of 32 and 16 bytes packet results.

Table A.4 MCI to AT&T Network Model Raw Data

Request Thesis Raw Data CD for results of 32 and 16 bytes packet results.

Table A.5 AT&T to MCI Network Model Raw Data

Request Thesis Raw Data CD for results of OC12, OC3, and T1 Link Bandwidth Raw Data results.

Table A.6 MCI to AT&T Network Model Raw Data

Request Thesis Raw Data CD for results of OC12, OC3, and T1 Link Bandwidth Raw Data results.

Table A.7 Cambridge Polling station AT&T Network 53, 32, and 16 Bytes Packet Results

Simulation Results	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	53	53	53	53	53
Cambridge_ROUTER	0.000677	0.00064	9.85E-05	0.000317	0.00088
Atlanta	0.018498	0.018547	0.000117	0.018398	0.018998
Austin	0.088386	0.088429	8.54E-05	0.088362	0.088822
Cambridge_FTP	0.000697	0.000712	2.77E-05	0.000577	0.000934
Buffalo	0.007907	0.007889	0.000126	0.007587	0.008127
Chicago	0.016112	0.016137	5.92E-05	0.016112	0.016392
Dallas	0.085211	0.085254	9.28E-05	0.085151	0.085711
Denver	0.071231	0.071239	0.000131	0.071011	0.071606
Detroit	0.020926	0.020936	0.000122	0.020766	0.021366
Houston	0.060449	0.05507	0.017596	0.031484	0.089534
Kansas_City	0.027804	0.027565	0.001683	0.025087	0.030482
Las_Vegas	0.061517	0.061497	0.000133	0.061177	0.061867
Los_Angeles	0.057035	0.057027	0.000133	0.056775	0.057315
Miami	0.025738	0.025725	0.000132	0.025438	0.026026
New_Orleans	0.066385	0.062355	0.017487	0.03724	0.09531
New_York	0.003621	0.003661	7.9E-05	0.003601	0.004041
Orlando	0.021535	0.021547	0.000117	0.021375	0.021935
Philadelphia	0.005463	0.005466	0.000129	0.005263	0.005803
Phoenix	0.064736	0.064735	0.000132	0.064496	0.065056
Pittsburgh	0.018778	0.017974	0.004978	0.011547	0.025994
Raleigh	0.01218	0.012146	0.000129	0.01184	0.012452
Salt_Lake_City	0.062735	0.062726	0.000131	0.062455	0.063087
San_Diego	0.059003	0.058999	0.00013	0.058743	0.059283
San_Francisco	0.050452	0.050453	5.52E-06	0.050452	0.050576
St_Louis	0.021003	0.021057	0.000113	0.020923	0.021483
Tampa	0.023164	0.023149	0.000131	0.022843	0.023436
Washington_DC	0.007688	0.007697	0.000125	0.007488	0.008048
Packet Size	32	32	32	32	32
Cambridge_ROUTER	0.00069	0.000663	6.49E-05	0.00043	0.000821
Atlanta	0.018491	0.018541	0.000118	0.018391	0.019032
Austin	0.088375	0.088425	8.91E-05	0.088354	0.088815
Cambridge_FTP	0.00069	0.000705	2.67E-05	0.00067	0.000968
Buffalo	0.0079	0.007881	0.000127	0.00756	0.008214
Chicago	0.016105	0.016127	5.92E-05	0.016105	0.016405
Dallas	0.085204	0.085247	9.61E-05	0.085144	0.085664
Denver	0.071224	0.071235	0.000131	0.071004	0.071596
Detroit	0.020919	0.020931	0.000123	0.020739	0.02137
Houston	0.060462	0.055472	0.017773	0.031477	0.089587
Kansas_City	0.027857	0.027765	0.001682	0.02508	0.030495
Las_Vegas	0.06151	0.061488	0.000134	0.06119	0.061803
Los_Angeles	0.057009	0.057016	0.000131	0.056768	0.057309
Miami	0.025733	0.025718	0.00013	0.025431	0.026009

New_Orleans	0.06877	0.062144	0.019147	0.037214	0.095323
New_York	0.003614	0.00365	7.87E-05	0.003594	0.004094
Orlando	0.021528	0.021538	0.000118	0.021368	0.021928
Philadelphia	0.005456	0.005458	0.000127	0.005256	0.005836
Phoenix	0.064729	0.064729	0.000132	0.06449	0.065076
Pittsburgh	0.018751	0.017768	0.004938	0.01156	0.025982
Raleigh	0.012174	0.01214	0.000127	0.011834	0.012414
Salt_Lake_City	0.062728	0.06272	0.00013	0.062448	0.063008
San_Diego	0.058996	0.05899	0.000133	0.058736	0.059409
San_Francisco	0.050445	0.050446	5.85E-06	0.050445	0.050569
St_Louis	0.020979	0.021047	0.000111	0.020896	0.021428
Tampa	0.023143	0.023132	0.00013	0.022857	0.023417
Washington_DC	0.007681	0.007683	0.000122	0.007501	0.008041
Packet Size	16	16	16	16	16
Cambridge_ROUTER	0.000686	0.000677	4.07E-05	0.000466	0.000972
Atlanta	0.018507	0.018538	0.000115	0.018367	0.018907
Austin	0.088371	0.088415	8.74E-05	0.08835	0.088791
Cambridge_FTP	0.000686	0.000701	2.67E-05	0.000666	0.000979
Buffalo	0.007895	0.007871	0.000126	0.007555	0.008098
Chicago	0.016101	0.016122	5.72E-05	0.0161	0.016386
Dallas	0.085194	0.085241	9.52E-05	0.08514	0.08562
Denver	0.07122	0.071226	0.000129	0.07098	0.07154
Detroit	0.020895	0.020921	0.000124	0.020715	0.021335
Houston	0.060491	0.056393	0.017871	0.031473	0.089583
Kansas_City	0.027833	0.027742	0.001677	0.025076	0.030451
Las_Vegas	0.061506	0.061485	0.000135	0.061165	0.061775
Los_Angeles	0.057024	0.057013	0.000131	0.056764	0.057284
Miami	0.025727	0.02571	0.000131	0.025407	0.025947
New_Orleans	0.066284	0.058703	0.018236	0.037229	0.095318
New_York	0.00361	0.003645	7.98E-05	0.00359	0.00407
Orlando	0.021524	0.021536	0.000119	0.021364	0.021903
Philadelphia	0.005451	0.005453	0.000125	0.005252	0.005792
Phoenix	0.064725	0.064729	0.00013	0.064485	0.065047
Pittsburgh	0.018767	0.017988	0.004951	0.011556	0.025978
Raleigh	0.012149	0.012129	0.000127	0.011829	0.012369
Salt_Lake_City	0.062724	0.062713	0.00013	0.062444	0.063004
San_Diego	0.058992	0.058984	0.000131	0.058732	0.059292
San_Francisco	0.050441	0.050442	5.68E-06	0.050441	0.050559
St_Louis	0.020972	0.021036	0.000107	0.020892	0.021466
Tampa	0.023132	0.023126	0.000131	0.022852	0.023451
Washington_DC	0.007677	0.007677	0.000125	0.007477	0.00813

Table A.8 Chicago Polling station AT&T Network 53, 32, and 16 Bytes Packet Results

Simulation Results	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	53	53	53	53	53
Chicago_ROUTER	0.000177	0.00023	0.000138	0.000177	0.000796
Atlanta	0.024298	0.024304	0.00012	0.024118	0.024692
Austin	0.072407	0.072469	0.000106	0.072347	0.072927
Cambridge	0.016415	0.016425	0.000131	0.016202	0.016768
Buffalo	0.009282	0.009247	0.00013	0.008922	0.009502
Chicago_FTP	0.000537	0.000563	8.67E-05	0.000323	0.000837
Dallas	0.069256	0.069298	0.000109	0.069156	0.069696
Denver	0.055256	0.055269	0.000134	0.055036	0.055753
Detroit	0.004971	0.004981	0.000126	0.004771	0.005371
Houston	0.07316	0.07322	0.000112	0.073059	0.073619
Kansas_City	0.011949	0.012079	0.001961	0.009112	0.014527
Las_Vegas	0.045502	0.045493	0.000135	0.045202	0.045815
Los_Angeles	0.04106	0.041055	0.000134	0.04078	0.041381
Miami	0.031497	0.03147	0.000135	0.031177	0.031733
New_Orleans	0.079135	0.079113	0.000136	0.078835	0.079461
New_York	0.019636	0.01969	9.59E-05	0.019596	0.020076
Orlando	0.027294	0.027303	0.000126	0.027103	0.027674
Philadelphia	0.018678	0.018625	0.002004	0.015678	0.021837
Phoenix	0.048761	0.048763	0.000133	0.048521	0.049062
Pittsburgh	0.009834	0.009799	0.000133	0.009494	0.010041
Raleigh	0.01794	0.0179	0.000131	0.01758	0.01814
Salt_Lake_City	0.04676	0.046757	0.000134	0.04648	0.04702
San_Diego	0.043026	0.043022	0.000135	0.042769	0.043316
San_Francisco	0.034457	0.034458	4.74E-06	0.034457	0.034503
St_Louis	0.005068	0.005097	0.000118	0.004909	0.005469
Tampa	0.028923	0.028895	0.000135	0.028583	0.029163
Washington_DC	0.013633	0.013637	0.000129	0.013413	0.013993
Packet Size	32	32	32	32	32
Chicago_ROUTER	0.00019	0.000242	0.000139	0.00019	0.00083
Atlanta	0.024291	0.024296	0.000123	0.024091	0.024691
Austin	0.0724	0.072465	0.000105	0.07234	0.07286
Cambridge	0.016405	0.016418	0.00013	0.016165	0.016766
Buffalo	0.009255	0.009234	0.000129	0.008915	0.009529
Chicago_FTP	0.00061	0.000622	5.69E-05	0.000316	0.00083
Dallas	0.06923	0.069283	0.00011	0.06915	0.06969
Denver	0.05525	0.055257	0.000132	0.055009	0.055609
Detroit	0.004965	0.004974	0.000125	0.004764	0.005364
Houston	0.073152	0.073212	0.000118	0.073052	0.073632
Kansas_City	0.011923	0.012026	0.001952	0.009105	0.01452
Las_Vegas	0.045475	0.045472	0.000135	0.045195	0.045775
Los_Angeles	0.041054	0.041044	0.000134	0.040774	0.041367
Miami	0.031471	0.031459	0.000132	0.03115	0.03175
New_Orleans	0.079108	0.079101	0.000133	0.078808	0.079401

New_York	0.019629	0.019682	9.65E-05	0.019589	0.020069
Orlando	0.027287	0.027292	0.000126	0.027087	0.027668
Philadelphia	0.018631	0.018384	0.001979	0.015672	0.021904
Phoenix	0.048755	0.048754	0.000133	0.048495	0.049095
Pittsburgh	0.009807	0.00979	0.000131	0.009487	0.010081
Raleigh	0.017893	0.017883	0.000131	0.017553	0.018167
Salt_Lake_City	0.046752	0.046743	0.000135	0.046454	0.047074
San_Diego	0.043022	0.043018	0.000134	0.042782	0.043342
San_Francisco	0.03445	0.034451	5.21E-06	0.03445	0.034549
St_Louis	0.005025	0.005077	0.000117	0.004922	0.005482
Tampa	0.028896	0.028881	0.000135	0.028576	0.029188
Washington_DC	0.013607	0.013627	0.000128	0.013406	0.014016
Packet Size	16	16	16	16	16
Chicago_ROUTER	0.000206	0.00026	0.0001392	0.000206	0.000786
Atlanta	0.024286	0.024285	0.0001239	0.024106	0.024667
Austin	0.072396	0.072458	0.0001081	0.072336	0.072916
Cambridge	0.016401	0.016409	0.0001307	0.01618	0.016741
Buffalo	0.009251	0.009218	0.0001267	0.008911	0.009491
Chicago_FTP	0.000691	0.000693	4.293E-05	0.000311	0.000806
Dallas	0.069225	0.06928	0.0001105	0.069145	0.069665
Denver	0.055245	0.055252	0.0001302	0.055005	0.055565
Detroit	0.00496	0.004961	0.0001267	0.00476	0.00532
Houston	0.073131	0.073201	0.0001154	0.073048	0.073588
Kansas_City	0.011859	0.011783	0.0019635	0.009081	0.014476
Las_Vegas	0.045471	0.045464	0.000131	0.045191	0.045721
Los_Angeles	0.041029	0.041033	0.0001283	0.040769	0.041349
Miami	0.031466	0.031449	0.0001276	0.031166	0.031705
New_Orleans	0.079104	0.079089	0.0001297	0.078784	0.079352
New_York	0.019625	0.019673	9.694E-05	0.019585	0.020065
Orlando	0.027283	0.027285	0.0001289	0.027083	0.027663
Philadelphia	0.018627	0.018373	0.0019573	0.015667	0.021806
Phoenix	0.04875	0.04875	0.0001315	0.04849	0.04905
Pittsburgh	0.009803	0.009781	0.0001278	0.009463	0.010023
Raleigh	0.017889	0.017868	0.0001272	0.017569	0.018111
Salt_Lake_City	0.046749	0.046734	0.0001331	0.046449	0.047009
San_Diego	0.042997	0.043007	0.0001312	0.042757	0.043317
San_Francisco	0.034446	0.034447	4.854E-06	0.034446	0.034492
St_Louis	0.005038	0.005075	0.0001208	0.004917	0.005438
Tampa	0.028892	0.028872	0.0001291	0.028552	0.029174
Washington_DC	0.013622	0.013627	0.0001306	0.013402	0.013955

Table A.9 San Francisco Polling station AT&T Network 53, 32, and 16 Bytes Packet Results

Simulation Results	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	53	53	53	53	53
San_Francisco_ROUTER	0.000137	0.000356	0.001331	0.000137	0.008868
Atlanta	0.052005	0.052021	0.000118	0.051844	0.052425
Austin	0.038067	0.038125	9.97E-05	0.037987	0.038567
Cambridge	0.050772	0.050787	0.000132	0.050572	0.051132
Buffalo	0.043642	0.043611	0.00013	0.043282	0.043822
Chicago	0.034497	0.034538	8.19E-05	0.034477	0.034877
Dallas	0.034896	0.034947	0.000103	0.034796	0.035356
Denver	0.020896	0.02091	0.000135	0.020676	0.021236
Detroit	0.039332	0.039348	0.000126	0.039151	0.039728
Houston	0.038819	0.038867	0.000108	0.038699	0.039279
Kansas_City	0.04629	0.046411	0.001689	0.043472	0.048885
Las_Vegas	0.011142	0.011129	0.000138	0.010842	0.011402
Los_Angeles	0.0067	0.006697	0.00014	0.00642	0.00698
Miami	0.058693	0.058657	0.000133	0.058353	0.058935
New_Orleans	0.044775	0.044756	0.000134	0.044455	0.045104
New_York	0.054016	0.054062	9.07E-05	0.053956	0.054476
Orlando	0.05447	0.054485	0.000122	0.05429	0.054885
Philadelphia	0.052998	0.052707	0.001936	0.050039	0.056198
Phoenix	0.014401	0.01441	0.000138	0.014161	0.014741
Pittsburgh	0.044194	0.044157	0.000135	0.043834	0.044426
Raleigh	0.05228	0.052256	0.000132	0.05194	0.0525
Salt_Lake_City	0.01242	0.012399	0.000134	0.01212	0.01268
San_Diego	0.008668	0.008467	0.001287	0.000457	0.008968
San_Francisco_FTP	0.000477	0.000486	9.28E-05	0.000137	0.000799
St_Louis	0.039409	0.039458	0.000116	0.039309	0.039873
Tampa	0.056099	0.056075	0.000134	0.055759	0.056339
Washington_DC	0.047993	0.048002	0.00013	0.047793	0.048413
Packet Size	32	32	32	32	32
San_Francisco_ROUTER	0.00013	0.000136	1.19E-05	0.00013	0.000253
Atlanta	0.051998	0.052014	0.000121	0.051838	0.052378
Austin	0.03806	0.038114	9.67E-05	0.03798	0.0385
Cambridge	0.050765	0.050777	0.000133	0.050565	0.051112
Buffalo	0.043636	0.043599	0.000131	0.043276	0.043836
Chicago	0.03449	0.034528	8.2E-05	0.03447	0.03491
Dallas	0.03489	0.03494	0.000101	0.034789	0.035369
Denver	0.020889	0.020905	0.000136	0.020669	0.021249
Detroit	0.039325	0.039335	0.000123	0.039145	0.039742
Houston	0.038812	0.038865	0.00011	0.038692	0.039292
Kansas_City	0.046263	0.046301	0.001702	0.043465	0.048863
Las_Vegas	0.011115	0.011118	0.000136	0.010835	0.011388
Los_Angeles	0.006694	0.006684	0.000136	0.006434	0.006974
Miami	0.058667	0.058645	0.000135	0.058347	0.058907

New_Orleans	0.044748	0.044746	0.000136	0.044468	0.045008
New_York	0.054009	0.054053	9.09E-05	0.053949	0.05449
Orlando	0.054464	0.054475	0.000121	0.054284	0.054884
Philadelphia	0.052971	0.052592	0.001954	0.050032	0.056192
Phoenix	0.014394	0.014394	0.000135	0.014154	0.014752
Pittsburgh	0.044187	0.044151	0.000134	0.043848	0.044428
Raleigh	0.052273	0.052248	0.000132	0.051933	0.052493
Salt_Lake_City	0.012393	0.012389	0.000136	0.012113	0.012673
San_Diego	0.008661	0.008659	0.000137	0.008402	0.008942
San_Francisco_FTP	0.00055	0.000568	5.39E-05	0.000336	0.000795
St_Louis	0.039402	0.039451	0.000115	0.039302	0.039862
Tampa	0.056072	0.056061	0.000136	0.055772	0.056395
Washington_DC	0.047987	0.047993	0.000129	0.047786	0.048408
Packet Size	16	16	16	16	16
San_Francisco_ROUTER	0.000206	0.000258	0.000144	0.000206	0.000826
Atlanta	0.052013	0.052024	0.000133	0.051813	0.052433
Austin	0.038056	0.038125	0.000116	0.037976	0.038576
Cambridge	0.050801	0.0508	0.000142	0.050541	0.051141
Buffalo	0.043651	0.043615	0.000138	0.043271	0.043851
Chicago	0.034506	0.034561	9.96E-05	0.034466	0.034966
Dallas	0.034885	0.034947	0.00012	0.034785	0.035385
Denver	0.020925	0.020921	0.000143	0.020665	0.021265
Detroit	0.03934	0.039353	0.000135	0.03912	0.03972
Houston	0.038828	0.038872	0.000126	0.038687	0.039328
Kansas_City	0.046319	0.046546	0.001653	0.043441	0.048908
Las_Vegas	0.01115	0.011137	0.000138	0.01081	0.01141
Los_Angeles	0.006709	0.006698	0.000143	0.006409	0.007029
Miami	0.058682	0.058659	0.00014	0.058302	0.058922
New_Orleans	0.044784	0.044766	0.000139	0.044404	0.045044
New_York	0.054025	0.054069	0.000103	0.053945	0.054505
Orlando	0.054479	0.054488	0.000136	0.054279	0.054859
Philadelphia	0.053007	0.052743	0.002098	0.050027	0.056212
Phoenix	0.01441	0.014411	0.000144	0.01413	0.01474
Pittsburgh	0.044203	0.044173	0.000138	0.043843	0.044423
Raleigh	0.052289	0.052263	0.000138	0.051929	0.052529
Salt_Lake_City	0.012429	0.012409	0.000143	0.012089	0.012779
San_Diego	0.008677	0.008675	0.000144	0.008397	0.008977
San_Francisco_FTP	0.000726	0.000718	4.41E-05	0.000351	0.000846
St_Louis	0.039457	0.03946	0.000133	0.039277	0.039878
Tampa	0.056108	0.05608	0.000139	0.055748	0.05635
Washington_DC	0.047982	0.048001	0.000137	0.047782	0.048362

Table A.10 Boston Polling station MCI Network 53, 32, and 16 Bytes Packet Results

Simulation Results	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	53	53	53	53	53
Boston_ROUTER	0.000241	0.000232	4.84E-05	0.000117	0.000324
Atlanta	0.017987	0.018322	0.001575	0.017658	0.035589
Austin	0.033567	0.034117	0.003337	0.03227	0.07855
Boston_FTP	0.000378	0.000369	4.65E-05	0.000261	0.000452
Buffalo	0.010497	0.01071	0.000571	0.010047	0.013314
Chicago	0.020391	0.019667	0.001218	0.017202	0.021534
Dallas	0.030767	0.031034	0.003086	0.029119	0.082151
Aurora	0.049897	0.0684	0.043923	0.032631	0.316527
Detroit	0.02212	0.02243	0.000736	0.021598	0.025761
Houston	0.032295	0.041591	0.016196	0.03075	0.160537
Kansas_City	0.023664	0.023953	0.000755	0.023077	0.027957
Las_Vegas	0.060466	0.060659	0.001388	0.058661	0.065936
Los_Angeles	0.050686	0.051349	0.004322	0.050188	0.100985
Miami	0.029533	0.029834	0.000787	0.028938	0.033405
New_Orleans	0.03908	0.051636	0.017609	0.036437	0.086813
New_York	0.003701	0.003856	0.00043	0.003584	0.007464
Orlando	0.025329	0.025619	0.000742	0.02479	0.029487
Philadelphia	0.010237	0.010556	0.000766	0.009686	0.014441
Phoenix	0.055674	0.055737	0.001464	0.053123	0.060556
Pittsburgh	0.012061	0.012369	0.000773	0.011459	0.016671
Raleigh	0.012353	0.01265	0.000766	0.011733	0.015343
Salt_Lake_City	0.043783	0.095359	0.053898	0.040559	0.150276
San_Diego	0.051497	0.051912	0.00136	0.049608	0.056074
San_Francisco	0.052926	0.052797	0.00459	0.050481	0.101321
St_Louis	0.019385	0.019724	0.001682	0.018996	0.038324
Tampa	0.026957	0.027348	0.00096	0.02622	0.031209
Washington_DC	0.007706	0.00796	0.000791	0.007383	0.015234
Packet Size	32	32	32	32	32
Boston_ROUTER	0.000225	0.000218	4.82E-05	0.0001	0.000309
Atlanta	0.017964	0.01831	0.001576	0.017642	0.035696
Austin	0.033739	0.034139	0.003425	0.03225	0.088376
Boston_FTP	0.000369	0.000361	4.46E-05	0.00025	0.000444
Buffalo	0.010457	0.010677	0.000571	0.010033	0.014009
Chicago	0.020454	0.019638	0.001224	0.01718	0.021356
Dallas	0.030551	0.030969	0.003351	0.029105	0.082481
Aurora	0.050434	0.072048	0.045562	0.03261	0.316611
Detroit	0.022121	0.02243	0.000769	0.021576	0.026772
Houston	0.032495	0.043816	0.01802	0.030738	0.160501
Kansas_City	0.023626	0.023935	0.000774	0.023056	0.027224
Las_Vegas	0.060611	0.060765	0.001396	0.05781	0.065397
Los_Angeles	0.050664	0.051335	0.004323	0.050168	0.100718
Miami	0.029478	0.029811	0.000791	0.028904	0.033282
New_Orleans	0.0384	0.046102	0.013642	0.036409	0.086636

New_York	0.003688	0.003847	0.000435	0.003573	0.007322
Orlando	0.025303	0.02561	0.000762	0.024769	0.029719
Philadelphia	0.010222	0.010563	0.000817	0.009654	0.014677
Phoenix	0.055782	0.05587	0.001405	0.053094	0.060251
Pittsburgh	0.012021	0.012348	0.000786	0.011441	0.015766
Raleigh	0.012303	0.01263	0.000797	0.011706	0.016175
Salt_Lake_City	0.043863	0.095343	0.053882	0.040495	0.150194
San_Diego	0.05144	0.051885	0.001349	0.049579	0.057219
San_Francisco	0.052854	0.052792	0.004598	0.050464	0.101341
St_Louis	0.019349	0.019703	0.001685	0.018978	0.038271
Tampa	0.026895	0.027334	0.000985	0.026192	0.031409
Washington_DC	0.007678	0.007957	0.000814	0.007369	0.015226
Packet Size	16	16	16	16	16
Boston_ROUTER	0.000221	0.000215	4.75E-05	9.57E-05	0.000304
Atlanta	0.017945	0.018289	0.001575	0.017633	0.035715
Austin	0.033801	0.034141	0.003427	0.032237	0.088279
Boston_FTP	0.000363	0.000354	4.45E-05	0.000246	0.00044
Buffalo	0.010442	0.01066	0.000554	0.010024	0.013173
Chicago	0.020551	0.019632	0.001233	0.017125	0.021302
Dallas	0.030383	0.030933	0.003435	0.029091	0.082647
Aurora	0.050644	0.07594	0.047589	0.032596	0.316506
Detroit	0.022078	0.022407	0.000755	0.021569	0.025815
Houston	0.032341	0.042542	0.017206	0.030727	0.160535
Kansas_City	0.023574	0.023903	0.000765	0.023045	0.02754
Las_Vegas	0.060534	0.060604	0.001472	0.05776	0.066167
Los_Angeles	0.050654	0.051334	0.004329	0.050156	0.100726
Miami	0.029473	0.029785	0.000765	0.028901	0.033137
New_Orleans	0.03818	0.046681	0.016759	0.036395	0.086561
New_York	0.003685	0.00384	0.000434	0.003566	0.00745
Orlando	0.02529	0.025582	0.00074	0.024757	0.028148
Philadelphia	0.010216	0.010511	0.000769	0.009651	0.013547
Phoenix	0.055712	0.055828	0.001388	0.053072	0.060433
Pittsburgh	0.012015	0.012322	0.000773	0.011421	0.015786
Raleigh	0.012282	0.012601	0.000785	0.011695	0.016512
Salt_Lake_City	0.043639	0.09529	0.05392	0.040486	0.150232
San_Diego	0.051922	0.052039	0.001276	0.049565	0.056928
San_Francisco	0.052872	0.052779	0.004592	0.050453	0.101312
St_Louis	0.019336	0.019692	0.001683	0.018969	0.038136
Tampa	0.026888	0.027301	0.00096	0.026178	0.031247
Washington_DC	0.007694	0.007944	0.000805	0.00736	0.015102

Table A.11 Chicago Polling station MCI Network 53, 32, and 16 Bytes Packet Results

Simulation Results	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	53	53	53	53	53
Chicago_ROUTER	0.000177	0.000176	5.03E-05	9.95E-05	0.000309
Atlanta	0.012648	0.012958	0.001153	0.012339	0.025069
Austin	0.02959	0.030969	0.005643	0.023861	0.081796
Boston	0.020442	0.019691	0.001244	0.01725	0.021516
Buffalo	0.023502	0.022608	0.001226	0.02007	0.024438
Chicago_FTP	0.000319	0.000322	4.88E-05	0.000236	0.000437
Dallas	0.028304	0.029116	0.005505	0.015664	0.089843
Aurora	0.029853	0.056406	0.044793	0.019082	0.295349
Detroit	0.006515	0.006668	0.001181	0.004906	0.009438
Houston	0.026063	0.026507	0.002268	0.025454	0.051404
Kansas_City	0.012419	0.011709	0.001233	0.009312	0.013507
Las_Vegas	0.060513	0.060546	0.005482	0.047366	0.070091
Los_Angeles	0.050046	0.052037	0.007586	0.041125	0.134031
Miami	0.024173	0.024488	0.000742	0.023638	0.028202
New_Orleans	0.031826	0.03221	0.000943	0.031138	0.035564
New_York	0.016921	0.015888	0.002115	0.013544	0.034347
Orlando	0.02	0.020305	0.000737	0.019493	0.023651
Philadelphia	0.015891	0.015716	0.001012	0.013874	0.018623
Phoenix	0.055093	0.054479	0.006022	0.039753	0.06504
Pittsburgh	0.01777	0.017628	0.000812	0.015636	0.02033
Raleigh	0.018033	0.017963	0.000993	0.015917	0.020694
Salt_Lake_City	0.02972	0.083108	0.055673	0.026612	0.139908
San_Diego	0.048843	0.049081	0.005903	0.036237	0.061159
San_Francisco	0.036636	0.037199	0.003151	0.03624	0.072856
St_Louis	0.008486	0.007425	0.001635	0.005084	0.017345
Tampa	0.021651	0.022008	0.000919	0.020929	0.025431
Washington_DC	0.012989	0.013254	0.001838	0.011566	0.040553
Packet Size	32	32	32	32	32
Chicago_ROUTER	0.000167	0.000168	4.89E-05	9.18E-05	0.0003
Atlanta	0.012653	0.012945	0.001153	0.012324	0.025097
Austin	0.029343	0.030083	0.005838	0.021903	0.095994
Boston	0.020403	0.019669	0.001233	0.017192	0.021404
Buffalo	0.023412	0.022567	0.001234	0.020173	0.024596
Chicago_FTP	0.000309	0.000314	4.38E-05	0.000233	0.000427
Dallas	0.02934	0.028683	0.005568	0.015676	0.08145
Aurora	0.036122	0.059215	0.046482	0.019019	0.295532
Detroit	0.006365	0.006489	0.001096	0.0049	0.009359
Houston	0.026023	0.026495	0.002267	0.025445	0.051479
Kansas_City	0.012466	0.011683	0.001229	0.00918	0.013852
Las_Vegas	0.060006	0.059414	0.006591	0.047333	0.069992
Los_Angeles	0.050708	0.052657	0.006947	0.041116	0.116956
Miami	0.024158	0.024473	0.000747	0.02362	0.027413
New_Orleans	0.031789	0.032185	0.000951	0.031116	0.036092

New_York	0.016806	0.015874	0.00211	0.013531	0.034329
Orlando	0.019982	0.020292	0.00075	0.019476	0.023298
Philadelphia	0.015914	0.01584	0.001003	0.013845	0.018582
Phoenix	0.054611	0.054466	0.006327	0.039803	0.064763
Pittsburgh	0.01734	0.017357	0.001171	0.015615	0.020315
Raleigh	0.018017	0.018054	0.001188	0.015897	0.020597
Salt_Lake_City	0.029863	0.083101	0.055651	0.026533	0.139689
San_Diego	0.049691	0.049612	0.006298	0.036053	0.06111
San_Francisco	0.036624	0.037196	0.003158	0.036224	0.073003
St_Louis	0.008531	0.007421	0.001618	0.00506	0.017371
Tampa	0.021608	0.022009	0.000955	0.020907	0.025628
Washington_DC	0.012958	0.013213	0.00185	0.01156	0.040473
Packet Size	16	16	16	16	16
Chicago_ROUTER	0.000162	0.000162	4.92E-05	7.9E-05	0.000296
Atlanta	0.012636	0.012944	0.001158	0.012315	0.025089
Austin	0.029319	0.030077	0.005923	0.02189	0.095977
Boston	0.020211	0.019657	0.001232	0.017267	0.021488
Buffalo	0.023272	0.022572	0.001224	0.020165	0.024489
Chicago_FTP	0.000305	0.000309	4.26E-05	0.000229	0.000431
Dallas	0.028231	0.028104	0.005387	0.015926	0.081436
Aurora	0.035782	0.057637	0.045088	0.018911	0.295519
Detroit	0.006502	0.006596	0.001132	0.004891	0.009477
Houston	0.025985	0.026479	0.002271	0.025434	0.051382
Kansas_City	0.012436	0.011673	0.001231	0.009224	0.013462
Las_Vegas	0.059123	0.058423	0.006253	0.047325	0.070512
Los_Angeles	0.050627	0.0526	0.006973	0.041113	0.116945
Miami	0.02416	0.024455	0.000745	0.023608	0.027888
New_Orleans	0.031772	0.032177	0.000949	0.031102	0.036618
New_York	0.016853	0.015859	0.002113	0.013522	0.034307
Orlando	0.019944	0.020254	0.000724	0.019473	0.02341
Philadelphia	0.015914	0.015918	0.00111	0.013842	0.018554
Phoenix	0.053936	0.053627	0.006324	0.039629	0.065429
Pittsburgh	0.017716	0.017631	0.001003	0.015604	0.020262
Raleigh	0.01797	0.017975	0.001221	0.015894	0.02065
Salt_Lake_City	0.029693	0.083073	0.055674	0.026516	0.139897
San_Diego	0.049494	0.049624	0.005928	0.036261	0.060816
San_Francisco	0.036615	0.03718	0.003159	0.036212	0.072997
St_Louis	0.008527	0.007404	0.001629	0.005051	0.017363
Tampa	0.02158	0.021959	0.000928	0.020894	0.026199
Washington_DC	0.012888	0.013185	0.001883	0.011551	0.040464

Table A.12 San Francisco Polling station MCI Network 53, 32, and 16 Bytes Packet Results

Simulation Results	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	53	53	53	53	53
San_Francisco_ROUTER	0.000122	0.000166	0.000106	6.52E-05	0.000343
Atlanta	0.052067	0.052956	0.006608	0.045656	0.129967
Austin	0.053912	0.05199	0.011058	0.03709	0.14112
Boston	0.051121	0.051416	0.000751	0.050541	0.054165
Buffalo	0.054008	0.054331	0.000765	0.053445	0.057698
Chicago	0.036754	0.037066	0.000724	0.036275	0.040485
Dallas	0.047269	0.049787	0.00687	0.037134	0.070985
Aurora	0.064882	0.087085	0.045962	0.047015	0.345182
Detroit	0.040865	0.041155	0.000726	0.040371	0.04446
Houston	0.032896	0.033341	0.002814	0.032564	0.06543
Kansas_City	0.038032	0.038315	0.000729	0.03746	0.04188
Las_Vegas	0.017899	0.018311	0.00093	0.017228	0.023039
Los_Angeles	0.006774	0.007029	0.000734	0.006457	0.013161
Miami	0.064841	0.064844	0.004158	0.058016	0.074505
New_Orleans	0.038786	0.039086	0.000738	0.038251	0.042317
New_York	0.047214	0.047779	0.004044	0.046954	0.094221
Orlando	0.060276	0.060417	0.006902	0.052817	0.177041
Philadelphia	0.048328	0.048662	0.000766	0.047804	0.051572
Phoenix	0.013033	0.013342	0.000717	0.012543	0.017024
Pittsburgh	0.050182	0.0505	0.000757	0.049592	0.053644
Raleigh	0.050441	0.050737	0.000764	0.049857	0.054037
Salt_Lake_City	0.058085	0.109685	0.053897	0.054906	0.164647
San_Diego	0.009032	0.009368	0.000731	0.008506	0.012909
San_Francisco_FTP	0.000421	0.000406	5.16E-05	0.00027	0.000491
St_Louis	0.033695	0.03416	0.002887	0.033354	0.067156
Tampa	0.062635	0.061937	0.00397	0.054258	0.071785
Washington_DC	0.045821	0.046368	0.003919	0.045515	0.091451
Packet Size	32	32	32	32	32
San_Francisco_ROUTER	0.000115	0.000155	0.000103	5.85E-05	0.000333
Atlanta	0.05209	0.053068	0.006829	0.045646	0.128463
Austin	0.04808	0.050343	0.010457	0.037067	0.139186
Boston	0.051095	0.051398	0.000765	0.050522	0.054782
Buffalo	0.053987	0.054307	0.000758	0.053427	0.057477
Chicago	0.036741	0.037045	0.000725	0.036257	0.04083
Dallas	0.046989	0.049335	0.007324	0.037024	0.072075
Aurora	0.065182	0.091198	0.04815	0.046995	0.345204
Detroit	0.040833	0.041145	0.000738	0.040344	0.04447
Houston	0.032887	0.03333	0.002817	0.03255	0.065654
Kansas_City	0.037977	0.038299	0.000737	0.037449	0.041461
Las_Vegas	0.017884	0.018273	0.00092	0.017198	0.023686
Los_Angeles	0.006763	0.007007	0.00073	0.006443	0.013134
Miami	0.06474	0.064698	0.00416	0.058003	0.074666

New_Orleans	0.038749	0.039062	0.000751	0.038229	0.043553
New_York	0.047205	0.04777	0.004046	0.046941	0.094198
Orlando	0.059185	0.059982	0.006642	0.05385	0.142798
Philadelphia	0.048315	0.04864	0.000777	0.047787	0.051813
Phoenix	0.013033	0.013321	0.000721	0.012525	0.016369
Pittsburgh	0.050153	0.050469	0.000751	0.049565	0.053542
Raleigh	0.050394	0.050715	0.000773	0.04983	0.054051
Salt_Lake_City	0.057973	0.109649	0.053921	0.054849	0.164567
San_Diego	0.009045	0.00934	0.000718	0.008487	0.012418
San_Francisco_FTP	0.000409	0.000396	4.97E-05	0.000275	0.000481
St_Louis	0.033686	0.034147	0.002887	0.033338	0.067137
Tampa	0.061041	0.061546	0.003481	0.054891	0.071941
Washington_DC	0.045801	0.046358	0.003921	0.045502	0.091487
Packet Size	16	16	16	16	16
San_Francisco_ROUTER	0.00011	0.000153	0.000105	5.4E-05	0.000337
Atlanta	0.053063	0.053659	0.007265	0.045635	0.154783
Austin	0.053794	0.051078	0.01057	0.037071	0.139173
Boston	0.051081	0.051379	0.000749	0.050503	0.054903
Buffalo	0.053965	0.054267	0.000735	0.053415	0.05756
Chicago	0.03672	0.037021	0.000707	0.036246	0.039964
Dallas	0.048037	0.049958	0.006923	0.03701	0.071011
Aurora	0.064783	0.08657	0.045895	0.046982	0.345191
Detroit	0.040833	0.041103	0.000696	0.040333	0.045209
Houston	0.032885	0.033327	0.002818	0.032532	0.065646
Kansas_City	0.037983	0.038282	0.000716	0.03743	0.041314
Las_Vegas	0.017863	0.018248	0.000915	0.017185	0.022041
Los_Angeles	0.006745	0.006995	0.000733	0.006434	0.013089
Miami	0.063662	0.064633	0.004098	0.057989	0.0743
New_Orleans	0.038748	0.039045	0.000735	0.038218	0.043181
New_York	0.047204	0.047753	0.004045	0.046932	0.094302
Orlando	0.059224	0.059922	0.006943	0.052759	0.154791
Philadelphia	0.048306	0.048629	0.000768	0.047767	0.051725
Phoenix	0.013007	0.013299	0.000716	0.012512	0.016377
Pittsburgh	0.05014	0.050469	0.000766	0.049554	0.053485
Raleigh	0.050389	0.050715	0.000775	0.049827	0.053692
Salt_Lake_City	0.05794	0.10964	0.053919	0.054792	0.1645
San_Diego	0.008988	0.009328	0.00073	0.008476	0.012328
San_Francisco_FTP	0.000405	0.000392	4.97E-05	0.000262	0.000482
St_Louis	0.033687	0.034134	0.002884	0.033329	0.067211
Tampa	0.063041	0.062542	0.00391	0.055093	0.071769
Washington_DC	0.045793	0.046349	0.003923	0.045493	0.091712

Table A.13 San Francisco Polling station AT&T to MCI Network 32 and 16 Bytes Packet Results

AT&T to MCI	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	32	32	32	32	32
ATT_SF_FTP	0.00071	0.000713	4.64E-05	0.00063	0.000902
ATT_SF_ROUTER	0.00067	0.000679	3.48E-05	0.00063	0.00079
MCI_Atlanta	0.069283	0.086333	0.054631	0.046366	0.410403
MCI_Austin	0.039752	0.052404	0.019633	0.039244	0.190923
MCI_Boston	0.055859	0.067609	0.024924	0.051684	0.22833
MCI_Buffalo	0.04633	0.057499	0.017814	0.044402	0.148268
MCI_Chicago	0.049489	0.05551	0.02477	0.035769	0.188546
MCI_Dallas	0.036857	0.049703	0.020142	0.036188	0.190157
MCI_Aurora	0.022347	0.02764	0.012516	0.021865	0.108441
MCI_Detroit	0.040958	0.052093	0.015191	0.040377	0.112325
MCI_Houston	0.044986	0.066084	0.055789	0.032504	0.393102
MCI_Kansas_City	0.032575	0.040427	0.015458	0.031674	0.125135
MCI_Las_Vegas	0.012487	0.014928	0.00877	0.012187	0.087948
MCI_Los_Angeles	0.02094	0.04349	0.055955	0.007836	0.367898
MCI_Miami	0.082193	0.09918	0.056108	0.059361	0.43347
MCI_New_Orleans	0.046582	0.060137	0.030936	0.039138	0.22146
MCI_New_York	0.053387	0.065263	0.024585	0.04848	0.206361
MCI_Orlando	0.076283	0.091767	0.054931	0.055257	0.428527
MCI_Philadelphia	0.067004	0.083653	0.054666	0.050169	0.414796
MCI_Phoenix	0.033551	0.053667	0.05703	0.01439	0.377509
MCI_Pittsburgh	0.063409	0.080573	0.054719	0.0453	0.414247
MCI_Raleigh	0.067879	0.084716	0.054235	0.052786	0.417268
MCI_Salt_Lake_City	0.013953	0.018764	0.013052	0.013531	0.125266
MCI_San_Diego	0.022494	0.045977	0.056973	0.009768	0.373665
MCI_San_Francisco	0.004382	0.013476	0.01958	0.001729	0.157061
MCI_St_Louis	0.080554	0.088942	0.048342	0.037857	0.341684
MCI_Tampa	0.070859	0.08888	0.054908	0.057293	0.434019
MCI_Washington_DC	0.060807	0.079246	0.053666	0.047154	0.408755
Packet Size	16	16	16	16	16
ATT_SF_FTP	0.000706	0.000711	4.82E-05	0.000626	0.000877
ATT_SF_ROUTER	0.000666	0.00068	3.57E-05	0.000626	0.000797
MCI_Atlanta	0.069924	0.084259	0.060489	0.045968	0.561276
MCI_Austin	0.039688	0.051745	0.018455	0.039221	0.144201
MCI_Boston	0.054141	0.065185	0.021161	0.051666	0.202608
MCI_Buffalo	0.051265	0.059283	0.01874	0.044468	0.133037
MCI_Chicago	0.050992	0.055433	0.022984	0.035777	0.189145
MCI_Dallas	0.036813	0.048018	0.017157	0.036201	0.124948
MCI_Aurora	0.022309	0.02587	0.009306	0.021879	0.095599
MCI_Detroit	0.040955	0.051647	0.016538	0.040384	0.138629
MCI_Houston	0.042592	0.062544	0.060251	0.032595	0.543701
MCI_Kansas_City	0.032557	0.039115	0.012653	0.031692	0.10521
MCI_Las_Vegas	0.012454	0.014918	0.007607	0.01219	0.054454

MCI_Los_Angeles	0.015611	0.039628	0.061848	0.007803	0.522616
MCI_Miami	0.082779	0.1011	0.064744	0.05827	0.607136
MCI_New_Orleans	0.047934	0.060452	0.026495	0.03958	0.196312
MCI_New_York	0.052229	0.062668	0.020672	0.04852	0.202058
MCI_Orlando	0.073623	0.092907	0.064137	0.05515	0.606038
MCI_Philadelphia	0.068255	0.084813	0.060837	0.049577	0.5621
MCI_Phoenix	0.027186	0.04938	0.062444	0.01402	0.525088
MCI_Pittsburgh	0.058245	0.079223	0.061302	0.048229	0.566768
MCI_Raleigh	0.069742	0.087057	0.062103	0.053737	0.567867
MCI_Salt_Lake_City	0.01399	0.019206	0.013083	0.013523	0.126088
MCI_San_Diego	0.019181	0.042597	0.062621	0.009968	0.525637
MCI_San_Francisco	0.005556	0.036466	0.076823	0.001727	0.606587
MCI_St_Louis	0.089986	0.081811	0.05107	0.002255	0.225749
MCI_Tampa	0.062965	0.071634	0.021708	0.040878	0.176136
MCI_Washington_DC	0.061628	0.081039	0.061467	0.04797	0.564022

Table A.14 San Francisco Polling station MCI to AT&T Network 32 and 16 Bytes Packet Results

AT&T to MCI	Median	Mean	Std Dev	Minimum	Maximum
Packet Size	32	32	32	32	32
MCI_SF_FTP	0.000376	0.000365	4.39E-05	0.000267	0.000434
MCI_SF_ROUTER	0.000234	0.000226	4.59E-05	0.000125	0.000309
ATT_Atlanta	0.084839	0.09451	0.045428	0.048212	0.317115
ATT_Austin	0.053626	0.062571	0.032135	0.036593	0.213969
ATT_Cambridge	0.062963	0.073361	0.028636	0.051664	0.209528
ATT_Buffalo	0.052481	0.061809	0.020643	0.044419	0.143046
ATT_Chicago	0.042098	0.052424	0.019916	0.035808	0.141948
ATT_Dallas	0.044129	0.053211	0.020389	0.036668	0.141398
ATT_Denver	0.026989	0.037475	0.018341	0.022352	0.123274
ATT_Detroit	0.049305	0.058404	0.020478	0.040751	0.142497
ATT_Houston	0.059669	0.079262	0.063504	0.034031	0.506174
ATT_Kansas_City	0.056473	0.066861	0.028445	0.038782	0.22175
ATT_Las_Vegas	0.017424	0.026548	0.017189	0.013077	0.094103
ATT_Los_Angeles	0.013031	0.021047	0.016361	0.009233	0.09273
ATT_Miami	0.082977	0.100124	0.060487	0.05485	0.531788
ATT_New_Orleans	0.050728	0.074911	0.064	0.039689	0.511013
ATT_New_York	0.06783	0.077108	0.031958	0.048472	0.219134
ATT_Orlando	0.082849	0.102097	0.066419	0.049445	0.54756
ATT_Philadelphia	0.068121	0.078339	0.033751	0.049278	0.223103
ATT_Phoenix	0.020585	0.025863	0.014302	0.014172	0.088644
ATT_Pittsburgh	0.066041	0.076883	0.032912	0.046098	0.244798
ATT_Raleigh	0.055279	0.070704	0.028849	0.051194	0.235389
ATT_Salt_Lake_City	0.019989	0.029273	0.017497	0.014388	0.094652
ATT_San_Diego	0.016487	0.025099	0.016754	0.011751	0.093554
ATT_San_Francisco	0.008186	0.015402	0.015864	0.003663	0.082601
ATT_St_Louis	0.051337	0.062812	0.029939	0.034943	0.211835
ATT_Tampa	0.08507	0.104121	0.066284	0.051458	0.547807
ATT_Washington_DC	0.074047	0.082072	0.033468	0.047043	0.234902
Packet Size	16	16	16	16	16
MCI_SF_FTP	0.000365	0.000358	4.66E-05	0.000262	0.000432
MCI_SF_ROUTER	0.000229	0.000219	4.78E-05	0.00012	0.000287
ATT_Atlanta	0.074969	0.08773	0.042634	0.048182	0.269616
ATT_Austin	0.05015	0.062038	0.034159	0.036444	0.257152
ATT_Cambridge	0.055852	0.068761	0.024707	0.051723	0.178286
ATT_Buffalo	0.048339	0.060701	0.021522	0.044365	0.144075
ATT_Chicago	0.040573	0.05133	0.020928	0.035868	0.142427
ATT_Dallas	0.041137	0.052364	0.021335	0.036101	0.142977
ATT_Denver	0.026987	0.037313	0.019874	0.02237	0.127598
ATT_Detroit	0.046484	0.058235	0.021713	0.041329	0.143526
ATT_Houston	0.056439	0.069119	0.044967	0.033959	0.2817
ATT_Kansas_City	0.053111	0.064504	0.029575	0.038798	0.255417
ATT_Las_Vegas	0.017407	0.026219	0.017911	0.013142	0.112505

ATT_Los_Angeles	0.012686	0.020621	0.017465	0.009153	0.111407
ATT_Miami	0.079927	0.092262	0.041536	0.056965	0.293645
ATT_New_Orleans	0.050574	0.069675	0.046987	0.039596	0.309468
ATT_New_York	0.069343	0.076626	0.02977	0.048488	0.212116
ATT_Orlando	0.07406	0.091695	0.049688	0.049441	0.309594
ATT_Philadelphia	0.069697	0.082207	0.036654	0.049324	0.284451
ATT_Phoenix	0.021365	0.027972	0.018274	0.014047	0.114795
ATT_Pittsburgh	0.061342	0.07323	0.028823	0.047666	0.199224
ATT_Raleigh	0.061198	0.070778	0.025521	0.051158	0.17117
ATT_Salt_Lake_City	0.019925	0.029274	0.018431	0.01494	0.114324
ATT_San_Diego	0.016277	0.025349	0.017669	0.0118	0.111956
ATT_San_Francisco	0.008132	0.01432	0.016323	0.004262	0.110857
ATT_St_Louis	0.051899	0.06379	0.030722	0.035085	0.246503
ATT_Tampa	0.076544	0.094254	0.04976	0.051068	0.310069
ATT_Washington_DC	0.072765	0.083112	0.035155	0.047065	0.185605

Table A.15 San Francisco Polling station AT&T to MCI Link Bandwidth Results

AT&T to MCI	Median	Mean	Std Dev	Minimum	Maximum
Link Bandwidth	OC12	OC12	OC12	OC12	OC12
ATT_SF_FTP	0.000721	0.000728	5.27E-05	0.00063	0.001059
ATT_SF_ROUTER	0.00069	0.000692	3.72E-05	0.00063	0.00083
MCI_Atlanta	0.051954	0.051585	0.00579	0.045645	0.099768
MCI_Austin	0.037991	0.038018	6.85E-05	0.037991	0.038245
MCI_Boston	0.050827	0.051891	0.005962	0.050476	0.101008
MCI_Buffalo	0.043187	0.043238	9.33E-05	0.043146	0.043587
MCI_Chicago	0.034561	0.034601	9.92E-05	0.034501	0.03499
MCI_Dallas	0.034881	0.034934	0.000109	0.034801	0.035309
MCI_Aurora	0.020701	0.020755	0.000108	0.02062	0.021081
MCI_Detroit	0.039276	0.039288	0.000118	0.039125	0.039614
MCI_Houston	0.034392	0.034482	0.004155	0.031437	0.068415
MCI_Kansas_City	0.034652	0.034908	0.004279	0.03026	0.048998
MCI_Las_Vegas	0.010966	0.010974	0.000124	0.010806	0.011266
MCI_Los_Angeles	0.006594	0.006603	0.000126	0.006405	0.006885
MCI_Miami	0.064351	0.061774	0.003876	0.057034	0.068142
MCI_New_Orleans	0.041425	0.041822	0.001291	0.040319	0.044355
MCI_New_York	0.049507	0.049518	0.00138	0.04728	0.051213
MCI_Orlando	0.054675	0.057113	0.003764	0.052959	0.063636
MCI_Philadelphia	0.051041	0.050878	0.00119	0.048972	0.0533
MCI_Phoenix	0.013439	0.01339	0.000372	0.012563	0.014671
MCI_Pittsburgh	0.048996	0.048379	0.002825	0.043819	0.053872
MCI_Raleigh	0.053103	0.053388	0.000802	0.052043	0.056325
MCI_Salt_Lake_City	0.012365	0.012367	0.000129	0.012124	0.012624
MCI_San_Diego	0.008823	0.008888	0.000308	0.008435	0.010301
MCI_San_Francisco	0.00069	0.000701	3.76E-05	0.00057	0.00083
MCI_St_Louis	0.039673	0.048624	0.02138	0.033981	0.091464
MCI_Tampa	0.056203	0.05882	0.003784	0.054292	0.065524
MCI_Washington_DC	0.048146	0.047984	0.00088	0.046304	0.049982
Link Bandwidth	OC3	OC3	OC3	OC3	OC3
ATT_SF_FTP	0.000728	0.000726	4.73E-05	0.00063	0.00086
ATT_SF_ROUTER	0.00069	0.000695	3.54E-05	0.00063	0.00079
MCI_Atlanta	0.047073	0.049479	0.008902	0.044333	0.154381
MCI_Austin	0.038001	0.03802	5.73E-05	0.037999	0.03826
MCI_Boston	0.050837	0.051801	0.006031	0.050504	0.101684
MCI_Buffalo	0.043198	0.04326	0.000107	0.043155	0.043597
MCI_Chicago	0.03457	0.03462	0.000107	0.03451	0.034951
MCI_Dallas	0.034889	0.034933	0.000101	0.03481	0.035231
MCI_Aurora	0.020759	0.020775	0.000111	0.020629	0.02115
MCI_Detroit	0.039229	0.039278	0.000111	0.039126	0.039624
MCI_Houston	0.032594	0.03319	0.005136	0.031371	0.089718
MCI_Kansas_City	0.03061	0.030843	0.00119	0.03027	0.040134
MCI_Las_Vegas	0.010997	0.010998	0.000119	0.010797	0.011237
MCI_Los_Angeles	0.006613	0.006614	0.000128	0.006414	0.006992

MCI_Miami	0.063007	0.063469	0.004996	0.055931	0.070741
MCI_New_Orleans	0.040237	0.040501	0.001892	0.037779	0.043996
MCI_New_York	0.047662	0.047865	0.00055	0.047148	0.049882
MCI_Orlando	0.054365	0.055316	0.002082	0.053035	0.060585
MCI_Philadelphia	0.051004	0.050922	0.000956	0.048398	0.053333
MCI_Phoenix	0.012903	0.012967	0.000299	0.012523	0.01394
MCI_Pittsburgh	0.049099	0.050345	0.002242	0.047069	0.054253
MCI_Raleigh	0.052367	0.052889	0.001621	0.050002	0.056135
MCI_Salt_Lake_City	0.012415	0.012409	0.000126	0.012153	0.01265
MCI_San_Diego	0.008762	0.008774	0.000203	0.008463	0.010105
MCI_San_Francisco	0.00069	0.000697	3.31E-05	0.00063	0.00081
MCI_St_Louis	0.036685	0.052734	0.031795	0.033582	0.143068
MCI_Tampa	0.056534	0.057243	0.001836	0.055366	0.062924
MCI_Washington_DC	0.047941	0.048048	0.001003	0.045902	0.049775
Link Bandwidth	T1	T1	T1	T1	T1
ATT_SF_FTP	0.00071	0.000713	4.64E-05	0.00063	0.000902
ATT_SF_ROUTER	0.00067	0.000679	3.48E-05	0.00063	0.00079
MCI_Atlanta	0.069283	0.086333	0.054631	0.046366	0.410403
MCI_Austin	0.039752	0.052404	0.019633	0.039244	0.190923
MCI_Boston	0.055859	0.067609	0.024924	0.051684	0.22833
MCI_Buffalo	0.04633	0.057499	0.017814	0.044402	0.148268
MCI_Chicago	0.049489	0.05551	0.02477	0.035769	0.188546
MCI_Dallas	0.036857	0.049703	0.020142	0.036188	0.190157
MCI_Aurora	0.022347	0.02764	0.012516	0.021865	0.108441
MCI_Detroit	0.040958	0.052093	0.015191	0.040377	0.112325
MCI_Houston	0.044986	0.066084	0.055789	0.032504	0.393102
MCI_Kansas_City	0.032575	0.040427	0.015458	0.031674	0.125135
MCI_Las_Vegas	0.012487	0.014928	0.00877	0.012187	0.087948
MCI_Los_Angeles	0.02094	0.04349	0.055955	0.007836	0.367898
MCI_Miami	0.082193	0.09918	0.056108	0.059361	0.43347
MCI_New_Orleans	0.046582	0.060137	0.030936	0.039138	0.22146
MCI_New_York	0.053387	0.065263	0.024585	0.04848	0.206361
MCI_Orlando	0.076283	0.091767	0.054931	0.055257	0.428527
MCI_Philadelphia	0.067004	0.083653	0.054666	0.050169	0.414796
MCI_Phoenix	0.033551	0.053667	0.05703	0.01439	0.377509
MCI_Pittsburgh	0.063409	0.080573	0.054719	0.0453	0.414247
MCI_Raleigh	0.067879	0.084716	0.054235	0.052786	0.417268
MCI_Salt_Lake_City	0.013953	0.018764	0.013052	0.013531	0.125266
MCI_San_Diego	0.022494	0.045977	0.056973	0.009768	0.373665
MCI_San_Francisco	0.004382	0.013476	0.01958	0.001729	0.157061
MCI_St_Louis	0.080554	0.088942	0.048342	0.037857	0.341684
MCI_Tampa	0.070859	0.08888	0.054908	0.057293	0.434019
MCI_Washington_DC	0.060807	0.079246	0.053666	0.047154	0.408755

Table A.16 San Francisco Polling station MCI to AT&T Link Bandwidth Results

MCI to AT&T	Median	Mean	Std Dev	Minimum	Maximum
Link Bandwidth	OC12	OC12	OC12	OC12	OC12
MCI_SF_FTP	0.00036	0.000361	4.71E-05	0.000267	0.000434
MCI_SF_ROUTER	0.000233	0.000221	5.17E-05	0.000117	0.0003
ATT_Atlanta	0.053233	0.052391	0.002422	0.048117	0.056571
ATT_Austin	0.037013	0.037073	0.000341	0.036625	0.038143
ATT_Cambridge	0.051031	0.051157	0.000668	0.050489	0.05324
ATT_Buffalo	0.043209	0.043226	5.15E-05	0.043168	0.043419
ATT_Chicago	0.034574	0.034598	6.2E-05	0.034514	0.034754
ATT_Dallas	0.034893	0.034903	5.85E-05	0.034813	0.035082
ATT_Denver	0.020712	0.020722	5.86E-05	0.020633	0.020873
ATT_Detroit	0.039212	0.039232	5.7E-05	0.039148	0.039391
ATT_Houston	0.033499	0.033568	0.000477	0.03276	0.034797
ATT_Kansas_City	0.041202	0.042005	0.002999	0.038033	0.046066
ATT_Las_Vegas	0.01092	0.010918	5.76E-05	0.010798	0.011028
ATT_Los_Angeles	0.006497	0.006515	6.64E-05	0.006407	0.006666
ATT_Miami	0.06118	0.060629	0.002724	0.056114	0.065168
ATT_New_Orleans	0.039391	0.039519	0.000593	0.038457	0.041213
ATT_New_York	0.047938	0.048037	0.000488	0.047142	0.049896
ATT_Orlando	0.049156	0.049205	0.000477	0.048297	0.05093
ATT_Philadelphia	0.049103	0.049226	0.000576	0.048087	0.051358
ATT_Phoenix	0.01399	0.014062	0.000621	0.012792	0.015624
ATT_Pittsburgh	0.047318	0.04739	0.002856	0.04382	0.052144
ATT_Raleigh	0.051112	0.05114	0.000553	0.050043	0.052984
ATT_Salt_Lake_City	0.012227	0.012227	6.19E-05	0.012106	0.012357
ATT_San_Diego	0.008565	0.008563	5.35E-05	0.008425	0.008705
ATT_San_Francisco	0.000411	0.000402	4.92E-05	0.000267	0.000524
ATT_St_Louis	0.036765	0.036111	0.00135	0.033596	0.038189
ATT_Tampa	0.050739	0.0508	0.000543	0.049766	0.052737
ATT_Washington_DC	0.046583	0.046636	0.000536	0.045734	0.04825
Link Bandwidth	OC3	OC3	OC3	OC3	OC3
MCI_SF_FTP	0.000369	0.000365	4.49E-05	0.000275	0.00045
MCI_SF_ROUTER	0.000225	0.000222	5.08E-05	0.000117	0.000306
ATT_Atlanta	0.050237	0.051008	0.007396	0.045708	0.116558
ATT_Austin	0.036401	0.036734	0.003958	0.035256	0.069247
ATT_Cambridge	0.050639	0.051316	0.004763	0.050499	0.101097
ATT_Buffalo	0.043207	0.043236	5.85E-05	0.043173	0.043425
ATT_Chicago	0.03457	0.034591	5.95E-05	0.034504	0.034819
ATT_Dallas	0.034887	0.034906	6.22E-05	0.034811	0.0351
ATT_Denver	0.020699	0.020716	5.87E-05	0.020626	0.020921
ATT_Detroit	0.039222	0.039233	6.01E-05	0.03913	0.039387
ATT_Houston	0.033028	0.033674	0.00393	0.032642	0.065633
ATT_Kansas_City	0.041177	0.041743	0.003983	0.037498	0.075116
ATT_Las_Vegas	0.010894	0.010906	6.32E-05	0.010776	0.011064
ATT_Los_Angeles	0.006511	0.006519	6.2E-05	0.006395	0.006684

ATT_Miami	0.059839	0.059024	0.005105	0.052194	0.067387
ATT_New_Orleans	0.038758	0.038937	0.000582	0.038308	0.041614
ATT_New_York	0.047438	0.047629	0.000496	0.047067	0.049427
ATT_Orlando	0.048684	0.048821	0.000523	0.048212	0.050453
ATT_Philadelphia	0.048759	0.048862	0.000699	0.047891	0.050979
ATT_Phoenix	0.013887	0.013996	0.000517	0.012937	0.015993
ATT_Pittsburgh	0.046968	0.047247	0.002587	0.043812	0.052775
ATT_Raleigh	0.050349	0.050681	0.00073	0.049922	0.053785
ATT_Salt_Lake_City	0.012252	0.012245	5.94E-05	0.012095	0.012382
ATT_San_Diego	0.008553	0.008551	5.86E-05	0.008406	0.008692
ATT_San_Francisco	0.000414	0.000412	4.05E-05	0.000267	0.000517
ATT_St_Louis	0.03666	0.036511	0.001625	0.033465	0.039595
ATT_Tampa	0.050095	0.050281	0.000514	0.049663	0.051885
ATT_Washington_DC	0.046039	0.046264	0.000553	0.045644	0.048275
Link Bandwidth	T1	T1	T1	T1	T1
MCI_SF_FTP	0.000376	0.000365	4.39E-05	0.000267	0.000434
MCI_SF_ROUTER	0.000234	0.000226	4.59E-05	0.000125	0.000309
ATT_Atlanta	0.084839	0.09451	0.045428	0.048212	0.317115
ATT_Austin	0.053626	0.062571	0.032135	0.036593	0.213969
ATT_Cambridge	0.062963	0.073361	0.028636	0.051664	0.209528
ATT_Buffalo	0.052481	0.061809	0.020643	0.044419	0.143046
ATT_Chicago	0.042098	0.052424	0.019916	0.035808	0.141948
ATT_Dallas	0.044129	0.053211	0.020389	0.036668	0.141398
ATT_Denver	0.026989	0.037475	0.018341	0.022352	0.123274
ATT_Detroit	0.049305	0.058404	0.020478	0.040751	0.142497
ATT_Houston	0.059669	0.079262	0.063504	0.034031	0.506174
ATT_Kansas_City	0.056473	0.066861	0.028445	0.038782	0.22175
ATT_Las_Vegas	0.017424	0.026548	0.017189	0.013077	0.094103
ATT_Los_Angeles	0.013031	0.021047	0.016361	0.009233	0.09273
ATT_Miami	0.082977	0.100124	0.060487	0.05485	0.531788
ATT_New_Orleans	0.050728	0.074911	0.064	0.039689	0.511013
ATT_New_York	0.06783	0.077108	0.031958	0.048472	0.219134
ATT_Orlando	0.082849	0.102097	0.066419	0.049445	0.54756
ATT_Philadelphia	0.068121	0.078339	0.033751	0.049278	0.223103
ATT_Phoenix	0.020585	0.025863	0.014302	0.014172	0.088644
ATT_Pittsburgh	0.066041	0.076883	0.032912	0.046098	0.244798
ATT_Raleigh	0.055279	0.070704	0.028849	0.051194	0.235389
ATT_Salt_Lake_City	0.019989	0.029273	0.017497	0.014388	0.094652
ATT_San_Diego	0.016487	0.025099	0.016754	0.011751	0.093554
ATT_San_Francisco	0.008186	0.015402	0.015864	0.003663	0.082601
ATT_St_Louis	0.051337	0.062812	0.029939	0.034943	0.211835
ATT_Tampa	0.08507	0.104121	0.066284	0.051458	0.547807
ATT_Washington_DC	0.074047	0.082072	0.033468	0.047043	0.234902

Bibliography

[AFD98] *Air Force Doctrine Document 2-5 Information Operations*, Department of the Air Force, Aug. 1998.

[ATT03] *AT&T Global IP Backbone Net*, www.business.att.com, May 2003.

[BeC02] Beverly, R. and K. Claffy, *Wide-Area IP Multicast Traffic Characterization (Extended Version)*, CAIDA Technical Report, 2002.

[Car03] Carr C., *Reverse Geographic Location of a Computer Node*, Thesis Air Force Institute of Technology, 2003.

[CCAS01] Chen, I., W. Chiang, S. Adnan, and L. Silacci, *Geographically Speaking*, University of California San Diego, 2001.

[Cla00] Claffy, K., *Measuring the Internet*, IEEE Internet Computing, 2000.

[CLR01] Cormen, T., C. Leiserson, and R. Rivest, *Introduction to Algorithms*, Second Edition, MIT Press, pg 588- 591, 2001.

[DeM96] Denning, D. and P. McDoran, *Location-based Authentication: Grounding Cyberspace for Better Security*, Computer Fraud & Security, 1996.

[DjR01] Djuknic, G. and R. Richton, *Geolocation and Assisted GPS*, Lucent Technologies, 2001.

[DoK01] Dodge, M. and R. Kitchin, *Mapping Cyberspace*, Routledge Publishing, 2001.

[EgM95] Egenhofer, M. and D. Mark, *Naive Geography*, State University of New York at Buffalo, 1995.

[FML03] Fraleigh, C., S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diat, *Packet-Level Traffic Measurements from the Sprint IP Backbone*, IEEE Network Vol. 17, No. 6, 2003.

[HPMC02] Huffaker, B., D. Plummer, D. Moore, and K. Claffy, *TopologyDiscovery by Active Probing*, Symposium on Applications and the Internet (SAINT), 2002.

[HSF85] Harrenstien, K., M. Stahl, and E. Feinler, *Request For Comments 954 Nickname/Whois*, Oct. 1985.

[HyBC03] Hyun, Y., A. Broido, and K. Claffy, *On Third-party Addresses in Traceroute Paths*, Passive and Active Measurement Workshop, 2003.

[Jai91] Jain, R., *The Art of Computer Systems Performance Analysis*, Wiley Professional Computing, John Wiley and Sons, Inc., pg 86, 216 – 218, 221 - 243, pg 1991.

[Ker02] *PING(8) FreeBSD System Manager's Manual*, www.kerneled.com/doc/man/freebsd/man8/ping.html, 2002.

[Map03] *City Center Driving Distances*, www.mapquest.com, Jul. 2003.

[McC00] McCreary, S. and K. Claffy, *Trends in Wide Area IP Traffic Patterns*, ITC Specialist Seminar, 2000.

[Mic03] *Pathping*, www.microsoft.com/technet/winxppro/proddocs/pathping.asp, 2003.

[MCI03] *Our Network*, global.mci.com/about/network, May 2003.

[NSA02] *Network Geolocation Technology Video*, National Security Agency, 2002.

[OPNET03] *OPNET Modeler 10.0 Product Documentation - Standard Models User Guide 16 RIP Model User Guide*, OPNET Technologies, 2003.

[PeN99] Periakaruppan, R., and E. Nemeth, *GTrace – A Graphical Traceroute Tool*, Usenix LISA, 1999.

[SMC01] Shannon, C., D. Moore, and K. Claffy, *Characteristics of Fragmented IP Traffic on Internet Links*, Passive and Active Measurement Workshop, 2001.

[SPr03] *About Sprint*, www.sprintworldwide.com/english/about, May 2003.

[Sun99] *Traceroute*, Sun Operating System version 5.8, Maintenance Commands, 1999

[Web04] *Hot Potato Routing definition*,

www.webopedia.com/TERM/H/hot_potato_routing.html, Jan. 2004.

[Yua01] Yuan, X., *Algorithms for Multi-Constrained Quality of Service Routing*,

IEEE/ACM Transaction on Networking, 2001.

Vita

Duane C. Sorgaard is a Master Sergeant in the U.S. Air Force. He is an M.S. graduate in the Department of Electrical and Computer Engineering, Graduate School of Engineering and Management, Air Force Institute of Technology. He received his B.S. in Computer Science from National University San Diego in 1998. His technical interests include information assurance, information warfare, network and computer systems security.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
23-03-2004	Master's Thesis	March 2003 – March 2004

4. TITLE AND SUBTITLE

GEOGRAPHIC LOCATION OF A COMPUTER NODE EXAMINING A TIME-TO-LOCATION ALGORITHM AND MULTIPLE AUTONOMOUS SYSTEM NETWORKS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Sorgaard, Duane C., Master Sergeant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCS/ENG/04-17

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Mr William Kroah
NSA/R5
Ft Meade, MD 20755
(301) 688-0348

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

To determine the location of a computer on the Internet without resorting to outside information or databases would greatly increase the security abilities of the US Air Force and the Department of Defense. The geographic location of a computer node has been demonstrated on an autonomous system (AS) network, or a network with one system administration focal point. The work shows that a similar technique will work on networks comprised of a multiple AS network. A time-to-location algorithm can successfully resolve a geographic location of a computer node using only latency information from known sites and mathematically calculating the Euclidean distance to those sites from an unknown location on a single AS network. The time-to-location algorithm on a multiple AS network successfully resolves a geographic location 71.4% of the time. Packets are subject to arbitrary delays in the network; and inconsistencies in latency measurements are discovered when attempting to use a time-to location algorithm on a multiple AS network. To improve accuracy in a multiple AS network, a time-to-location algorithm needs to calculate the link bandwidth when attempting to geographically locate a computer node on a multiple AS network.

15. SUBJECT TERMS

Geographical Information Systems, Information Systems, Network Topology, Networks, Computer Communications, Computer Networks.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Baldwin, Rusty O., Major, USAF
U	U	U	UU	110	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4445 (rusty.baldwin@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18