ABSTRACT

Title of dissertation: IP Geolocation in Metropolitan Areas

Author: Satinder Pal Singh

Advisor: Professor Mark Shayman
Department of Electrical and Computer Engineering
University of Maryland, College Park

In this thesis, we propose a robust methodology to geolocate a target IP Address in a metropolitan area. We model the problem as a Pattern Recognition problem and present algorithms that can extract patterns and match them for inferring the geographic location of target's IP Address.

The first algorithm is a relatively non-invasive method called Pattern Based Geolocation (PBG) which models the distribution of Round Trip Times (RTTs) to a target and matches them to that of the nearby landmarks to deduce the target's location. PBG builds Probability Mass Functions (PMFs) to model the distribution of RTTs. For comparing PMFs, we propose a novel 'Shifted Symmetrized Divergence' distance metric which is a modified form of Kullback-Leibler divergence. It is symmetric as well as invariant to shifts. PBG algorithm works in almost stealth mode and leaves almost undetectable signature in network traffic.

The second algorithm, Perturbation Augmented PBG (PAPBG), gives a higher resolution in the location estimate using additional perturbation traffic. The goal of this algorithm is to induce a stronger signature of background traffic in the vicinity

of the target, and then detect it in the RTT sequences collected. At the cost of being intrusive, this algorithm improves the resolution of PBG by approximately 20-40%.

We evaluate the performance of PBG and PAPBG on real data collected from 20 machines distributed over 700 square miles large Washington-Baltimore metropolitan area. We compare the performance of the proposed algorithms with existing measurement based geolocation techniques. Our experiments show that PBG shows marked improvements over current techniques and can geolocate a target IP address to within 2-4 miles of its actual location. And by sending an additional traffic in the network PAPBG improves the resolution to within 1-3 miles.

IP GEOLOCATION IN METROPOLITAN AREAS

by

Satinder Pal Singh

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Dr. Mark Shayman, Chair/Advisor
Dr. Richard J. La
Dr. Rama Chellappa
Dr. Ashok Agrawala
Dr. Jonathan Katz, Dean's Representative

Dedication

Dedicated to my beloved parents, and to my beautiful and loving wife.

Acknowledgements

I owe my gratitude to several people who made important contributions in making this thesis possible and enriching my graduate life experience.

First and foremost I would like to thank my advisor, Professor Mark Shayman, for giving me an opportunity to work with him on this exciting and challenging problem. He has been always there with help and advice whenever I needed. He encouraged me to take my own decisions and responsibilities during the course of this research. It has been a real pleasure and a great learning experience working with him.

I would also like to thank Professors Richard La and Bobby Bhattacharjee for their extraordinary help and guidance to my research during the last four and a half years. Thanks are also due to Professors Rama Chellappa, Jonathan Katz and Ashok Agrawala for serving on my committee, and for sparing their invaluable time to accommodate my thesis defense examination inspite of their busy schedules.

My fellow graduate students, Randy Baden and Choon Lee, have been of great help as well. Randy in particular deserves a special mention for developing and maintaining the measurement infrastructure needed for this research. But for his contributions, it was extremely difficult to collect data to develop algorithms for my research problem. I would also like to acknowledge help and support from all the volunteers who acted as landmarks for this project and allowed us to flood their home routers (intermittently!) for latency measurements.

My housemates and friends have been a crucial factor in ensuring a smooth

and enjoyable graduate life for me. I would like to express my gratitude to Kiran Kumar, Eduardo Arvelo, Siddhartha Jain, Vivek Srivastava, Saurabh Khandelwal, Mohit Thakral, Neha Gupta, Anuj Rawat and Masoumeh Haghpanahi for their friendship and support.

I would also like to acknowledge financial support from the Labortary for Telecommunications Sciences for this research.

I owe my deepest thanks to my parents who stood by me and encouraged me to pursue higher studies at USA, far away from home. The past few years have been really tough for them as they have been missing me every second. Words cannot express the gratitude I owe them.

Last but not the least, I thank my best friend and wife, Monica Syal. She has been an excellent companion for the last few years of my life, always encouraging me to work harder and reach for the stars. Her love and support have been, and will always be, a constant source of inspiration for me.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

a	Weight factor for PMF comparison
d_{KL}	Kullback-Leibler Divergence
d_l^p	Divergence between landmark l and target from probe node p
$\bar{d}_l$	Mean divergence between landmark l and target over all probe nodes
d_l^{max}	Maximum divergence between landmark l and target over all probe nodes
d_{SD}	Symmetrized Divergence
d_{SSD}	Shifted Symmetrized Divergence
$\mathcal{E}$	Mean error of a geolocation algorithm
l	Landmark
L^*	Location Estimate given by a geoloation algorithm
$\mathcal{L}$	Set of Landmarks
o	Order of an AR Model
p	Probe node
$\mathcal{P}$	Set of Probe nodes
ϕ	Penalty Function for computing d_{SSD}
ψ	Parameter for AR Model
ρ	Correlation Coefficient
s_{min}	shift that minimizes d_{SSD} between two PMFs
$\mathcal{V}$	Variance of error of a geolocation algorithm

AR	Auto-Regressive
CBG	Constraint Based Geolocation
DNS	Domain Name System
E-911	Emergency 911
FiOS	Fiber Optic Service
ICMP	Internet Control Message Protocol
IP	Internet Protocol
MPD	Mean Pairwise Distance
PAPBG	Perturbation Augmented Pattern Based Geolocation
PBG	Pattern Based Geolocation
PAPBGv1	PAPBG Version I
PBGv1	PBG Version I
PMF	Probability Mass Function
RTT	Round Trip Time
sPBG	smart PBG
sPAPBG	smart PAPBG
SVM	Support Vector Machines
WAN	Wide Area Network

Chapter 1

Introduction

Internet Protocol (IP) Geolocation algorithms map IP addresses to geographic locations. IP Geolocation aid several location aware services. Geolocation can be used for targeted advertising [17], efficient content distribution, and location-specific content customization [25]. The knowledge of an IP address's geographic location is critical for emergency services including E-911 for Voice-over-IP telephones; IP address locations are also increasingly being used as a tool for detecting online fraud and identity theft [10].

State-of-the-art IP geolocation techniques resolve addresses to approximately 30 miles [1, 2, 4, 5], roughly the diameter of a metropolitan area. This resolution is acceptable for some applications, e.g., content distribution, but is insufficient for others, in particular, location-based advertising and E-911.

In this thesis, we present two new approaches for finer resolution IP Geolocation. Our work departs from prior measurement-based geolocation approaches, all of which correlate latency with distance. However, techniques that rely on first-order statistics correlating latency and distance are impractical on a metropolitan area scale[1].

We model geolocation as a pattern recognition problem. Our algorithms iden-

[1]In particular, geolocating an address to within 10 miles based on propagation delay requires latency measurements with accuracy on the order of 100μseconds

tify and extract patterns from network statistics to geolocate an IP address. We propose a new Pattern Based Geolocation (PBG), which captures patterns in the distribution of latencies or Round Trip Times (RTTs) observed to a target. PBG models the signature of background traffic in the vicinity of the target and uses this 'signature' to geolocate the target to approximately 5 miles of its actual location. To further improve the resolution of PBG, we develop Perturbation Augumented PBG (PAPBG), which is inspired by Stochastic Resonance [7, 15]. PAPBG sends a small amount of signal traffic in the network to enhance the signature of background traffic. At the cost of sending an additional 600 Kbps aggregrate traffic to 20 nodes for approximately 2 minutes, PAPBG gives a higher resolution in the location estimate and geolocates the target to within 3 miles.

The rest of this thesis is organized as follows. In Chapter 2, we describe current geolocation approaches, and discuss reasons why they do not peform well within metropolitan areas. We discuss our performance evaluation measures and our measurement infrastructure in Chapter 3. In Chapter 4, we present our approach to this problem and describe some of the initial techniques we explored for geolocation. We describe our two algorithms in Chapters 5 and 6. We present results from experiments on this testbed in Chapter 7. In Chapter 8 we present initial versions of our algorithms and their performance. We describe avenues for future work in Chapter 9 and summarize the work in Chapter 10.

Chapter 2

Prior Work

Current techniques of geolocation can be classified into two major categories:- a) 'static techniques' that use passive approach to geolocate an IP address, and b) 'measurement based techniques' that use active measurements of network statistics. In this chapter we will give examples of techniques in each category, and their shortcomings when applied to a metropolitan area.

2.1 Static techniques

Static techniques either use a database or Domain Name Service (DNS) names of nearby routers to geolocate an IP address.

2.1.1 Database Lookup

A straightforward passive method of determining the geographic location of an IP Address is to use the public *whois* databases [3], which provide information about the registrant or assignee of an IP address block. However, the whois database information may be incomplete, obsolete, or inaccurate. Further, if a large block of IP addresses is allocated to a single entity, then the whois database does not provide information about the geographic location of individual IP addresses within that block [14]. There are a few geolocation approaches which use look-up from an

exhaustive tabulation between IP addresses and their exact locations [1, 2, 4, 5]. However, such databases are difficult to build and maintain. Since service providers regularly refresh IP addresses of their customers, these databases need to be updated frequently as well [4]. The location estimate obtained from these techniques gives a resolution of around 25 miles [4].

We present an example of the performance of one of these techniques (MaxMind [4]) on one of the IP addresses on Comcast network in our testbed. Table 2.1 shows the actual location of this IP address over a 7 week duration as well as its estimated location given by MaxMind during these times.

Table 2.1: IP Geolocation using MaxMind

	Actual Location	**Estimated Location**
Week 1	Greenbelt	Hyattsville
Week 3	Greenbelt	Hyattsville
Week 5	Germantown	Hyattsville
Week 7	Germantown	Hyattsville

As can be seen from the results, during Weeks 1 and 3 the IP address is located in Greenbelt while MaxMind gives an estimated location as nearby city of Hyattsville. After Week 5, the IP address is re-allotted to another landmark in Germantown while MaxMind database still shows its estimated location as Hyattsville. Note that if we were geolocating this IP address over entire US, then the resolution provided by MaxMind is acceptable as it geolocates the IP address to within 20

miles of its actual location. However, this resolution is not sufficient enough for geolocation in a metropolitan area.

2.1.2 DNS names

An alternate approach for geolocation is based on extracting geographic information from the DNS name of the end-host or a nearby router [22, 23]. Network operators often assign domain names to the network routers embedded with geographic codes. Extracting and identifying these geographic codes from a network router in the vicinity of the target can provide a useful estimate of its geographic location. However, this approach is not reliable since not all routers have desciriptive names. Moreover, since there is no standard for naming the routers, identifying this information can be a challenging task.

The following example illustrates the shortcoming of this technique when applied to a metropolitan area. We have a target IP address on Comcast network in Greenbelt, Maryland. When we run traceroute utility from one of our probe nodes from University of Maryland College Park (Qwest network), the closest router to the target that shows up is located in Lanham, Maryland. This is the gateway router between Comcast and Qwest networks in Washnington DC area. In fact using traceroute for any target IP address on Comcast network always shows the Lanham router as the nearest router to the target. No other router inside Comcast network is visibile using traceroute. Thus, we cannot follow this strategy for geolocation in a metroploitan area.

2.2 Measurement based techniques

Measurement based geolocation involves active measurements of RTTs to a target IP address from a machine at a known location. The Internet Control Message Protocol (ICMP) echo requests (pings) are used to collect RTT values between a pair of machines. These techniques assume that RTTs and distances between machines are correlated [16].

Delay based geolocation techniques use two sets of nodes: a) *probe nodes*, which initiate pings to the other nodes, and b) *landmark nodes*, which respond to pings sent by the probe nodes [27, 23]. Both the probe nodes and the landmark nodes have known locations. GeoPing [23] pings each landmark and the target from multiple probe nodes to create a delay vector for each of the landmarks and target. The delay vector consists of RTT values measured from each probe node. GeoPing compares the target's delay vector to those of all the landmarks using Euclidean distance, and the landmark which gives the smallest distance is the location estimate of the target. This method uses a finite number of locations and thus gives a discrete output. The resolution of this technique is of the order of 10^2 kilometers [23].

Another technique is Constraint Based Geolocation (CBG) [16]. Instead of mapping the target to one of the landmarks, CBG uses multilateration to combine delay values from multiple probe nodes to get a region for target's location. To estimate distance to the target from RTT values, each probe node pings the landmarks to get a 'latency map' of (distance, RTT) pairs [16]. Figure 2.1 shows a representative latency map constructed from data collected over a Wide Area Network (WAN)

by CBG [16]. It fits a 'bestline' on this data, which gives an upper bound on the distance, say r, of a target from the probe node with a given RTT value [23]. The target is assumed to lie inside a circle of radius r centered around this probe node. This circle forms one constraint. Each probe node constructs similar constraints (circular regions), and the intersection of these constraints gives an estimate of the region where the target is located (Figure 2.2). This technique geolocates a target to within 55 miles with 50% confidence. CBG modifications, in particular Topology Based Geolocation [18] and Octant [26], use additional constraints to refine the target location estimate within 22 miles with 50% confidence.

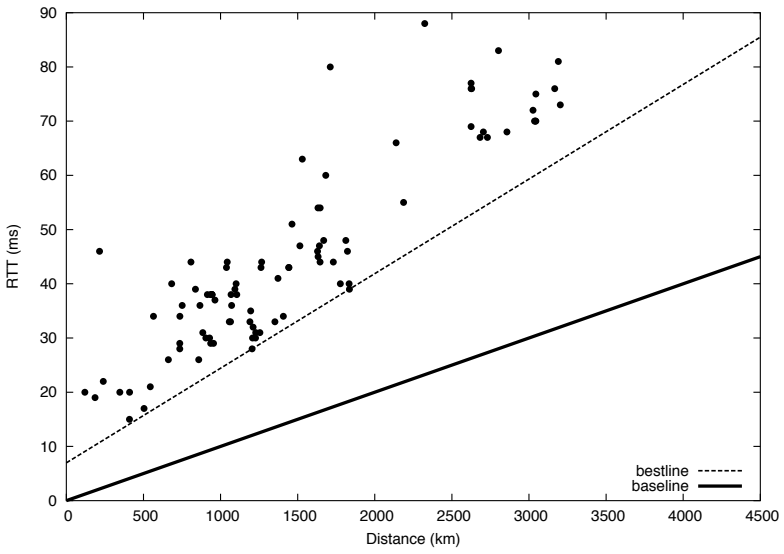


Figure 2.1: Representative latency map in a WAN[16] (included here with permission). RTTs are correlated with distance.

State-of-the-art delay based techniques can resolve the location of an IP ad-

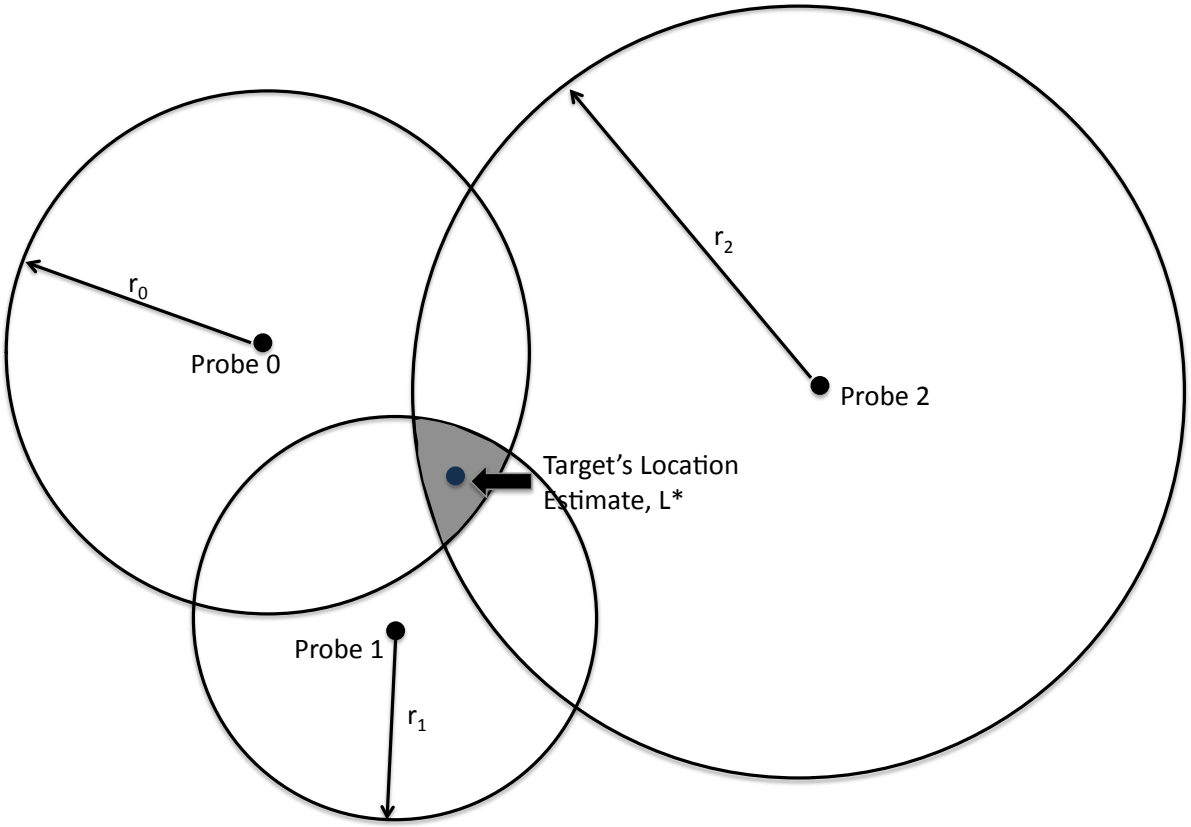
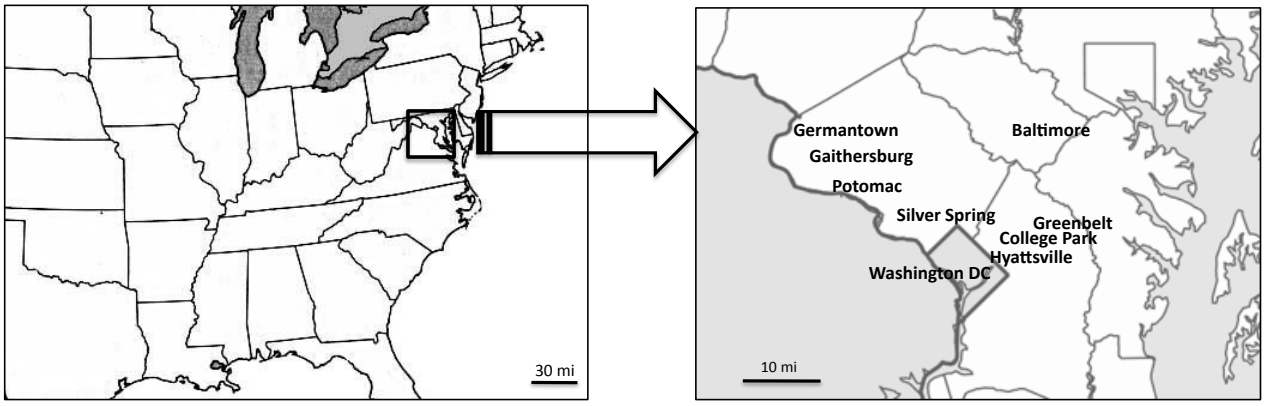


Figure 2.2: CBG in a WAN



Existing geolocation techniques on a WAN geolocate a target IP address to a 30 mile wide metropolitan area.

Our techniques geolocate an IP address inside a metropolitan area to within 5 miles.

Figure 2.3: Geolocation in a WAN and a metropolitan area

dress to roughly the size of a metropolitan area. The goal of our research is to develop algorithms that complement the existing measurement based techniques and provide a higher resolution geolocation estimate inside a metropolitan area (Figure 2.3). A possible solution to this problem is to directly use the existing geolocation techniques over a metropolitan area. Unfortunately this does not work as none of these delay based techniques has the capability to model the geolocation problem in a metropolitan area. Existing geolocation techniques use the correlation between distance and RTTs to geolocate a target. However, in a metropolitan area propagation delay is a small component of the RTT values, and the dominant component is queuing delay [8]. This violates the assumption of correlation between RTTs and distances between machines, and makes it difficult to use latency maps, since queuing delay is dynamic and needs to be modeled on the fly. Even if latency maps were constructed online to incorporate queueing delays, they are in-

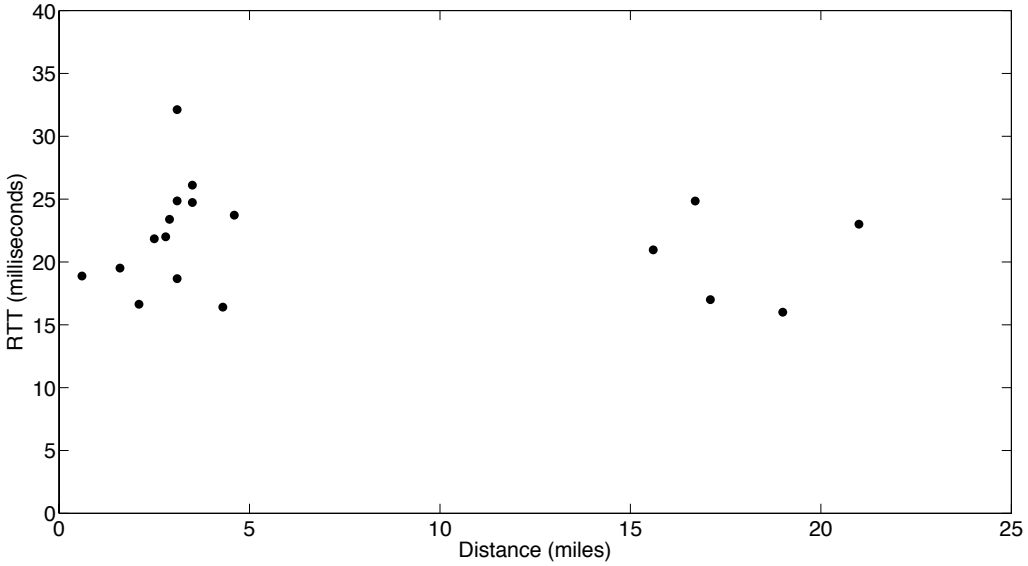


Figure 2.4: Representative latency map collected from landmarks in Baltimore-Washington DC metropolitan area. RTTs are not correlated with distance within a metropolitan area.

sufficient. Figure 2.4 shows a representative latency map collected from landmarks in Baltimore-Washington metropolitan area, which shows no correlation between latency and distance. The resulting upper bound estimate for the distance to the target from this latency map will be of the size of the metropolitan area itself, as we illustrate later in Chapter 7.

2.3 Challenges to geolocation in a metropolitan area

As discussed in the previous sections, the challenges to IP geolocation in a metropolitan area can be summarized as follows:

- The knowledge of an IP address alone is not sufficient enough to estimate the geographic location using database lookup.
- Extracting geographic information from domain names of the intermediate routers does not work.
- Measurement based approaches that correlate distance with latency fail in a metropolitan area.

In the next chapters of this thesis, we will discuss our measurement based approach for geolocation in a metropolitan area. In contrast to existing techniques we follow an alternate strategy and geolocate a target IP address by extracting patterns from its RTT sequences.

Chapter 3

Measurement Infrastructure

To develop and evaluate geolocation algorithms for a metropolitan area we needed real data. Since there is no public database of RTT measurements collected from machines in a metropolitan area, we deployed our own measurement infrastructure of more than 50 machines. Our infrastructure consists of a collection of probe and landmark nodes spread over 700 square miles in Baltimore-Washington metropolitan area. In this chapter we discuss our deployment to collect data over metropolitan area and the evaluation strategy for evaluating the performance of our geolocation algorithms.

3.1 Testbed

Our testbed consists of 3 probe nodes and 52 landmarks throughout the metropolitan area surrounding Washington, DC. We administer the probe nodes and are able to send active measurement packets from these nodes. We know probe node locations because we personally deploy the probe machines. For landmarks, we rely on volunteers entering their location information on a web form. We do not regulate the landmarks, and instead rely on their passive responses to ICMP echo request packets. Both probe nodes and landmarks have known geographical locations.

3.1.1 Probes

Our primary design constraint for the probe nodes was to be able to deploy as many of them as possible to cover a diverse set of vantage points. Our probe machines are Shuttle PCs running the 2.6.27-9 revision of the Linux kernel. Routers running the Linux kernel may be a viable and cheaper alternative, but in our testbed we found the extra memory and hard disk space to be useful for development.

Since a diverse set of vantage points includes homes, schools, etc., we deployed the probe nodes in both academic and home networks. These nodes were *inside* the home and academic firewalls, and thus these nodes had to be well secured. To achieve this, we use an *iptables* firewall that blocks all incoming traffic to the probe nodes except for ICMP echo response packets and packets in TCP streams that were initiated by the probe node. The probe nodes establish a reverse SSH tunnel to a central server, granting us remote access. The central server only allows remote login via an RSA key.

Our final requirement for the probe nodes is that they be able to send measurement packets as synchronously as possible from multiple probe nodes to multiple destinations. We realize this with two pieces of software; one coordinates an experiment with the available probe nodes from a central server, and the other sends ICMP echo request packets on a specific schedule as defined by the experiment script. Our software also collects a *tcpdump* of the relevant packets for post-mortem experiment analysis.

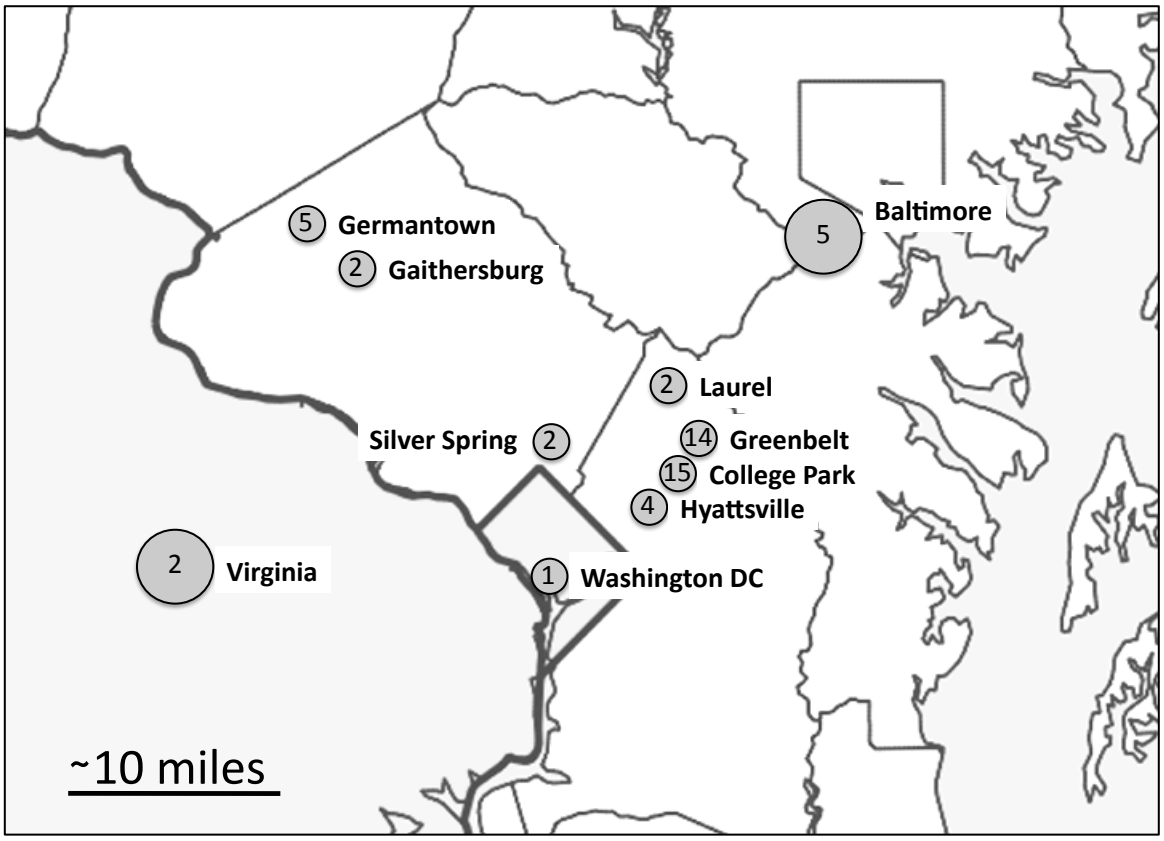


Figure 3.1: Landmark Locations in Baltimore-Washington metropolitan area. Each circle represents a different region, with the numbers representing the number of landmarks in the region and the diameter of the circle representing the size of the region.

3.1.2 Landmarks

Our landmarks are a mapping of IP addresses to geographical locations that provide a ground truth of accurately geolocated IP addresses. To collect and maintain this mapping, we enlist volunteers to provide their information on a web site. The volunteer only needs to enter his or her e-mail address and to identify his or her

location on a Google map, and we collect the volunteer's IP address automatically. Since a volunteer's IP address can change, each week we send an e-mail to volunteers with links for them to update their IP addresses and, if necessary, geographical locations. Figure 3.1 shows the number of landmarks in each city within our testbed. We do not display the exact location of our landmarks to preserve privacy of the landmarks.

3.2 Performance Evaluation

Our geolocation approach matches a target IP address to one of the landmarks in the testbed. Given a set of landmarks, the best possible estimate of a target's geographic location is the landmark which is geographically closest to it. Suppose s_{min} is the distance between the target and the geographically closest landmark. Let s^* be the distance between 'the best matching landmark' given by a geolocation algorithm and the target. Then the error of the location estimate of that geolocation algorithm is

$$\mathcal{E} = s^* - s_{min} \tag{3.1}$$

Here $\mathcal{E} \geq 0$, with equality when the 'best matching landmark' given by the geolocation algorithm is in fact the geographically closest landmark. As a strawman, selecting a random landmark as the location estimate of the target will give an error

$$\mathcal{E}_{random} = \bar{s} - s_{min}, \tag{3.2}$$

where $\bar{s}$ is the mean pairwise distance between the landmarks and the target. Random selection provides a baseline to which we compare our geolocation algorithms.

Chapter 4

IP Geolocation in a Metropolitan Area

Unlike existing measurement based techinques which infer a target's location using one RTT value, our method aims to identify, extract and match 'patterns' from RTT sequences instead. Thus, we model this problem as a *pattern recognition problem*. In this chapter we will present our approach to solve this problem and some of the initial temporal pattern matching techniques that we explored to extract patterns.

4.1 Geolocation Strategy

Our geolocation strategy involves geolocating a target IP address using two sets of nodes: probe nodes and landmark nodes. Figure 4.1 shows our deployment schematic. We know the exact locations of all landmarks (and the probe nodes). Probe nodes send synchronous probe packets (ICMP Echo Requests) to all landmarks and the target. The landmarks serve as location references and respond to probes sent by the probe nodes. We assume that the target responds to probes as well.

By sending back to back *synchronous* probe packets, the probe node measures synchronous RTT sequences for each landmark and the target. We match the target's sequence with those of all the landmarks. Our goal is to develop algorithms

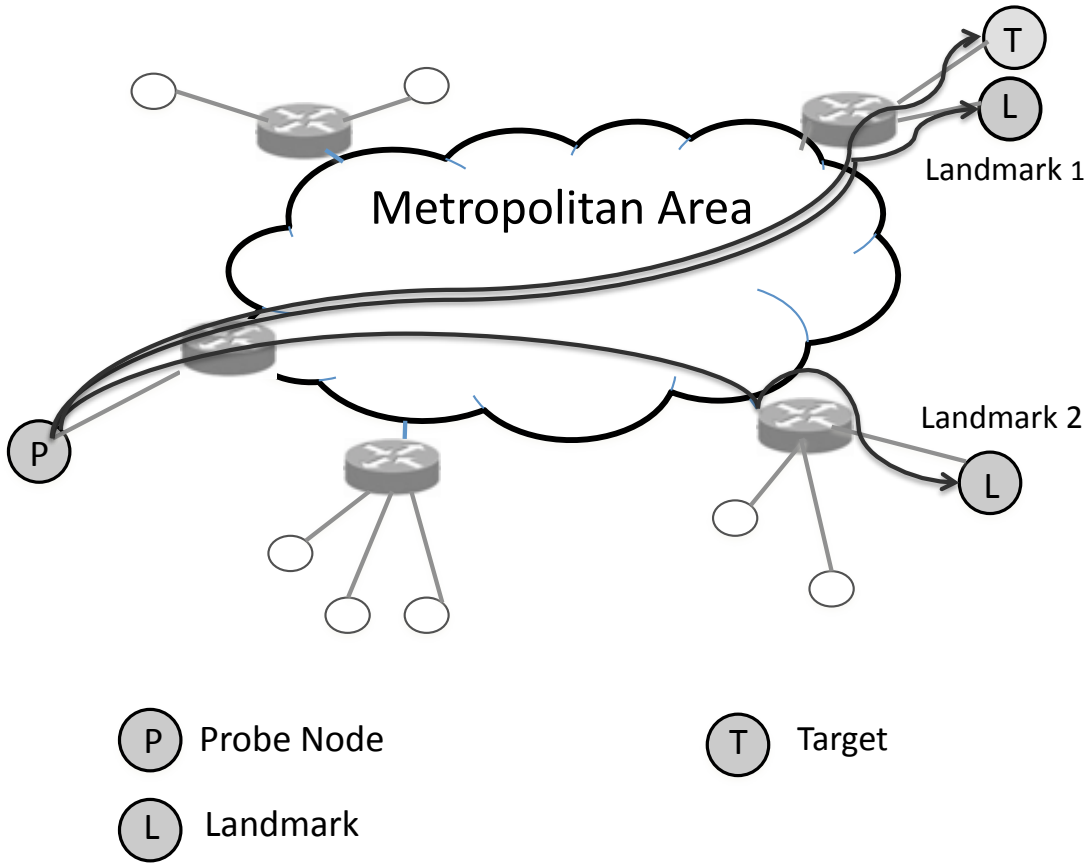


Figure 4.1: Geolocation Setup in a Metropolitan Area

which give the best match for the landmark which is in the vicinity of the target. Thus, the resolution of our approach is limited by the number and distribution of landmarks in the area. However, as we show later, using this approach, we are able to geolocate the target to within a few miles of its actual location.

4.2 Initial Approaches Explored

For extracting patterns from RTT sequences we initially explored standard temporal pattern matching techniques. In this section we discuss these techniques and present some our observations.

4.2.1 Correlation

Perhaps the most intuitive approach to match the RTT sequences is to use their cross-correlation coefficient [13]. Given an RTT sequence of the target $x_t, t = 1, 2, ..., N$ and that of a landmark $y_t, t = 1, 2, ..., N$, the cross-correlation coefficient, ρ, between the two is:

$$\rho = \frac{N \sum_t x_t y_t - \sum_t x_t \sum_t y_t}{\sqrt{N \sum_t x_t^2 - (\sum x_t)^2} \sqrt{N \sum_t y_t^2 - (\sum y_t)^2}} \tag{4.1}$$

If the two sequences are strongly correlated, then ρ will be approximately 1. If they are weakly correlated, then ρ will be close to 0. If the target shows a higher correlation with a nearby landmark than a landmark which is further away, we can use cross-correlation coefficient as the pattern for geolocation.

To test this approach, we collected multiple sets of RTT sequences from geographically close landmarks on the Comcast cable network in our testbed (Figure 2.3) and computed cross-correlation coefficients between them. Our experiments show that irrespective of how geographically close the two landmarks are, the cross-correlation coefficient obtained is consistently low. Figure 4.2 shows a sample cumulative density (cdf) plot of cross-correlation coefficients between synchronously

collected RTT sequences of two landmarks (less than 0.01 miles apart) in the city of Greenbelt, Maryland. We collected 100 sets of RTT sequences for the two landmarks at different times of the day using our probe node at Potomac on Verizon FiOS. Each sequence consists of 1000 RTT values collected from each landmark at a rate of 10 samples per second for 100 seconds. As the plot shows, the cross-correlation coefficient is very low and around 75% of the times it is less than 0.05. We show later in Chapter 7 that the mean error obtained with correlation based matching is in fact worse than random selection.

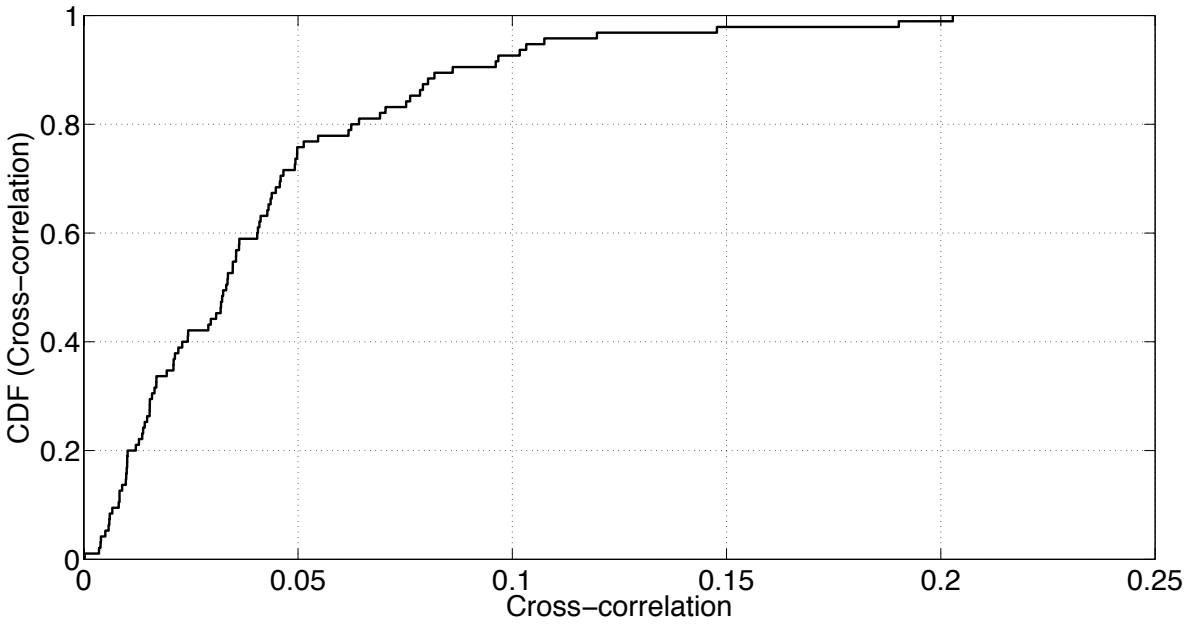


Figure 4.2: Representative CDF plot of cross correlation coefficient between RTT sequences of two geographically close landmarks in Greenbelt, Maryland.

In fact even the autocorrelation coefficient of the RTT sequence of a landmark

falls off rapidly with small shifts, as shown in Figure 4.3 for another landmark on the Comcast network. This shows that even with small shifts in measurements the RTT sequences observed to the same machine look uncorrelated.

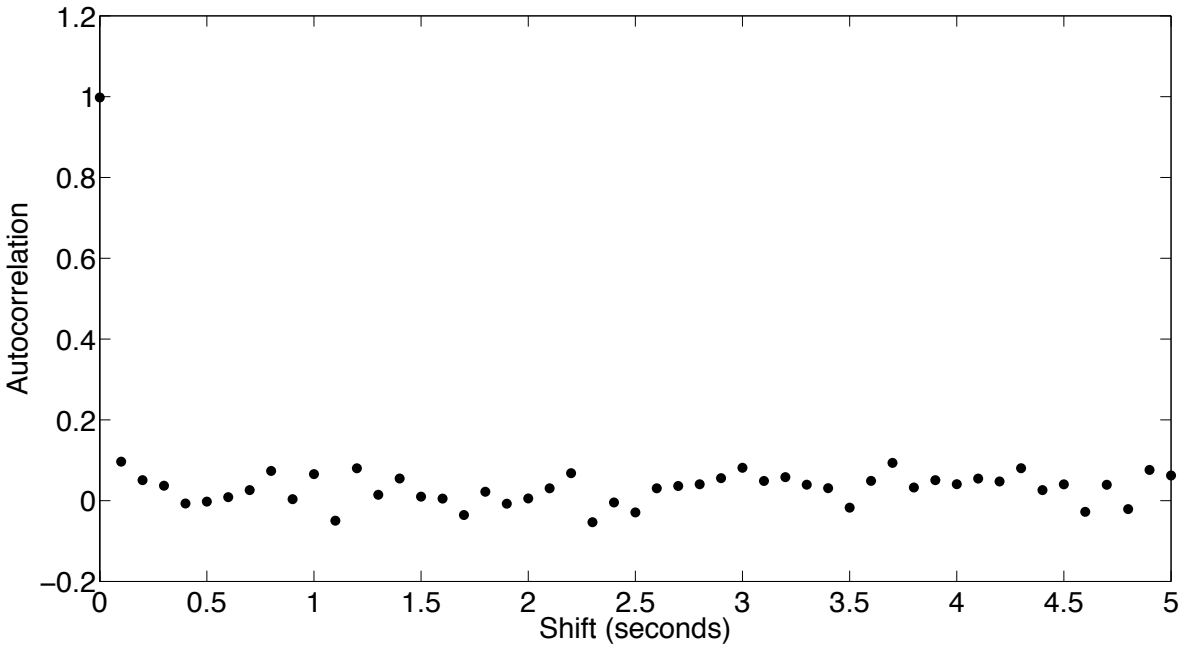


Figure 4.3: Representative plot of Autocorrelation of an RTT Sequence of a landmark in Greenbelt, Maryland.

4.2.2 Auto-Regressive Models

Auto-Regressive (AR) Models is a tool for modeling and predicting future values in a time series ([6]). Given a time series RTT sequence $x_i, t = 1 \ldots N$, an AR Model of order o is given by:

$$x_t = \sum_{i=1}^{o} \varphi_i x_{t-i} + \epsilon_t \tag{4.2}$$

where $\varphi_1, \ldots, \varphi_o$ are parameters of the model and ϵ_t is white noise.

The hypothesis for using AR Models was that using these models we can capture trends in the RTT sequences and match the trend in the target with those of the landmarks to get the landmark with the closest match. Since trends in RTT values are reflective of the traffic in the vicinity of the target, this should serve as a good estimate of the target's location. We followed two approaches of using AR Models for geolocation:

4.2.2.1 Approach 1

In this approach, AR models of order o were built to model the RTT sequences of the target and each landmark. Thus, we had a set of AR model parameters $\varphi_1, \ldots, \varphi_o$ for each of the machines. To get the best match, the model parameters of the target were compared to those of each landmark using Euclidean distance as a measure. The landmark with the least distance was chosen as the location estimate for the target.

4.2.2.2 Approach 2

In this approach, AR models of order o were built to model the RTT sequences of the landmarks alone. Each landmark's model was then used to predict the target's RTT sequence and the landmark whose model gave the minimum prediction error

was chosen as the output.

Different orders o were tried for both the approaches. However, I will show later in Chapter 7, the AR models failed to capture any trends in the RTT sequences and did not perform well. No significant 'temporal pattern' existed in the RTT sequences that could be captured.

4.2.3 Moving Averages

In this method, the RTT sequence of each landmark and the target was converted to a 'parameterized vector' using a moving average window. The parametrized vectors of the landmarks were compared to the target with Euclidean distance as the measure and the one with the least distance was assumed to be closest to the target. However, this approach suffers from the same problems as the other approaches mentioned above. In the absence of any significant temporal patterns, this method is essentially using only first order statistics of the RTT sequences which are not sufficient for geolocation in a metropolitan area.

In addition to the above mentioned patterns, we also evaluated the performance of an existing measurement based geolocation approach, CBG, and explored using 'mean' RTT value as a pattern for geolocation. None of these gave good results and in fact all these techniques performed worse than randomly selecting any one landmark as the location estimate for the target.

Chapter 5

Pattern Based Geolocation

Temporal pattern matching techniques were not useful in identifying patterns in the RTT sequences. Instead our Pattern Based Geolocation (PBG) approach considered the *distribution* of the RTT values. PBG assumes that RTT values are drawn from an underlying probability distribution function. In this chapter, we will present a detailed analysis of our PBG approach.

PBG consists of two steps: First, we construct Probability Mass Functions (PMFs) of RTTs for the target and the landmarks from the collected RTT sequences to model the distribution of RTTs. Next, we compare the PMFs of the landmarks to the PMF of the target. We output the landmark corresponding to the "best" match as the target's location estimate.

5.1 PMF Construction

To estimate the PMF of RTT values from a given RTT sequence, we use the 'k Nearest Neighbor' (kNN) density estimation method [13]. Given an RTT sequence $X_t, t = 1, \ldots, N$, the PMF value at a point x is given by:

$$p(x) = \frac{k/N}{V} \tag{5.1}$$

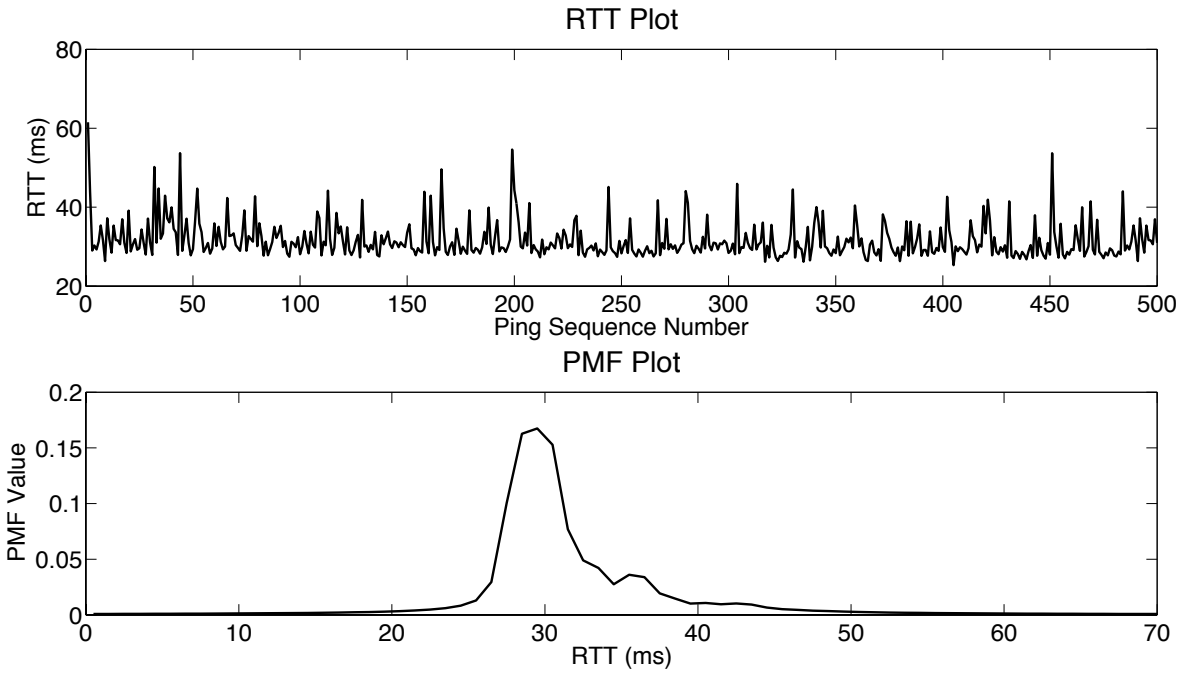


Figure 5.1: Plot of RTT Sequence and PMF of a landmark in College Park, measured from a probe node in Potomac

where V is the minimum volume of space centered around x which contains k nearest points from the sequence X_t. Since the RTT values are one dimensional, the volume V is the minimum one dimensional distance around x which contains k RTT values of the sequence. We choose $k = \sqrt{N}$, where N is the length of the RTT sequence.

We compute the PMF values at intervals of 1 millisecond. The range of the RTT values gives us the support of the PMF. For instance, if the RTT values lie in the interval of $(0, 100)$ milliseconds, we compute the PMF at 100 points at $0.5, 1.5, 2.5, \ldots, 99.5$ milliseconds. Figure 5.1 shows an RTT sequence and its associated PMF with support over $(0, 70)$ milliseconds.

5.1.1 PMF Parameters

There are two parameters involved in PMF construction:- Sampling Frequency of RTTs and Measurement Duration. Both these parameters play an important role in the performance of PBG. If RTTs are sampled at too low a frequency, we can miss capturing and detecting critical patterns in the PMFs. On the contrary if the sampling frequency is too high, the probe node will be sending too many packets per second resulting in congestion near the probe node. This can introduce artifacts in the RTT sequence which can result in mis-classification. Similarly, a much smaller observation duration means too few samples to estimate the PMF. The poorly estimated PMFs result in poor performance. On the other hand, a much longer observation duration is also not favorable, since the network statistics may change. In Appendix A we empirically obtain the 'best' values of sampling frequency and observation duration for our testbed by using a validation dataset. For now we will assume that the observation duration is 100 seconds and sampling frequency is 5 samples per second.

5.2 PMF Comparison

The PMF models the spatial distribution of the RTT values. The next step is to compare the PMFs, using a distance metric that can match their *shapes*. One way to compare PMFs is to use symmetrized Kullback-Leibler (KL) divergence [12, 20]. Given two PMFs $p(i)$ and $q(i), i = 1, \ldots, M$, the symmetrized KL divergence distance, d_{SD}, is given by:

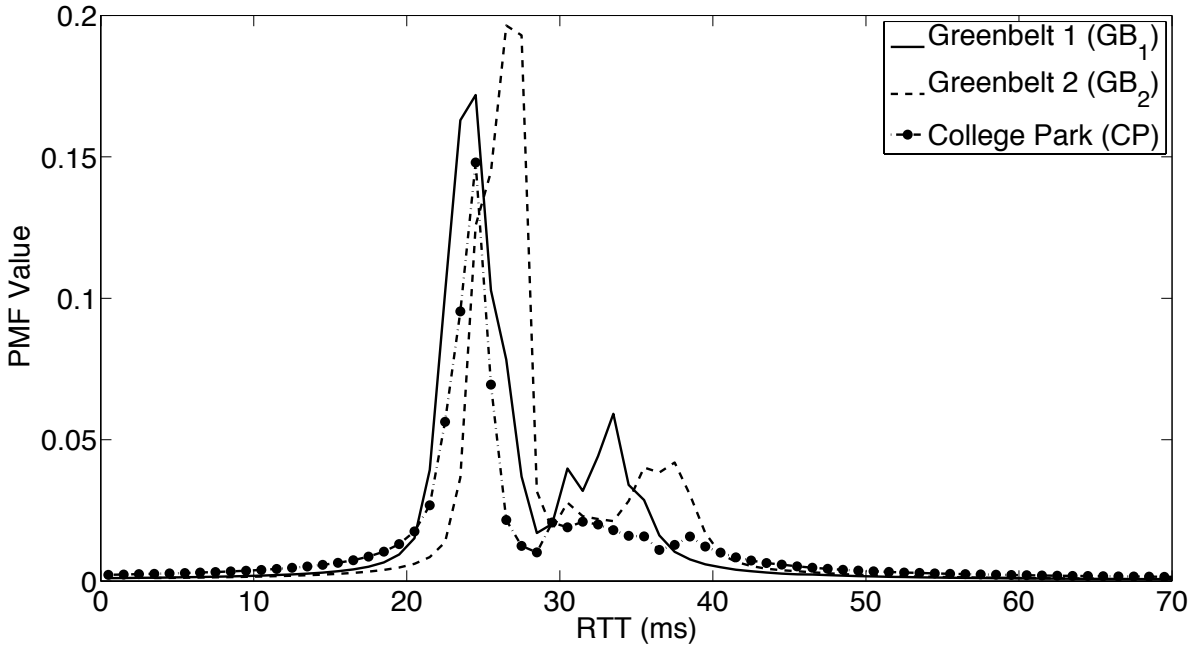


Figure 5.2: PMF Plots for machines in Greenbelt and College Park

$$
\begin{aligned}
d_{SD}(p\|q) &= d_{KL}(p\|q) + d_{KL}(q\|p) \qquad (5.2)\\
d_{KL}(p\|q) &= \sum_{i=1}^{M} p(i)\log_2\frac{p(i)}{q(i)}
\end{aligned}
$$

d_{SD} can be used as a distance metric. However, it has a critical drawback when applied to our problem, which is evident from the example shown in Figure 5.2. The figure shows the plot of PMFs for three machines: two machines in the city of Greenbelt (GB_1 and GB_2) and one in the city of College Park (CP) in Maryland.

As seen in this plot, the PMFs of GB_1 and GB_2 are similar in shape to each other but are shifted. This shift is the result of an extra hop in the $Probe - GB_2$ path as compared to the $Probe - GB_1$ path. The PMF of CP is aligned with that

of GB_2. The symmetrized divergence values for this set of machines are

$$\begin{aligned} d_{SD}(GB_2 \| GB_1) &= 1.4690, \text{ and} \\ d_{SD}(GB_2 \| CP) &= 0.9796. \end{aligned}$$

Due to the shift in the PMFs of GB_1 and GB_2, the d_{SD} distance fails to match them. To address this problem, we introduce a new distance metric called "Shifted Symmetrized Divergence" distance, (d_{SSD}), defined as:

$$\begin{aligned} d_{SSD}(p \| q) &= a \times \min_s \big(d_{SD}(p \| q_s)\big) + \\ & \qquad (1-a) \times \phi(s_{min}) \end{aligned} \tag{5.3}$$

Here

$$\begin{aligned} p, q &= \text{two PMFs} \\ q_s &= \text{PMF } q \text{ shifted by } s \\ s_{min} &= \arg\min_s \big(d_{SD}(p \| q_s)\big) \\ &= \text{shift (in milliseconds) that minimizes } d_{SD}(p \| q_s) \\ \phi &= \text{penalty function for shift} \\ a &= \text{weight} \end{aligned}$$

To compute d_{SSD} between two PMFs p and q, we first shift the PMF q to minimize the symmetrized divergence distance $d_{SD}(p \| q_s)$. This gives us the *divergence (or shape) distance*. The shift that minimizes the shape distance, s_{min}, is

then used to compute the *shift penalty distance* ($\phi(s_{min})$). It is important to add shift penalty as the shift contains information about the mean of RTT values, and adding this penalty avoids shifting the PMFs by arbitrary amounts. In Appendix B we empirically derive the best values of the weight parameter a and the penalty function expression ϕ. For now, we assume $a = 0.8$ and an exponential penalty function $\phi(s_{min}) = 2^{s_{min}}$. Using these values, the values of d_{SSD} obtained for the case mentioned in Figure 5.2 are:

$$\begin{aligned} d_{SSD}(GB_1 \| GB_2) &= 0.4513, \text{ and} \\ d_{SSD}(GB_2 \| CP) &= 0.7837. \end{aligned}$$

In addition to the above mentioned divergence based distance, we also explored *total variation* as a metric for PMF comparison. Given two pmfs, p and q, the total variation, V_q^p, between the two pmfs is given by:

$$V_q^p = \sum_{i=1}^{N} |p_i - q_i| \tag{5.4}$$

In Appendix C we compare the performance of shifted symmetrized divergence and total variation as distance metrics for PBG. Our experiments show that the divergence based metric shows much better performance. For the rest of the discussion, we will assume that PMFs are compared using shifted symmetrized divergence.

5.3 Multi-probe PBG

Our testbed consists of multiple probe nodes that try to geolocate a target IP address. Each probe node collects RTT sequences, and performs PMF calculations and comparisons to obtain divergence between the target's PMF and those of each landmark. To combine the results from different probe nodes, we have designed a multi-probe PBG method. We explored two decision rules for this as discussed below.

Suppose $\mathcal{P}$ denotes the set of probe nodes and $\mathcal{L}$ denotes the set of landmarks in our testbed. Each probe node $p \in \mathcal{P}$ computes the divergence, d_l^p, between the PMF of the target and PMF of each landmark $l \in \mathcal{L}$, given by

$$d_l^p = d_{SSD}(T^p \| l^p). \tag{5.5}$$

where, T^p and l^p are the PMFs of the target, T, and landmark, l, respectively, measured by probe node p.

5.3.1 Decision Rule 1 - Minimum Mean Divergence

The mean divergence, $\bar{d}_l$, between landmark l and the target T is given by

$$\bar{d}_l = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} d_l^p. \tag{5.6}$$

The location estimate of this rule is the landmark, L^*, given by

$$L^* = \arg\min_{l} \ \bar{d}_l. \tag{5.7}$$

Thus, this decision rule chooses the location estimate as the landmark with the 'minimum mean divergence' over all probe nodes.

5.3.2 Decision Rule 2 - Min Max

We also explored using a *minmax* rule for combining statistics from multiple probe nodes. In this case, we first compute the highest divergence, d_l^{max}, (worst case scenario) of each landmark l over all probe nodes as

$$d_l^{max} = \max_{p \in \mathcal{P}} d_l^p. \tag{5.8}$$

The final location estimate is the landmark, L^*, given by

$$L^* = \arg\min_{l} \; d_l^{max}. \tag{5.9}$$

The two decision rules use slightly different methods to combine results from multiple probe nodes. While the first uses average statistics from multiple probe nodes, the second finds the landmark with the best 'worst' performance. We analyse the performance of these two rules in Appendix D and show that the 'minimum mean divergence' gives better performance of the two. The results later presented in Section 7 are derived using this decision rule.

Multi-probe PBG combines statistics from all probe nodes to find the landmark which gives the best overall PMF match to the target. For the rest of this paper, we will simply use the term PBG instead of multi-probe PBG.

5.4 PMF variation with time

PBG uses PMF as a feature for geolocating a target IP address. To analyze the variation of PMFs of one machine with time of the day, we collected RTT sequences of 20 landmarks (12 on Comcast and 8 on Verizon) for one week from a probe node on UMD network. The RTT sequences were collected every half an hour. Each RTT sequence consisted of 500 RTT values collected at a rate of 5 packets per second for 100 seconds. We computed PMFs for different times of the day for each machine using these RTT sequences. We then compared the PMFs of one machine at different times of the day to study the variation of PMFs with time.

Figure 5.3 shows a representative plot of PMF variation of one landmark on Comcast network in Greenbelt, Maryland for one week. The X & Y axes denote the times of the week, and each point on this contour plot represents the divergence between the PMF collected from this landmark at two different times. As can be seen from this plot, the PMFs fall into two different 'types'. Type1 PMF is observed on most of the times of the day, other than 9am-3pm on weekdays when we observe Type2 PMF. The divergence values between PMFs of the same type remain mostly low (< 1), but across types the divergence values are much higher ($>> 1$).

Figure 5.4 shows a representative plot of a few PMFs from the two types from this landmark. As shown in this plot, the Type 1 PMFs which correspond to high traffic times of the day show a higher RTT value than Type 2 PMFs which correspond to work times during weekdays. This makes sense in the light of the fact that these landmarks are in fact the Wi-Fi routers placed at volunteers' homes.

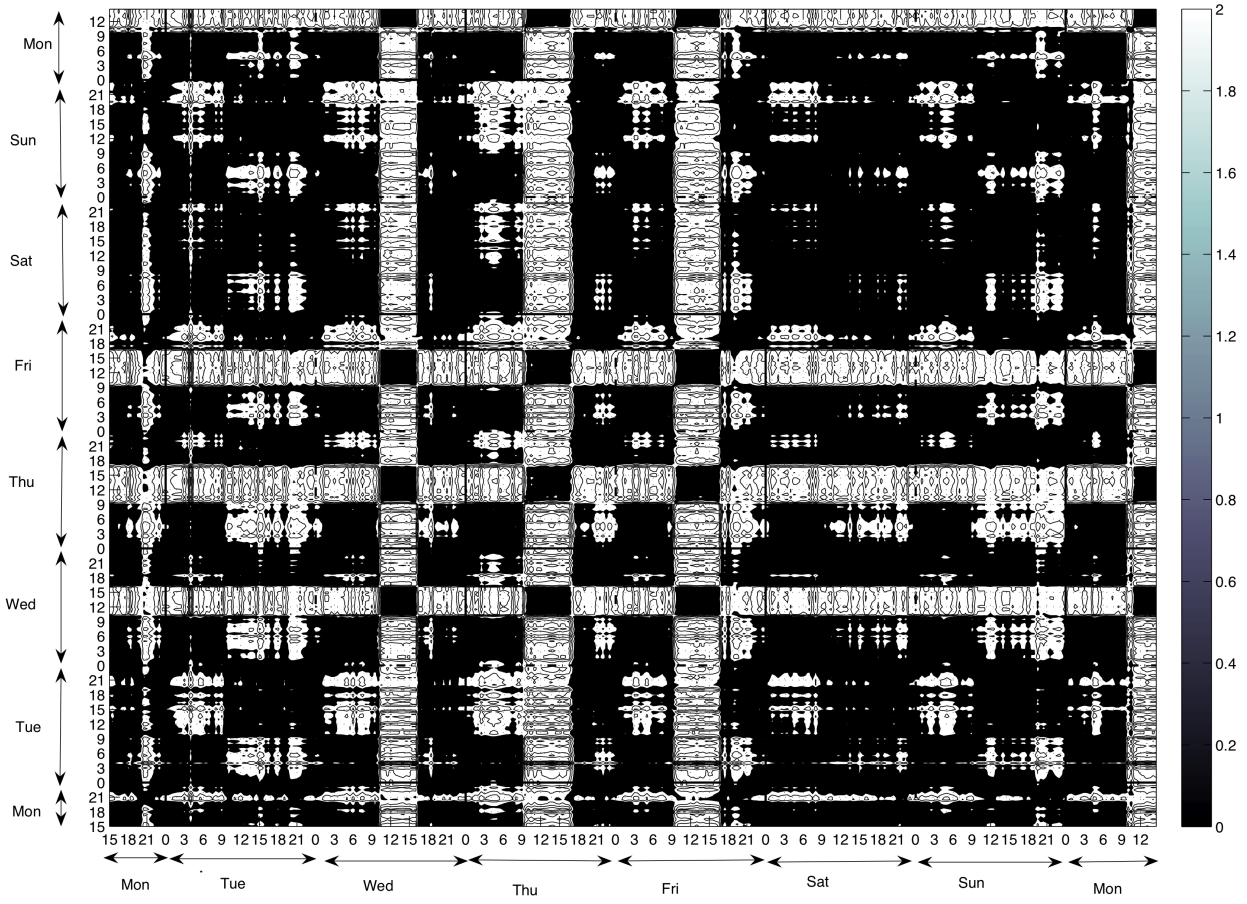


Figure 5.3: Representative contour plot of PMF variation between PMFs of one landmark on Comcast network in Greenbelt, MD

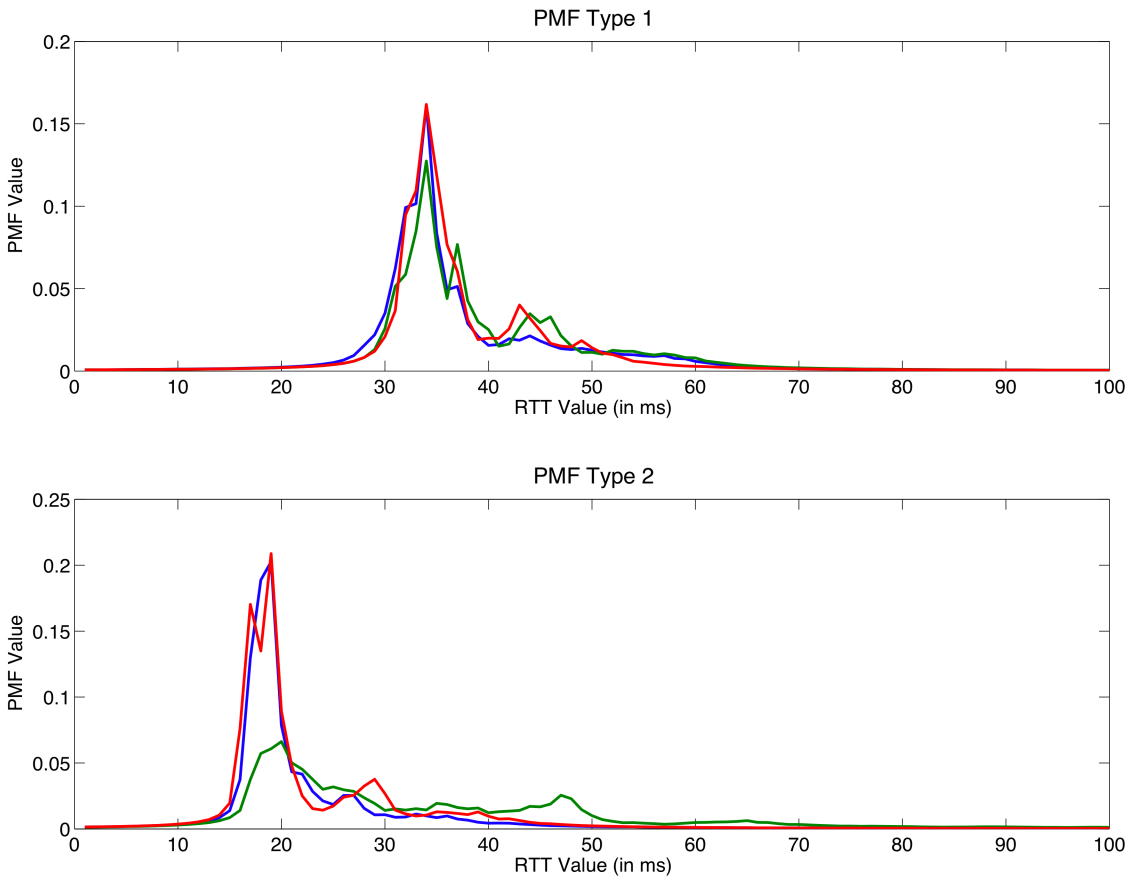


Figure 5.4: Representative plot of PMFs from the two types collected from one landmark on Comcast network in Greenbelt, MD

During regular working hours of weekdays, the traffic intensity at residential places is low, which gives rise to Type 2 PMFs.

These plots may suggest that instead of building PMFs for landmarks on the fly, we can pre-compute and characterize PMFs of the landmarks to build a unique 'PMF Bank' for each landmark. At the time of geolocating a target, we can observe only a few RTT values for each landmark to 'predict' the most suitable PMF for that landmark. This can reduce the amount of traffic sent in the network and help

scale PBG to a large number of landmarks. However simply constructing a 'static PMF bank' and picking one PMF from this bank does not work. Nevertheless, the study of variation of PMFs with time of the day does give useful insights into the way network traffic varies over different times of the day.

5.5 PBG Analysis

PBG tries to model the signature of background traffic using PMFs. For constructing the PMFs, the RTT sequences are obtained by sending ICMP echo requests to the landmarks and the target at a nominal rate (5 packets per second, 30 bytes per packet) for 100 seconds. The echo requests are sent synchronously to each landmark and target from each probe node. This amounts to a traffic of roughly 1.2 Kbps to each destination per probe node. Thus, for a 20 node testbed this approach involves sending 24 Kbps aggregrate traffic to the network per probe node for 100 seconds. As we show in Chapter 7, PBG can locate a target to within 5 miles in our testbed delpoyed in 700 square miles large Baltimore-Washington metropolitan area.

Chapter 6

Perturbation Augmented PBG

PBG relies on the background traffic in the vicinity of a target. It tries to capture a signature or a pattern of the background traffic using the RTT distribution of the target and match it to the nearby landmarks. However, in some instances, the background traffic signature is not strong enough, and PBG fails to map the target to geographically close landmark. Longer observation periods may help, though are not guaranteed to, develop a detectable and unique signature. Instead, we next describe an approach whereby we *enhance* the background traffic signature by introducing controlled amount of "perturbation" traffic into the network.

Perturbation Augmented PBG (PAPBG) is inspired by Stochastic Resonance [7, 15], where a small amount of stochastic input noise amplifies the feeble input information in a weak signal. PAPBG involves a new set of nodes called **perturbers**. These nodes send a low intensity signal traffic to all landmarks and the target, thereby increasing the background traffic. The technique works as follows. One of the probe nodes, acting as perturber, sends ICMP echo request packets (e.g. of size 100 bytes each) to all the landmarks and the target at a rate, say 50 packets per second. This corresponds to signal traffic of 40 Kbps to each landmark and target. The remaining probe nodes send small ICMP request packets (of size 30 bytes each) at a nominal rate of 5 packets per second for 100 seconds to measure

the RTT sequences. These probe nodes run PBG algorithms on the measured RTT sequences to give the best matching landmark. Thus, PAPBG is essentially PBG with an additional perturber which introduces a controlled amount of perturbation traffic in the network for better differentiation of PMFs.

6.1 Improved resolution with PAPBG

PAPBG can succeed where PBG fails. Consider a scenario with two landmarks in nearby cities, Greenbelt and College Park in Maryland, with a target in Greenbelt. Figure 6.1 shows the PMFs for the target, T, whose location we want to find, and the two landmarks in Greenbelt (GB) and College Park (CP). The distance between the target and the two landmarks is 0.8 miles (GB) and 3.2 miles (CP). In this case, the probe node sent synchronous probe packets at a rate of 5 packets per second to the two landmarks and the target for 100 seconds each.

Using PBG, with $a = 0.8$ and $\phi(s_{min}) = 2^{s_{min}}$, the divergence values obtained are

$$\begin{aligned} d_{SSD}(T\|GB) &= 1.09, \text{ and} \\ d_{SSD}(T\|CP) &= 1.06. \end{aligned}$$

In this case PBG fails to give the correct location estimate. In fact both the landmarks show similar values for the divergence, and it is not clear which landmark is truly closer to the target.

With PAPBG, the probe node sends probe packets to the target and landmarks

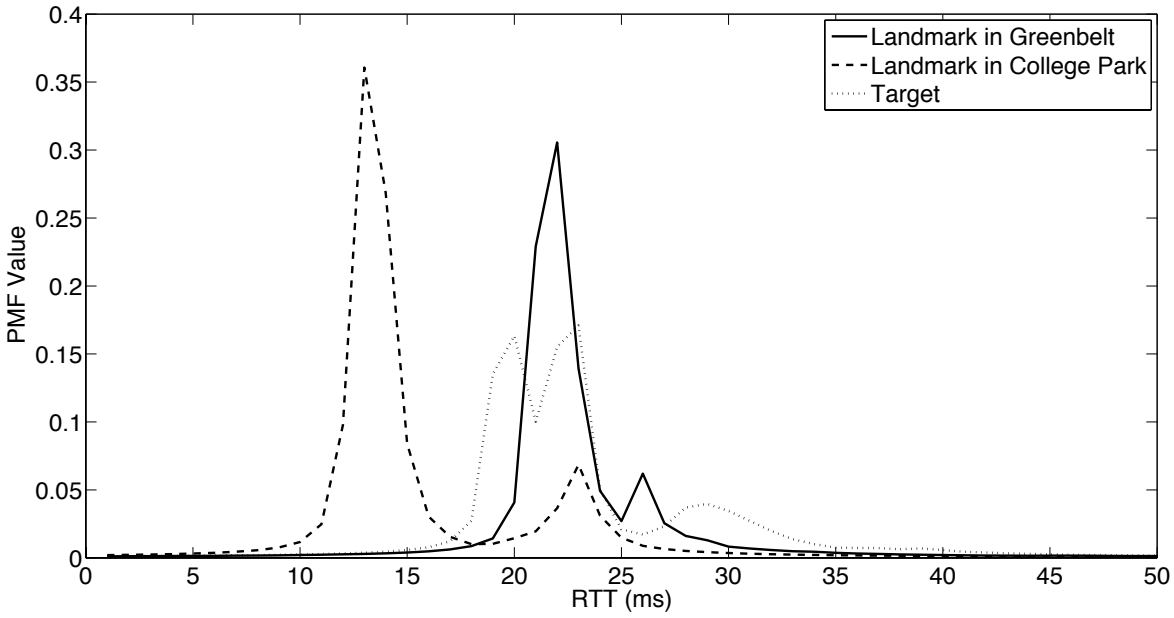


Figure 6.1: Plots of PMFs of the target and two landmarks with no perturbation signal

at the rate of 5 packets per second for 100 seconds. Simultaneously, the perturber sends signal traffic to the two landmarks and the target. The signal packets in this instance are 100 bytes each, sent at a rate of 50 packets per second resulting in a traffic intensity of 40 Kbps sent to each landmark and the target. The resultant PMFs constructed from the RTT sequences collected by the probe node are as shown in Figure 6.2. The new divergence values obtained are:

$$d_{SSD}(T\|GB) = 0.58, \text{ and}$$

$$d_{SSD}(T\|CP) = 1.14.$$

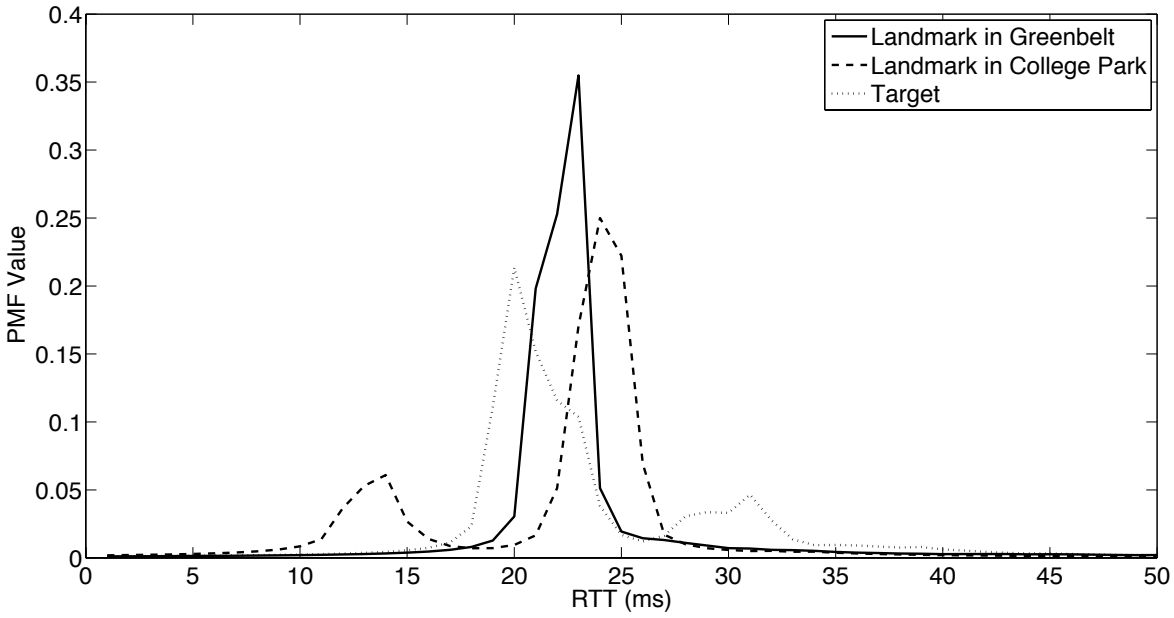


Figure 6.2: Plots of PMFs of the target and two landmarks with signal sent at 40 Kbps to the two landmarks and the target

In this case PAPBG is able to discriminate between neighboring cities and correctly geolocate the target to Greenbelt.

6.2 Perturbation Intensity

An important aspect of PAPBG is the intensity of peturbation signal. A too high intensity of perturbation signal can result in congestion near the perturber resulting in inefficient traffic injection in the network. Further a too high traffic can make the target aware that is being probed as well as may cause concerns of traffic disruption in the network. On the other hand a too low intensity may not be sufficient to induce a strong signature for detection via PMFs. In this section

we will discuss our experiments that we used to arrive at good working values of perturbation intensity.

Our goal is to enure that the overall traffic rate remains less than the bandwidth limits of the perturber and the landmarks (and the target). Our perturber (as discussed later in Section 7) is on Verizon FiOS with an upload bandwidth limit of 15 Mbps and download limit of 5 Mbps. Our landmarks on Comcast and Verizon have download/upload limits of 12/2 Mbps. We need to ensure that the overall perturbation traffic sent from a perturber to all landmarks (and target) remains less than 5 Mbps. At the same time perturbation intensity per destination node has to be less than 2 Mbps. Note that we quote these limits as our 'theoretical' upper bounds. In actual scenario we ensure that perturbation traffic intensity remains less than 10% of these limits at all times.

Perturbation intensity depends on two parameters: Packet frequency and packet size. Packet frequency introduces an additional constraint in the form of processing power of the routers. To explore the effect of packet frequency on network traffic we conducted the following experiments. We sent small probe packets (30 bytes each) at varying packet frequencies from the perturber node to one random landmark in our testbed. Different packet frequencies were explored - 125, 250, 500 & 1000 packets per second. We collected the RTT sequences from the perturber and analysed them for congestion and packet drops. Note that the four packet frequencies correspond to overall traffic intensities of 30, 60, 120 & 240 Kbps. Thus, we were under the maximum bandwidth limits for the perturber as well as the landmarks. The experiments were repeated for all landmarks. Figure 6.3 shows

a representative plot of RTT values for one landmark on Comcast network.

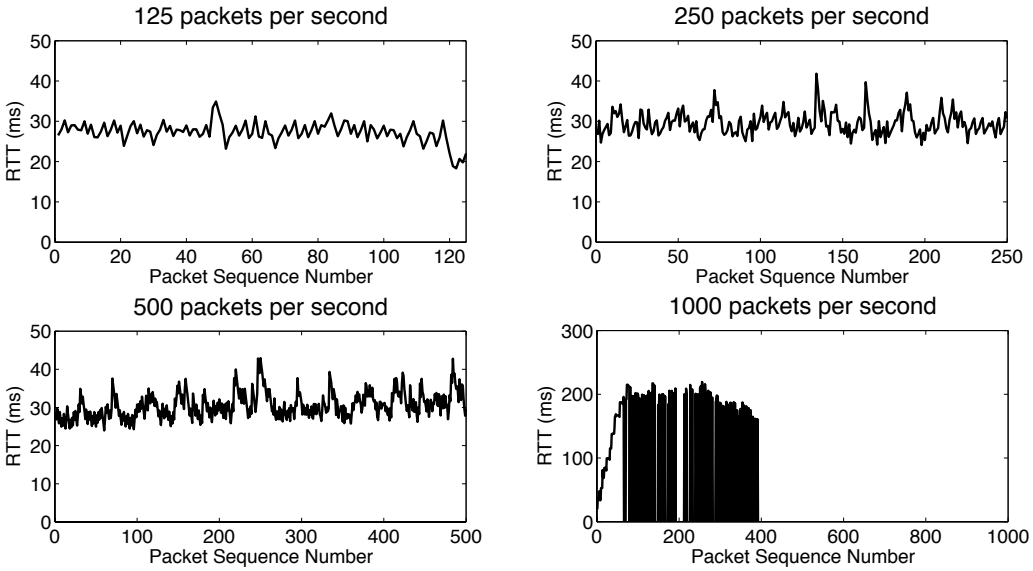


Figure 6.3: Plots of RTTs of a landmark observed from perturber for 125, 250, 500 & 1000 packets per second.

As can be seen from this figure, the RTT values remain more or less stable for packet intensities of 500 packets per second. However, if we go higher than this we observe congestion as well as packet drops in the network. Since the overall traffic intensity is less than the bandwidth limits of the perturber and the landmark, this effect can be attributed to the lack of processing power at one of the ends. To avoid this, we keep the maximum overall packet frequency from the perturber to close to 500 packets per second. This gives us the value of one of the parameters. As we will discuss later in Chapter 7, the packet size is adjusted so that PAPBG gives a good performance (in terms of mean error) and the overall traffic intensity remains

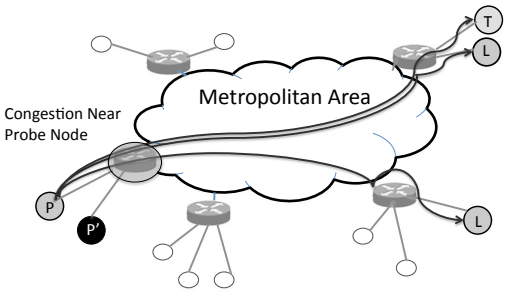
under the bandwidth constraints.

6.3 Perturber Placement

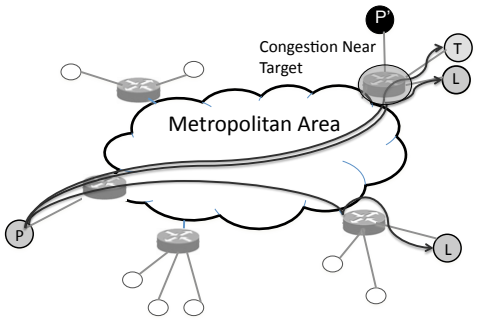
PAPBG gives a more accurate and finer estimate of the target's location when PBG fails. However, the setup for this technique has to be chosen carefully. Otherwise, we can end up with artifacts and false signatures in the RTT sequences. For example, if the perturber is too close to the probe node, then congestion is created near the probe node itself (Figure 6.4a) which can results in artifacts and misclassification at the probe node. Similarly if the perturber is too close to the target, then the target's traffic gets congested (Figure 6.4b). Therefore, the placement of perturber is critical for PAPBG. A desired setup in as shown in Figure 6.4c, where the paths from the perturber to the different landmarks' locations do not have much overlap.

6.4 PAPBG Analysis

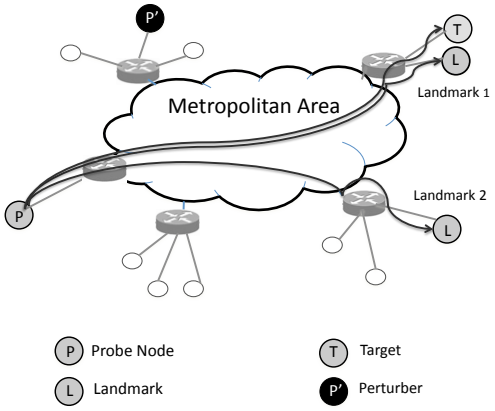
We show later in Section 7 that by sending an additional 600 Kbps traffic to 20 nodes for 100 seconds, PAPBG can geolocate a target IP address to within 3 miles on an average. Note that the traffic intensity per destination node is ≤ 50 Kbps, which is nominal compared to the bandwidth limits of these nodes (10 Mbps download and 2 Mbps upload). There is a tradeoff between accuracy and stealth. PAPBG is more intrusive than PBG. By sending extra traffic to the vicinity of the target, we leave a detectable traffic signature in the network which can make the



(a) Perturber too close to the probe node



(b) Perturber too close to the target



(c) Perturber in the desired locations

Figure 6.4: Perturber Placement

target aware that it is being probed.

Chapter 7

Experiments and Results

7.1 Data Collection

We collected data on our testbed to evaluate the performance of our algorithms. We used 12 landmarks on the Comcast network and 8 landmarks on the Verizon network for evaluation. The distribution of the landmarks in different cities for the two networks is shown in Tables 7.1 and 7.2. Table 7.3 shows the mean pairwise distance (in miles) between landmarks on the two networks. We used three probe nodes in our experiments. These probe nodes are located as shown in Table 8.3. Probes 0 and 1 acted as regular probe nodes for collecting RTT sequences, while Probe 2 acted as a perturber for PAPBG experiments.

Table 7.1: Landmark Locations on Comcast Cable Network

City	**# Landmarks**
Greenbelt	5
College Park	4
Germantown	2
Gaithersburg	1

Using the 2 probe nodes, we collected 50 sets of synchronous RTT sequences

Table 7.2: Landmark Locations on Verizon FiOS Network

City	**# Landmarks**
Severn	2
College Park	1
Millersville	1
Columbia	1
Vienna	1
Adelphi	1
Hyattsville	1

Table 7.3: Mean Pairwise Distance (miles) between landmarks on Comcast and Verizon

Network	**MPD**
Comcast	8.4
Verizon	11.8

Table 7.4: Probe Node Locations

Probe Node	**City**	**Network**
Probe 0	College Park	Qwest
Probe 1	Silver Spring	Verizon FiOS
Probe 2	Potomac	Verizon FiOS

from the 20 landmarks. In each set, every probe node synchronously sent probe packets to the 20 landmarks at a rate of 5 packets per second per landmark for 100 seconds. This generated a 500 sample RTT sequence for each landmark from every probe. The 2 probe nodes collected the 50 sets of data at different times of the day, with a random interval ($\in [30, 45]$ minutes) between consecutive data collections.

In this section, we first present results on this dataset using an existing measurement based geolocation technique, CBG [16], as well as explore mean RTT values and correlation between RTT sequences as the pattern for geolocation. We also compute error obtained with a Random Selection approach (see Equation 3.2). The best among these approaches gives us a baseline performance to compare PBG and PAPBG subsequently.

To evaluate the performance of these algorithms we use the *leave-one-out* [19] approach. We choose one landmark as the target and try to geolocate it with the rest of the landmarks in each dataset. We compute the mean error in geolocating this target over all 50 datasets. We iterate this procedure over all landmarks, with one landmark serving as the target in each iteration. And finally, we compute the mean error in geolocating the target over all iterations. Note that we evaluate all algorithms separately for the two networks, i.e., we use the landmarks on the Comcast network to geolocate a target on Comcast and the landmarks on the Verion network to geolocate a target on Verizon.

7.2 RTT Artifacts

Our algorithms are aimed at detecting and matching patterns in the RTT sequences. We assume that these patterns are introduced in the sequences due to the background network activity in the path to the landmark (or the target) from the probe node. However, if artifacts are introduced in the RTT values at or near the end nodes of this path, our algorithms pattern recognition algorithms will fail. The source of these artifacts can either be hardware at either end (i.e. Network Interface Card) or software (data collection software at the probe node or the OS of the Wi-Fi router at the landmark/target).

To study the effect of these 'end' sources on the RTT sequences we connected an off-the-shelf Linksys Wi-Fi router directly to our probe node and collected RTT sequences. Note that in our actual data collection, the probe node sends probe packets over ISP's network to a similar Wi-Fi router placed at the landmark's(target's) home. So this experiment setup is similar to an actual setup except for the intermediate network. We collected RTT sequences with two different probe traffic intensities:

- **Low Traffic:** 30 bytes probe packets at 5 packets per second
- **High Traffic:** 500 bytes probe packets at 250 packets per second

Table 7.5 lists the mean and variance of RTT values observed during these experiments. Note that the RTT values in both the experiments remain more or less constant with negligible variance. Compare this to RTT values collected in ac-

Table 7.5: RTT Artifacts

	Low Traffic	**High Traffic**
Mean (ms)	0.5	0.6
Variance (ms^2)	10^{-4}	10^{-4}

tual scenario with the probe node and the end Wi-Fi router spread across an actual network, where the variance in RTT values is of the order of tens of milliseconds. This proves that the end-host software and hardware do not introduce any appreciable or detectable artifacts in the RTT sequences and any patterns detected by our algorithms are a reflection of network traffic.

7.3 Baseline Performance

To get a baseline performance for comparing our algorithms against we explored the following techniques:

- CBG
- Mean RTT Value
- Correlation Coefficient
- Moving Averages
- AR Models

In this section we will present results obtained by all of these approaches.

7.3.1 CBG

We built latency maps [16] for each probe node using the mean RTT value for each landmark. Figure 2.4 shows a representative latency map obtained for one such dataset for Comcast network. Since there is no correlation between distance and latency, the maximum distance r_p obtained to the target from a probe node p is of the order of the size of the metropolitan area itself. Figure 7.1 shows the representative constraints and the location estimate given by CBG for a target in our testbed. Note that CBG gives a region as output and the centroid of this region serves as the location estimate for the target. The geographical distance between this estimate and the target's actual location is the error. In our testbed, the final region given by CBG was approximately the entire metropolitan area in all instances. Thus, the location estimate for CBG is the centriod of the metropolitan area. The mean error in geolocating the target using CBG, $\mathcal{E}_{CBG}$, on our testbed is 15.39 miles for the Comcast network and 18.06 miles for the Verizon network.

7.3.2 Mean RTT Value

We also explored mean RTT value as a pattern for geolocation. Since, mean matches the target's location to one of the landmarks, we evaluate the performance using error expression mentioned in Equation 3.1. We first compute the difference Δ_l^p between the mean RTT values of each landmark, l and the target from the RTT sequences collected by each probe node p. We then sum these over the two probe nodes to get an overall difference Δ_l for each landmark l and the target. The final

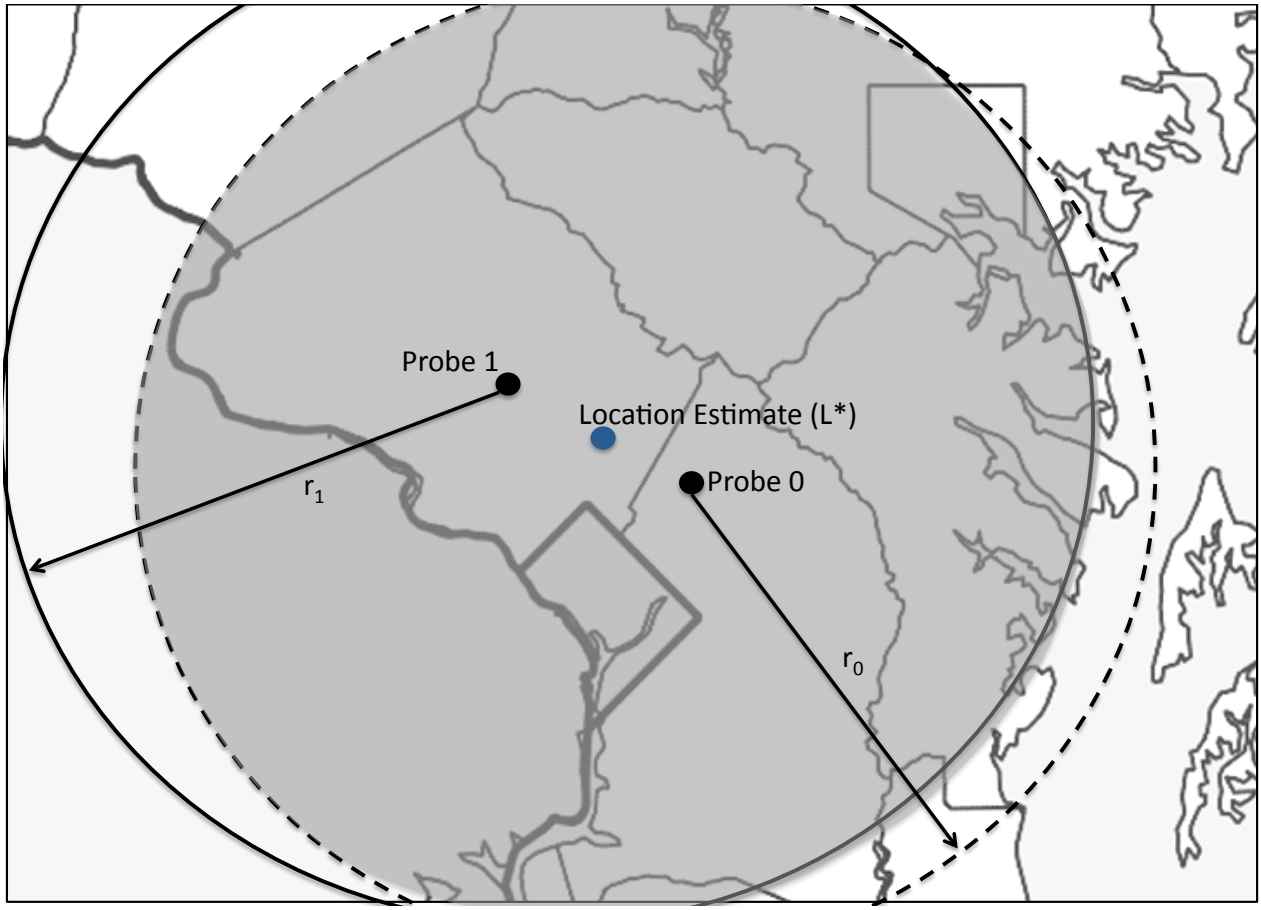


Figure 7.1: CBG on a metropolitan area

location estimate, L^*, is the landmark which has the minimum overall difference (Δ_l) in the mean RTT values to the target over the two probe nodes. The mean error obtained over all the targets using this approach, $\mathcal{E}_{mean}$, is 16.57 miles for the Comcast network and 14 miles for the Verizon network.

7.3.3 Correlation Coefficient

In case of correlation, we follow a similar strategy as used in case of mean. We compute correlation coefficient, ρ_l^p, between the RTT sequences of the target and landmark, l, obtained from each probe node. We then sum the statistics from the two probe nodes to get ρ_l for each landmark. The final location estimate, L^*, is the landmark which has the maximum ρ_l. The mean error obtained with correlation based matching, $\mathcal{E}_{correlation}$, is 13 miles for the Comcast network and 17.8 miles for the Verizon network.

7.3.4 AR Models

For AR Models we used both Approach 1 and Approach 2 to evaluate the performance. In case of Approach 1 we constructed AR models for all landmarks and the target from the RTT sequences collected from each probe node p. For each probe node p, we thus have a set of parameters φ_l^p for each landmark l and a set of parameters, φ_T^p for the target. We compute the Euclidean distance between the parameters of the target and that of each landmark, l, to get Δ_l^p for each probe node, p. The distances are then summed for each landmark over the two probe nodes to get Δ_l. The final location estimate, L^*, is the landmark with the minimum Δ_l.

For Approach 2, we build AR models for each landmark l from each probe node p to get a set of parameters φ_l^p. We use these models to predict the RTT sequence of the target. The prediction error obtained by a given model φ_l^p is e_l^p.

We combine the prediction errors for each landmark, l, over the two probe nodes to get e_l. The landmark l with the minimum overall prediction error is the location estimate of the target.

We explored different values of the order of AR models for the two appraoches. And the best performance was obtained with order 10 for Approach 2. The mean error, $\mathcal{E}_{AR}$, is 15.8 miles for Comcast network and 14.3 miles for the Verizon network.

7.3.5 Moving Averages

For the moving averages, we chose a window size of 5 samples with an overlap of 2 samples between subsequent windows. Since the sampling rate of RTT sequences is 5 samples per second, this corresponds to a window size of 1 second and overlap of approximately 0.5 seconds. We converted the RTT sequences collected from each probe node for all landmarks and the target to a 'parameterized vector' of means. We computed the Euclidean distance between the vectors of each landmark and the target from each probe node. And the landmark with the minimum overall Euclidean distance over the two probe nodes was chosen as the location estimate. The mean error, $\mathcal{E}_{moving\_averages}$, with this approach is 13.5 miles for Comcast network and 14.7 miles for Verizon network.

7.3.6 Random Selection

A random selection technique matches the target's location randomly to any one landmark in the testbed. The average distance between a target and its ge-

ographically closest landmark, s_{min}, is 0.67 miles and 3.04 miles for Comcast and Verizon networks respectively. The mean error in the location estimate obtained with the random selection approach, $\mathcal{E}_{random}$, (see Equation 3.2) is 7.62 miles for the Comcast and 8.76 miles for the Verizon network.

7.4 Summary

Table 7.6 summarizes the results obtained with all the above mentioned approaches. The errors obtained with CBG, mean and correlation based matching, and AR models are worse than random selection. Thus, random selection gives us the baseline performance for geolocation in a metropolitan area.

Table 7.6: Mean Error (miles)

Network	$\mathcal{E}_{CBG}$	$\mathcal{E}_{mean}$	$\mathcal{E}_{correlation}$	$\mathcal{E}_{AR}$	$\mathcal{E}_{moving\_averages}$	$\mathcal{E}_{random}$
Comcast	15.39	16.57	13	15.8	13.5	**7.62**
Verizon	18.06	14	17.8	14.3	14.7	**8.76**

7.5 PBG Performance

As discussed in Appendix B we chose an exponential penalty function, $\phi(s_{min}) = 2^{s_{min}}$, and $a = 0.9$ for the Comcast network and $a = 0.95$ for the Verizon network. We chose d_{SSD} as the distance metric for comparing PMFs (Appendix C) and used minimum mean divergence as the decision rule for multi-probe PBG (Appendix D). We followed the same *leave-one-out* approach to evaluate the performance of PBG

over the 50 sets of data. Table 8.4 shows the mean errors, $\mathcal{E}$, obtained for landmarks over the two networks. PBG can successfully geolocate a target with a mean error, $\mathcal{E}_{PBG}$, of 2.13 miles on Comcast network and 4.34 miles on Verizon network (Equation 3.1). The table also shows the variances in error, $\mathcal{V}$, obtained from PBG and random selection. These statistics are computed from errors in the location estimates obtained from all targets in 50 datasets. Compared to random selection PBG gives $\approx 75\%$ reduction in mean error and $\approx 50\%$ reduction in variance of error for targets on the Comcast network. For targets on Verizon network, the performance gain for PBG is $\approx 50\%$ in both mean and variance of error.

Table 7.7: Error Mean (miles) and Variance (miles^2) using PBG

Network	$\mathcal{E}_{PBG}$	$\mathcal{V}_{PBG}$	$\mathcal{E}_{random}$	$\mathcal{V}_{random}$
Comcast	2.13	41.09	7.62	98.91
Verizon	4.34	56.54	8.76	116.7

7.5.1 Matching Statistics

Table 7.8 shows the mean distance of the target to the top three nearest landmarks (L_1, L_2 and L_3 respectively) and the mean distance to the remaining landmarks (*Rest*) for the Comcast network. The table also shows the proportion of times the target is mapped to each landmark. The statistics presented are again average statistics obtained in multiple iterations, with each landmark serving as the target in each iteration. As can be seen, in majority of the cases, 51% of the times,

the target is mapped to the geographically closest landmarks.

Table 7.8: Target to Landmark Mapping for Comcast Network

	L_1	L_2	L_3	*Rest*
Distance (miles)	0.67	1.75	5.53	13.1
Match (Percent)	51	19	12	18

Table 7.9 shows the matching statistics for Verizon network. As can be seen from Tables 7.2 and 7.3, the landmarks on Verizon network are much more sparse. The resultant matching percentages show that even in this sparse distribution of landmarks PBG is able to geolocate the target to one of the closest landmarks 54% of the times. However, due to the large distance between the landmarks, the mean error is higher for the targets on Verizon.

Table 7.9: Target to Landmark Matching for Verizon Network

	L_1	L_2	L_3	*Rest*
Distance (miles)	3.04	5.8	6.92	18.7
Match (Percent)	54	16	11	19

7.5.2 PBG performance Versus Density of landmarks

To characterize the variation of PBG performance versus density of landmarks we iteratively drop landmarks from our testbed to simulate a sparse distribution of

landmarks. We start with PBG performance on 12 landmarks on Comcast network and 8 landmarks on Verizon network. This gives us a baseline performance.

Next, we randomly remove two landmarks from each network and recompute the mean error using PBG. The two landmarks are dropped in such a manner that increases the mean pairwise distance between landmarks. We repeat this step for all combinations of two landmarks that satisfy the above criterion. If by dropping two landmarks the mean pairwise distance decreases, then we consider that combination as invalid and choose a different pair of landmarks to be dropped. We compute the mean error for PBG over all possible and valid combinations of dropping two landmarks. This gives us the performance of PBG with a sparse distribution of landmarks.

In the next step we follow a similar strategy to drop four landmarks from each network with the validity criterion that mean pairwise distance between landmarks increases. We continue this strategy further and drop six landmarks from Comcast network to compute the performance of PBG. Note that we do not drop any more than four landmarks for Verizon network since, there are only 8 landmarks on this network. Dropping any number of landmarks more than 4 will be equivalent to trying to geolocate a target with only one landmark.

Figure 7.2 shows the variation of mean error of geolocating a target using PBG versus number of landmarks dropped for the two networks. As can be seen, with a sparser distribution of landmarks, PBG performance deteriorates and the mean error converges to mean pairwise distance between landmarks. Conversely with a denser distribution of landmarks PBG resolution will improve.

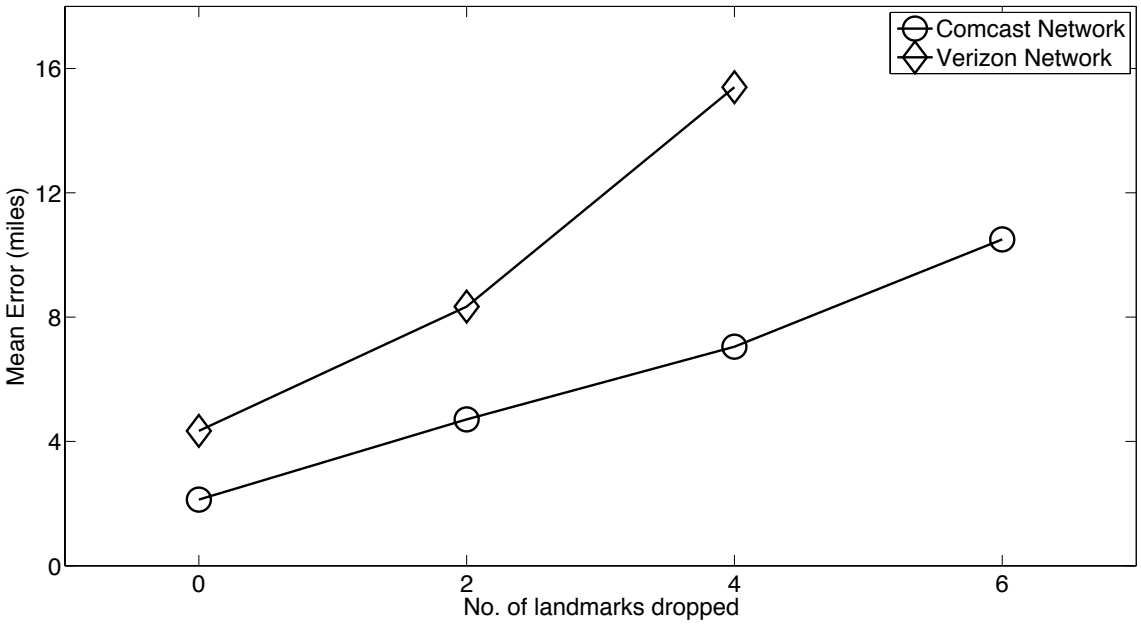


Figure 7.2: PBG Performance Vs Number of Landmarks

7.5.3 PBG with PMF bank

To study the feasibility of using a pre-collected PMF bank, we simulated a PMF bank using the following approach. As discussed in the previous sections, we collected 50 sets of data for evaluating PBG. For a given data set, we created a PMF bank for all landmarks using PMFs of these landmarks computed from the remaining 49 datasets. To choose the best suited PMF for each landmark from the PMF bank, we first computed the 'true' PMF of this landmark using the entire 500 RTT values from the present dataset. And then we selected the PMF from the PMF bank which gives the minimum divergence from the true PMF. This serves as the 'estimated' PMF for this landmark. Thus for each dataset, the PMFs obtained in the remaining 49 datasets were used for creating PMF bank for each landmark.

We geolocate the target using estimated PMFs of all landmarks. The mean error obtained with PMF bank based PBG is 9.2 miles for Comcast network and 9.6 miles for Verion network, which is worse than random selection. The results show that even though we chose the best possible matching PMF from a PMF bank, the minute differences between a true PMF and a representative PMF can degrade the performance of PBG to worse than random selection.

Note that in this experiment we used the entire 500 RTT values for each landmark to estimate the best PMF to confirm if this strategy is feasible or not. In actual scenario we would have hoped to used fewer values (e.g 50). But even after using all values, the estimated PMF performed worse than random selection. A simple 'static' PMF bank cannot be used for PBG.

7.5.4 PBG Costs

To geolocate a target PBG measures RTT sequences of the landmarks and the target for 100 seconds sampled at a rate of 5 RTT values per second from two probe nodes. At the cost of sending approximately 50 Kbps traffic (corresponding to the ICMP echo request packets) over 20 nodes, PBG can geolocate the target to approximately 5 miles within 100 seconds. Note here that the time taken for PBG to geolocate the target predominantly consists of time taken to measure the RTT sequences; the computation time for PMF calutations and comparisons is negligible. A higher density of landmarks can increase the resolution of PBG further. However, as we demonstrate in the next subsection, under certain constraints it is possible to

increase the resolution without adding more landmarks by using PAPBG.

7.6 PAPBG Performance

To evaluate PAPBG, we chose 5 intensities of perturbation : 10, 20, 30, 40 and 50 Kbps per destination node (landmarks and target). Our landmarks on Comcast and Verizon networks have a download bandwidth of 10 Mbps and upload bandwidth of 2 Mbps. We send signal at a maximum intensity rate of 50 Kbps to each landmark and target, which is lower than the upload bandwidth of these nodes. Probe 2 acted as perturber, while Probes 0 and 1 acted as regular probe nodes. As discussed in Chapter 6 we wanted to keep the aggregrate packet frequency at the perturbed around 500 packets per seconds. So we chose a traffic frequency of 45 packets per second per destination node. The experiments for Comcast and Verizon network were conducted at different times. So at any given time the maximum packet frequency from the perturber was 540 packets per second. With this packet frequncy we sent packets with varying packet sizes (30, 60, 85, 110 and 140 bytes) for the above mentioned peturbation intensities.

We collected 50 sets of RTT data (in addition to the datasets used for evaluating PBG) for each signal intensity at different times of the day, with a random interval ($\in [30, 45]$ minutes) between subsequent data collections. Using the same *leave-one-out-appraoch* we evaluated the performance of PAPBG on the 50 sets of data for each signal intensity. The parameter values used for PMF comparisons were the same for the two networks as for the PBG. Figures 7.3 and 7.4 show the

mean and variance of errors in the location estimate versus signal intensity for the Comcast network. The statistics presented are average statistics computed for all targets in the 50 datasets. Further signal intensity of zero corresponds to the PBG dataset collected without any perturbation.

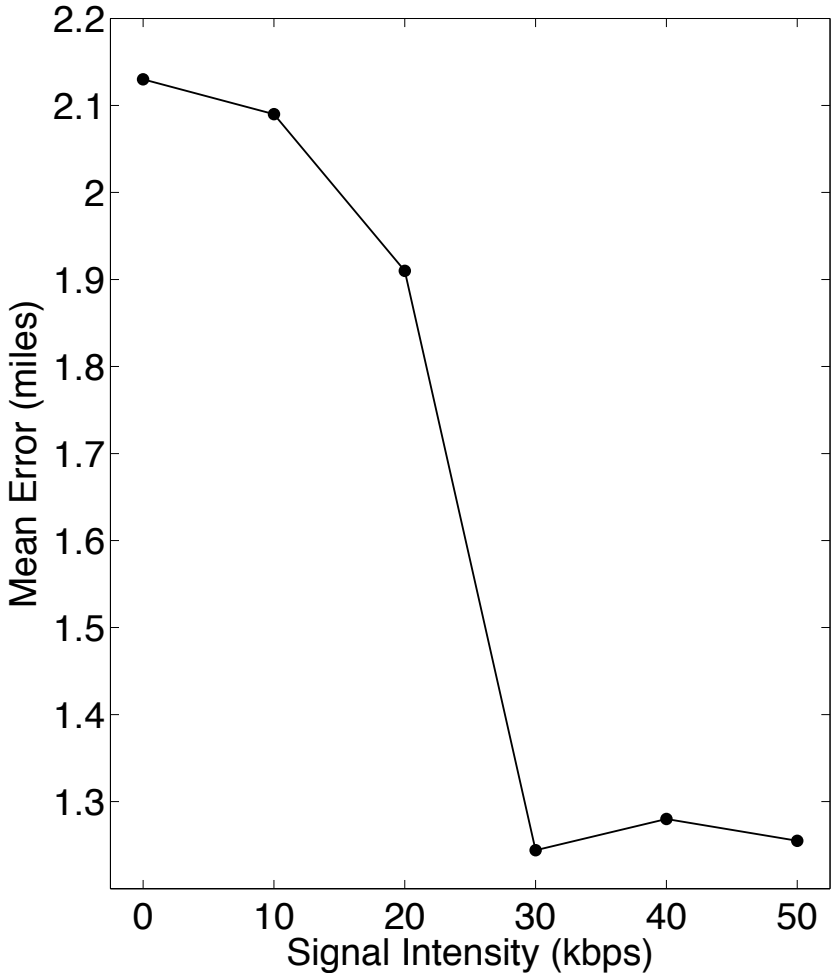


Figure 7.3: Error Mean vs Signal Intensity (Kbps) per node for Comcast

For the Comcast network PAPBG shows an improvement in the mean and variance of error in target's location with an increase in the signal intensity. The

Table 7.10: Target to Landmark Matching vs Signal Intensity (Kbps) per node for Comcast Network

Landmarks	**Distance (miles)**	**Noise Intensity (Kbps)**					
		0	10	20	30	40	50
L_1	0.67	51	55	59	60	61	60
L_2	1.75	19	18	19	15	10	11
L_3	5.53	12	13	9	11	14	15
Rest	13.1	18	15	13	14	15	14

Table 7.11: Target to Landmark Matching vs Signal Intensity (Kbps) per node for Verizon Network

Landmarks	**Distance (miles)**	**Noise Intensity (Kbps)**					
		0	10	20	30	40	50
L_1	3.04	54	58	60	61	59	60
L_2	5.8	16	14	11	10	15	13
L_3	6.92	11	10	12	14	11	10
Rest	18.7	19	18	17	15	15	17

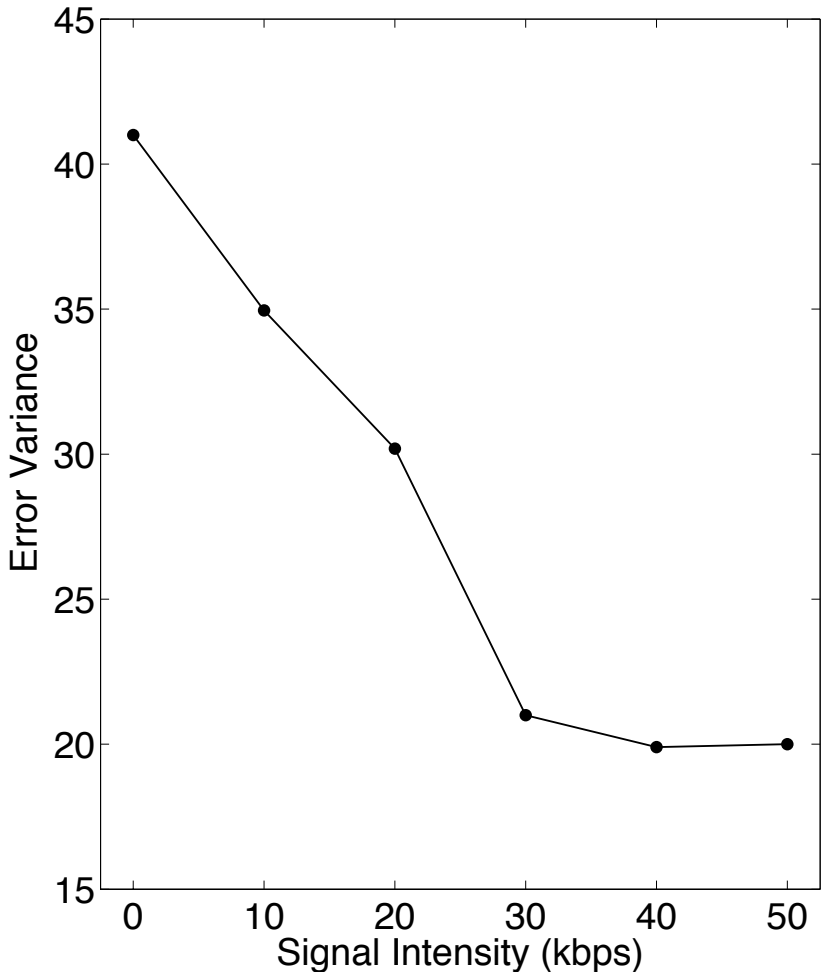


Figure 7.4: Error Varaince vs Signal Intensity (Kbps) per node for Comcast

best performance is obtained with signal sent at 30 Kbps to each landmark and the target. PAPBG reduces the mean error to 1.4 miles and variance of error to 20 miles^2. This is a gain of 40% compared to PBG and to 80% compared to random selection.

To explain the performance gain for PAPBG on Comcast network we show

an example. We choose one landmark as target on Comcast network and use the remaining 11 landmarks to geolocate it. We compute the average divergence values between the PMFs of the landmarks and that of the target from the two probe nodes and obtained $\bar{d}_l$ for each landmark l (see Equation 5.6). Using these divergence values we construct a 'divergence map' for this target, which is a scatter plot of (distance, divergence) values for the 11 landmarks with respect to the target. Each point $(s_l, \bar{d}_l)$ on this plot represents the distance s_l and divergence $\bar{d}_l$ between the target and the landmark l. Figure 7.5 shows representative 'divergence maps' for a target on Comcast network obtained from PBG and PAPBG datasets. From PBG divergence map, we can see distant landmarks show lower divergences compared to nearby landmarks, and, are thus, sources of high error in the location estimate of this target. The best matching landmark from PBG (marked L^*) is at a distance $s^* = 22$ miles from the target. Consequently, for this target the PBG error $\mathcal{E}_{PBG} \approx 20$ miles.

Now compare this to the divergence map obtained for the same target from PAPBG dataset with signal intensity of 30 Kbps to each landmark and the target (Figure 7.5). The distant landmarks show an increase in the divergence values, while the divergence of the nearby landmarks decreases. The best matching landmark, L^*, in this case is at a distance $s^* = 2.5$ miles. Hence, $\mathcal{E}_{PAPBG} \approx 0$. Thus, PAPBG helps in differentiating the PMFs of nearby landmarks from the distant ones.

Table 7.10 shows the target to landmark matching statistics for Comcast network for various signal intensities (in Kbps) per node. The first column of signal intensity "0" corresponds to PBG dataset with no signal injected in the network. As shown in this table, with a small amount of signal sent to the network, PAPBG

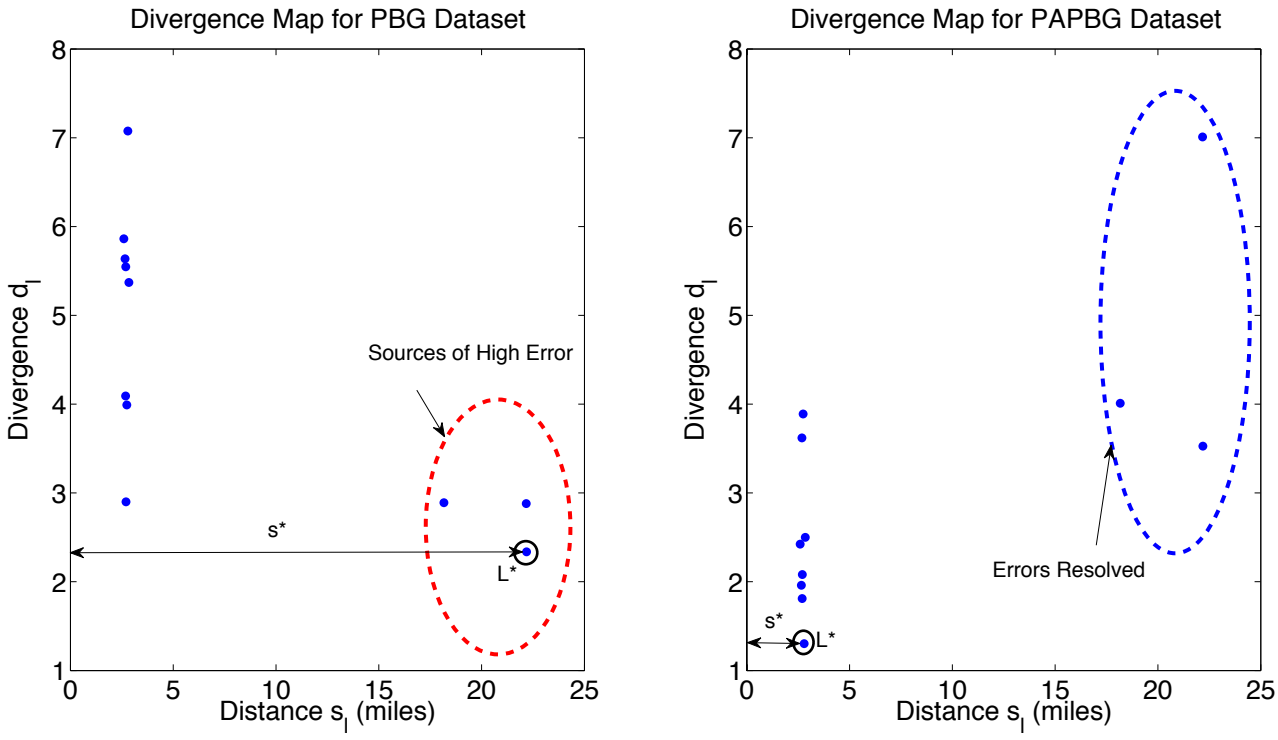


Figure 7.5: Representative Divergence Maps for a target on Comcast network from PBG and PAPBG datasets

matches the target to the geographically close landmarks more frequently. Consequently this results in a decrease in the mean error. Thus, at the cost of sending 360 Kbps extra traffic to 12 nodes on Comcast network (30 Kbps per node), PAPBG improves the resolution of geolocation search on Comcast network. However, after the signal intensity reaches 30 Kbps, we enter a region of diminishing returns. No significant gains are achieved after this point, and the performance is now limited by the distribution of landmarks.

Figures 7.6 and 7.7 show the variation of mean and variance of error with signal

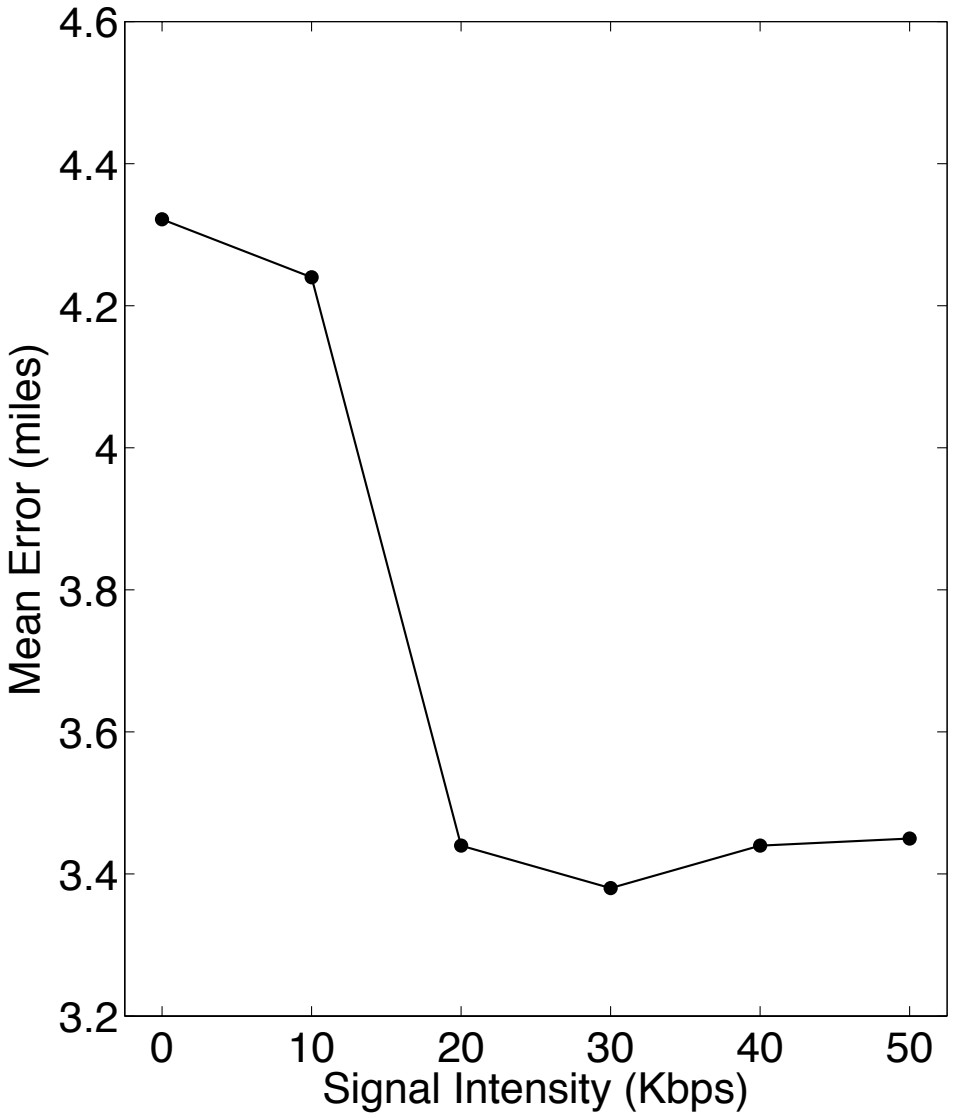


Figure 7.6: Error Mean vs Signal Intensity (Kbps) per node for Verizon

intensity per destination node for Verizon network. With an added perturbation, PAPBG improves the resolution of geolocation search in this case as well. The best performance is achieved for signal intensity of 30 Kbps per destination. The mean error goes down to 3.4 miles, an improvement of approximately 20% over PBG

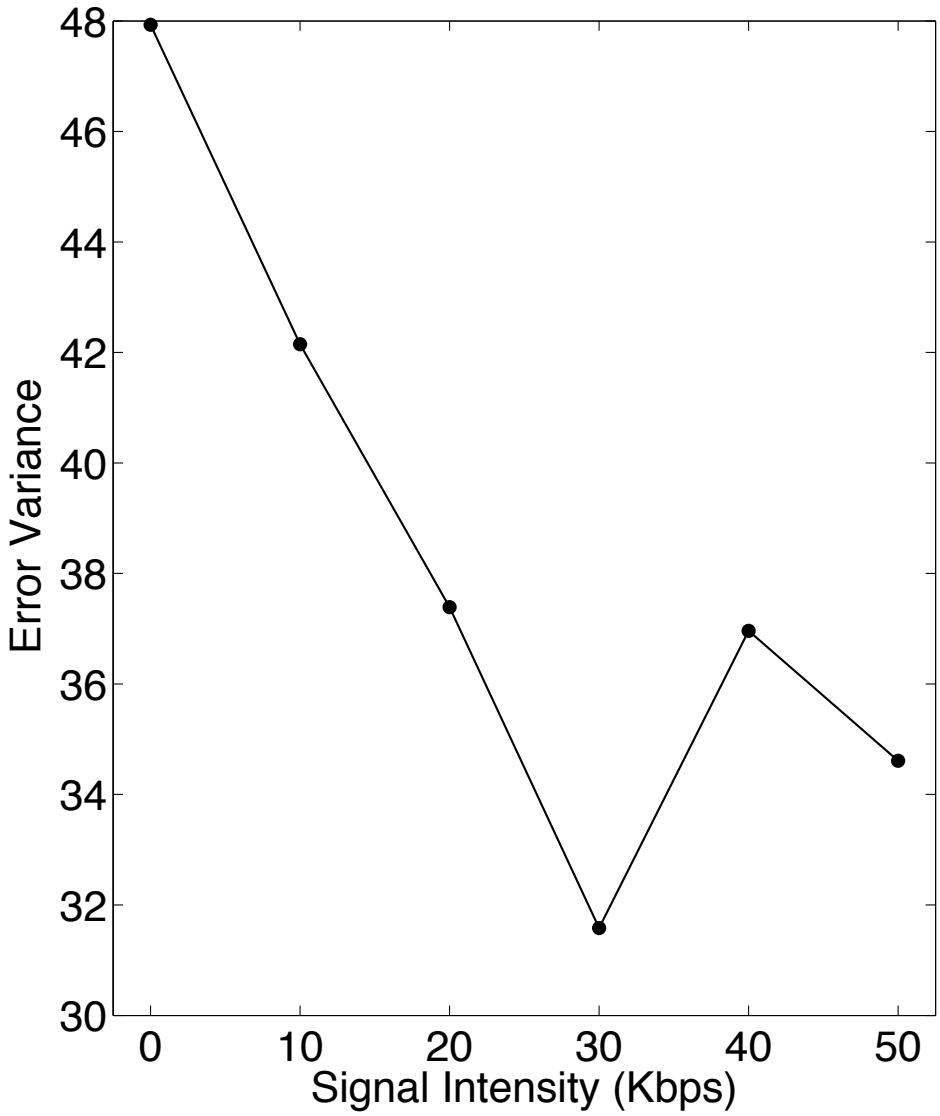


Figure 7.7: Error Variance vs Signal Intensity (Kbps) per node for Verizon

and approximately 60% over random selection. The variance of error also goes to approximately 32, an improvement of 40% on PBG and 75% over random selection. Note that ratio of improvements for PAPBG on the Verizon network is slightly less

as compared to that for the Comcast network. We believe that this is due to the sparse distribution of landmarks on the Verizon network.

Table 7.11 shows the matching statistics for targets to landmarks on the Verizon network. Again with an increase in signal intensity more targets are mapped to the geographically closer landmarks, which explains the gain in the performance of PAPBG. The best performance is again achieved for signal intensity of 30 Kbps per destination node, after which the gains vanish.

7.6.1 PAPBG Costs

PAPBG increases the resolution of geolocation in a metropolitan area by approximately 20-40% as compared to PBG. It achieves this at the cost of sending an additional 600 Kbps data in the network for 100 seconds. The time taken to geolocate the target for PAPBG is again primarily composed of the time taken to collect the RTT seqeunces.

Chapter 8

Initial versions of PBG and PAPBG

PBG and PAPBG can geolocate a target IP address to within a few miles of its actual location in our testbed. However, the exact methodology followed in these algorithms evolved during the course of various experiments that were conducted as a part of this research over the last few years. In this chapter we will present initial verions of these algorithms that laid the foundation to the final versions discussed in Chapters 5 and 6. We also present results that we obtained using these algorithms on some of the datasets collected initially over landmarks on Comcast network.

8.1 PBG Version I

PBG Version I (PBGv1) uses PMFs as a classification feature to geolocate the target to the geographically closest landmark; similar to the approach followed in PBG. However, PBGv1 gives as output the city of the best matching landmark. The techniques followed to compute and compare PMFs are the same as discussed for PBG. However, the difference lies in the way PBGv1 combines results from multiple probe nodes.

8.1.1 PBGv1 algorithm

Given a set of RTT sequences for each landmark and the target collected from one probe node, the PBGv1 algorithm works as follows:

1. Construct PMFs for the RTT sequences measured from the landmarks and the target.

2. Find d_{SSD} of the target's PMF to each of the landmarks' (See Equation 5.3).

3. The landmark with the lowest d_{SSD} serves as the target's location estimate.

Consequently a probe node gives as output the landmark that it believes to be nearest to the target. To combine results from multiple probe we assign a *score* to the location estimate of each probe node as follows. Suppose a probe node is trying to geolocate a target, which is either in city A or city B. The target's PMF is compared using d_{SSD} to the landmarks in city A and city B. Suppose d_A and d_B are the minimum divergences observed over all landmarks in cities A and B, respectively. If $d_A < d_B$, then this probe node will give city A as the location estimate with the following score S.

$$S = \frac{d_B - d_A}{d_B} \tag{8.1}$$

The score $S \in (0, 1)$. Note that if $d_A \ll d_B$, then $S \approx 1$. Conversely, if $d_A \approx d_B$, then $S \approx 0$. Thus, the score S shows the relative confidence the probe node has in its estimate. A higher score signifies a higher difference between the top two candidate cities.

In case of multiple probes nodes, each computes the score for its location estimate. Scores are added for the same location estimates over different probe nodes. The final location estimate of the target is the city (location) which has the highest cumulative 'multi-probe' score over all the probes. For N probe nodes, the multi-probe score, $S_m, \in (0, N)$.

8.1.2 Experiments and Results

8.1.2.1 Data Collection

For evaluating the performance of PBGv1 we used 20 landmarks on Comcast network and 4 probe nodes distributed in Washington-Baltimore area. The distribution of the landmarks in different cities is shown in Table 8.1. Table 8.2 shows the mean pairwise distance (in miles) between landmarks in different cities. The diagonal values in the matrix are the mean pairwise distances between landmarks in the same city. The probe nodes are located as shown in Table 8.3.

Table 8.1: Landmark Locations on Comcast Cable Network

City	# **Landmarks**
Greenbelt (GB)	6
College Park (CP)	5
Hyattsville (HY)	3
Gaithersburg (GA)	1
Germantown (GT)	5

Table 8.2: Mean Distance (miles) between landmarks in different cities

	GB	**CP**	**HY**	**GA**	**GT**
GB	0.7	4.1	4.2	21.8	23.7
CP	4.1	1.4	3.2	19.6	21.0
HY	4.2	3.2	1.1	18.7	20.8
GA	21.8	19.6	18.7	NA	3.8
GT	23.7	21.0	20.8	3.8	1.7

Table 8.3: Probe Node Locations

Probe Node	**City**	**Network**
Probe 0	College Park	University of Maryland
Probe 1	Greenbelt	Comcast Cable
Probe 2	Silver Spring	Verizon FiOS
Probe 3	Potomac	Verizon FiOS

Using the 4 probe nodes, we collected 250 sets of synchronous RTT sequences from the 20 landmarks. In each set, every probe node synchronously sent probe packets to the 20 landmarks at a rate of 5 packets per second per landmark for 100 seconds. This generated a 500 sample RTT sequence for each landmark from every probe. The 4 probe nodes synchronously collected the 250 sets of data at different times of the day, with a random interval between consecutive data collections. We separated 50 sets of data for selecting the PBGv1 parameters (a and ϕ) and used

the remaining 200 sets of data to evaluate the performance of the algorithm using the optimum values of the parameters.

8.1.2.2 Selection of PBGv1 parameters

We followed an approach similar to the one discussed in Appendix B to obtain the best values of a and ϕ using the 50 training sets of data. We explored three penalty functions, ϕ (Logarithmic, Linear and Exponential) for different values of $a \in [0, 1]$ using the *leave-one-out* [19] approach. A target is declared to be geolocated correctly if it lies in the same city as the best matching landmark. Figure 8.1 shows the average performance of correctly matching a target to its actual city for the three penalty functions over different values of a.

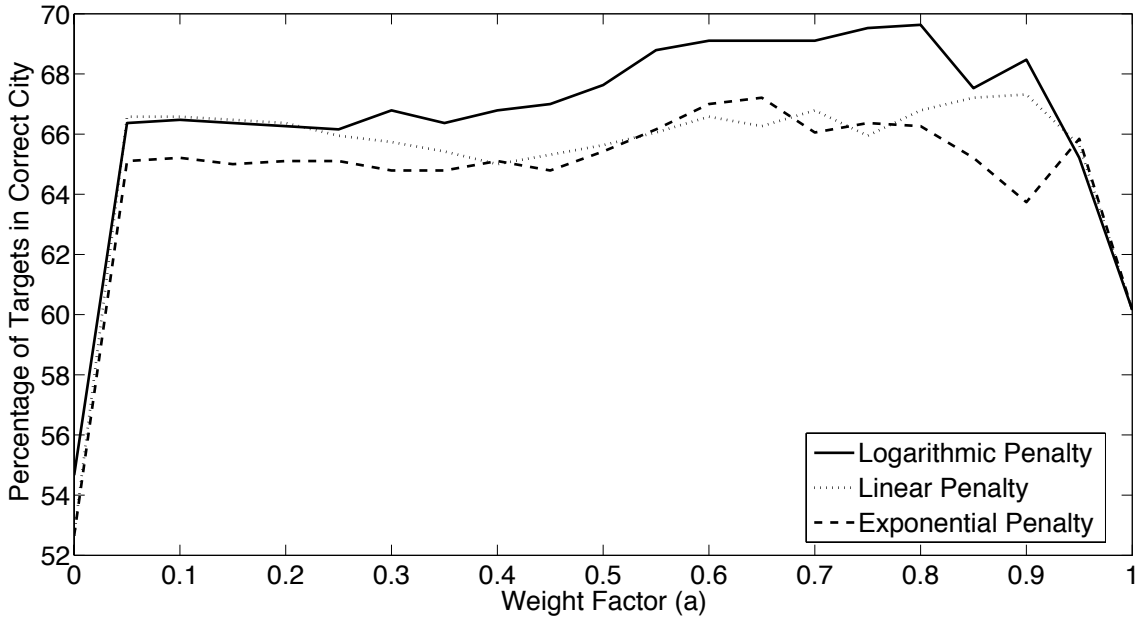


Figure 8.1: Performance of Penalty Functions vs a

As seen from Figure 8.1, the best performance is obtained for the logarithmic penalty function with $a = 0.8$. We chose these as our parameter values for evaluating PBGv1 performance on the remaining 200 sets of data.

8.1.2.3 PBGv1 Performance

Using the parameter values discussed above, we followed the same *leave-one-out* [19] approach to evaluate the performance of PBGv1 algorithm over the 200 sets of data. Table 8.4 shows the location estimates obtained using PBGv1 for targets in different cities[1].

PBGv1 can geolocate an IP Address to the correct city in approximately 70% of the experiments. When it does fail, PBGv1 usually maps the target to a neighboring city. Overall, this approach can geolocate the target to its correct or the nearby city with high confidence ($\approx 85\%$ times). Using the mean distance between landmarks in neighboring cities, we can see that PBGv1 gives us a resolution of around 5 miles in target's location.

In this experiment, 4 probe nodes were used, which means that the confidence score of the multi-probe PBGv1, $S_m \in (0, 4)$(Equation 8.1). Figure 8.2 shows the cumulative density function (CDF) plot of S_m for the correct and the incorrect decisions. Here, a correct decision signifies that the target is matched to its actual city. As shown in the figure, the values of S_m for correct decisions are generally higher than those for the incorrect decisions. This also validates our use of PMF

[1]Since there is only one landmark in Gaithersburg, we assume that the PMF algorithm is correct if this landmark is geolocated in the nearby city of Germantown.

Table 8.4: PMF Results

Target's True Location	**Target's Estimated Location Match (Percent)**				
	Greenbelt	College Park	Hyattsville	Gaithersburg	Germantown
Greenbelt	**71**	18	5	0	6
College Park	13	**69**	11	3	4
Hyattsville	10	16	**58**	2	14
Germantown	8	4	3	7	**78**
Gaithersburg[2]	9	6	1	0	**84**

as a feature for geolocation, since the score for PMF comparisons for the correct decisions are on average higher. We can use this information to formulate a threshold to reduce the probability of an incorrect decision. For instance, with a threshold of 1, all decisions with $S_m \geq 1$ will be taken as valid, while those with $S_m < 1$ will be indeterminate and, hence, invalid. Using the CDF plot, a threshold of 1 will render approximately 45% of the incorrect decisions and 15% of correct decisions invalid. Thus, our probability of an incorrect decision is reduced by 45% to around 20%, while the probability of a correct decision goes down by 15% to 60%. And, the remaining 20% of the cases will be considered indeterminate.

8.1.3 PBGv1 vs PBG

PBGv1 geolocates the target IP address to the city of the best-matching landmark. Thus, its resolution is limited to city-level. Further, it usually fails at city

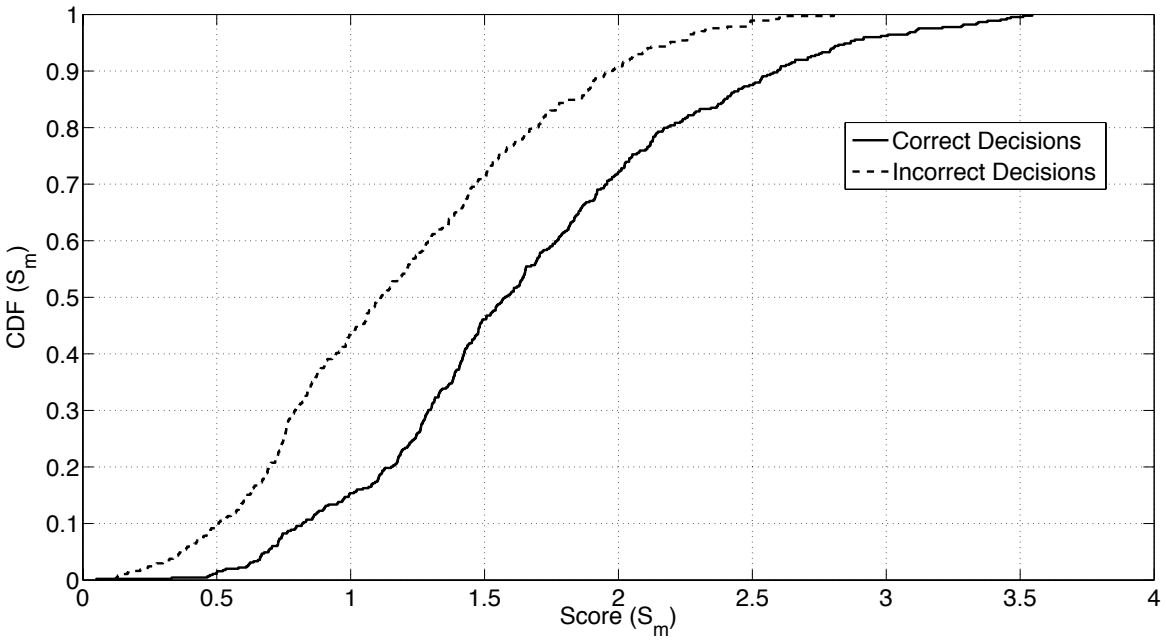


Figure 8.2: CDF plot of multi-probe scores, S_m

boundaries, since network topology does not follow these boundaries. For example, suppose a target is in City A, located close to A's boundary with City B and a landmark lies geographically close, but in City B. In this case, even though PBGv1 matches the target to the geographically closest landmark, but since the two lie in different cities the result will be taken as an incorrect decision. PBG overcomes this shortcoming by geolocating the target to the nearest landmark itself. Nevertheless, PBGv1 was a good first-cut to IP geolocation in a metropolitan area, and the insights developed from PBGv1 helped us develop the final version of PBG.

8.2 PAPBG Version I

PAPBG Version 1 (PAPBGv1) is similar to PAPBG but the perturbation intensity involved is much higher. Our earlier attempts involved introducing perturbation in the RTT sequences of landmarks in one particular city at a time and detect the presence of perturbation in the sequence of the target. These experiments were done with high intensity perturbation traffic (say 5 Mbps) sent for a very short interval (1-2 seconds).

8.2.1 PAPBGv1 algorithm

The technique requires a set of **noise generator** nodes. Note that we use the terms 'noise' and 'noise generator' instead of 'perturbation' and 'perturber', since the intensity of induced signal in this instance is much higher and looks more like noise. These noise generators send noise traffic to a set of landmarks in the same geographic location (usually within the same city). This traffic induces a strong signature in the RTT sequences of the target if it is nearby.

The technique works as follows. One of the probe nodes sends small ICMP Echo Request packets (of size 20 bytes each) to the target at a nominal rate, say 10 packets per second. The remaining probe nodes, acting as noise generators, send noise traffic to the landmarks in one city. This noise traffic comprises large ICMP requests packets (of size > 250 bytes) sent at a high rate (e.g. 500 packets per second). This noise is sent for a very small duration, around $1-2$ seconds. The goal is to introduce a small but detectable traffic signature in the network around the

landmarks in one location. In case the target is in the vicinity of where the noise is directed, the noise traffic is noted as a strong signature in the RTT sequence of the target. If the target is not in the vicinity of the noise destinations, its RTT sequence is not affected by the noise.

Consider a scenario with two landmarks in nearby cities, Greenbelt and College Park in Maryland, with a target in Greenbelt.

Figures 8.3 show the plot of RTT sequences for the target (T, whose location we want to find) and the two landmarks in Greenbelt (GB) and College Park (CP). In this case, the probe node sent synchronous probe packets at a rate of 5 packets per second to the two landmarks and the target for 100 seconds each.

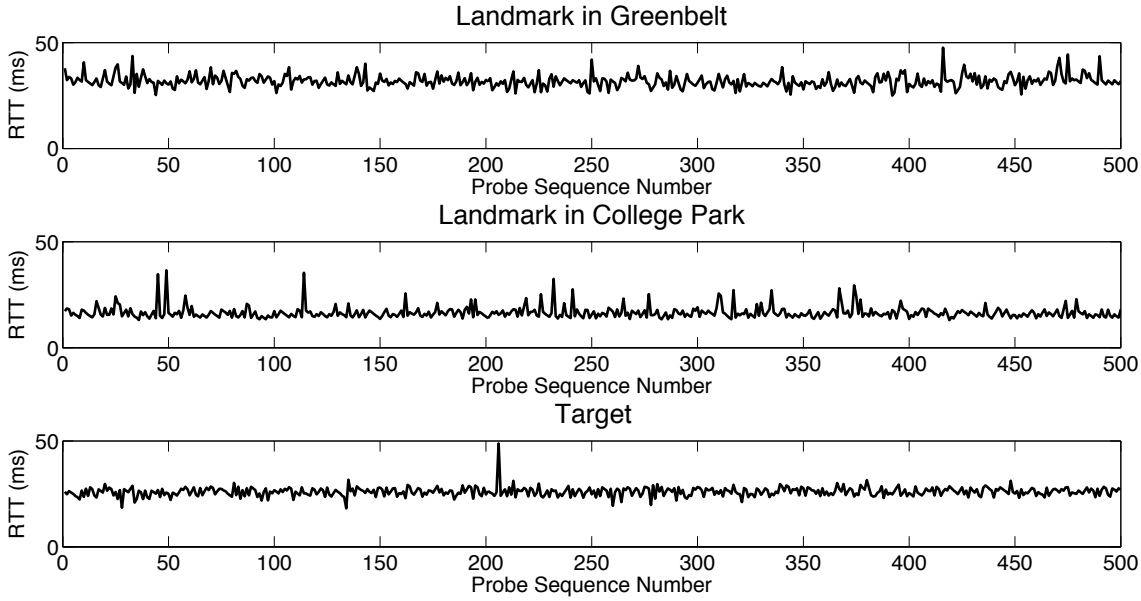


Figure 8.3: Plots of RTT Sequences of the target and two landmarks with no noise

Using PBGv1, with $a = 0.8$ and $\phi() = 1 + \log_2()$, the divergence values obtained are as follows:

$$d_{SSD}(T\|GB) = 1.06$$

$$d_{SSD}(T\|CP) = 1.09$$

As can be seen, in this case PBGv1 fails to give a location estimate with high confidence. Both the landmarks show similar values for the divergence and it is not clear which landmark is truly closer to the target. With PAPBGv1 the probe node sends 20 byte probe packets to the target at a rate of 10 packets per second for 20 seconds. Simultaneously, two noise generators send directed noise traffic during time intervals $t = 5-7, 10-12$ and $15-17$ seconds to the landmark in Greenbelt. The noise packets in this instance are 256 bytes each, sent at a rate of 500 packets per second. The probe node detects the signature of noise in the RTT sequence of the target (see Figure 8.4). However, sending the noise to the landmark in College Park does not exhibit any strong pattern in the target's RTT sequence. Thus, the injected noise is able to discriminate between neighboring cities and correctly geolocate the target to Greenbelt.

8.2.1.1 Noise Pattern Matching

Suppose our landmarks are distributed in a set of cities denoted by $\mathcal{C}$. Our goal is to find in which city in this set $\mathcal{C}$ our target is present. The target is probed while noise is sent, at pre-determined intervals, to all landmarks in one city at a

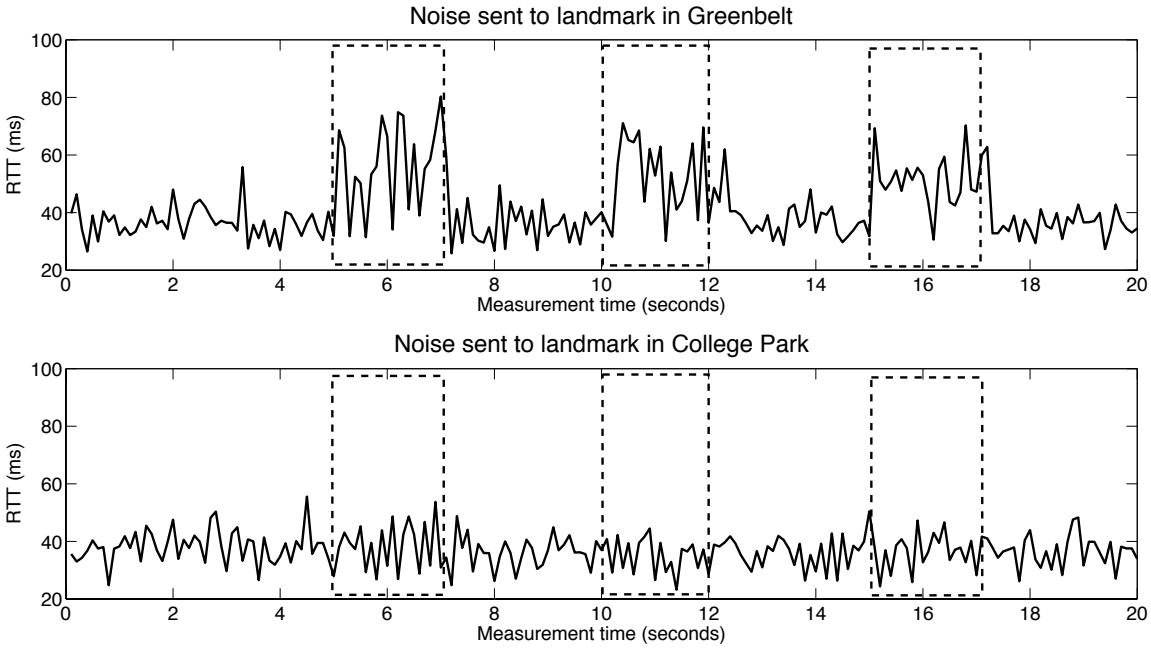


Figure 8.4: RTT Sequence of the target with noise injected at landmarks in Greenbelt and College Park

time. By construction, the noise injection intervals for different cities are orthogonal. To detect the noise signature in the target's RTT sequence, we use the following approach.

For each city $B \in \mathcal{C}$, we construct a signature vector $S_B(t), t = 1, \ldots, N$, such that

$$S_B(t) = \begin{cases} 1, & \forall\ t \in W^B_{noise} \\ \epsilon, & \forall\ t \notin W^B_{noise} \end{cases}$$

Here W^B_{noise}, is the noise window for city B, which signifies the interval when noise is sent to landmarks in city B. And ϵ is chosen so that $\sum_{t=1}^{N} S_B(t) = 0$. Now, the

inner product of $S_B(t)$ with the target's RTT sequence $X(t)$ is taken to give IP_B. The $S_B(t)$ is designed in this manner for the following reasons:

1. **Target in city** B: When the target is in city B, the signature sequence $S_B(t)$ and the target's RTT sequence $X(t)$ will have a correlated pattern, and the inner product $IP_B \gg 0$.

2. **Target in city** $B' \in \mathcal{C}$: When the target is not in city B, but in some other city $B' \in \mathcal{C}$, then $X(t)$ will see an increase in RTT values outside the noise window of B (inside the noise window of B'). In this case we would expect $IP_B \ll 0$ and $IP_{B'} \gg 0$.

3. **Target in city** $Z \notin \mathcal{C}$: In case the target is not present in any city in the set $\mathcal{C}$, the target' s RTT sequence will not show any pattern. As a result the inner product $IP_B \approx 0 \; \forall B \in \mathcal{C}$.

Thus, by carefully designing a noise signature vector $S_B(t)$ for each city B, we can map the target to one of the cities (assuming that the induced noise is sufficient and that the target is in a city with landmarks).

8.2.2 Experiments and Results

8.2.2.1 Data Collection

We used the same 20 landmarks on Comcast network (Table 8.1) and the 4 probe nodes (Table 8.3) used for evaluating the performance of PBGv1 (See Section 8.1.2). For each landmark serving as the target, we started with a smaller set

of candidate cities where it can possibly be located. For this we chose the top two cities (based on S_m) from the PMF algorithm results. Probe 0 at the University of Maryland, College Park served as the regular probe node. Probes 2 and 3 at Silver Spring and Potomac acted as noise generators. Probe 1 at Greenbelt was not used in this setup since Greenbelt was one of the candidate cities, and using this probe node would have resulted in artifacts. Finally, we removed the landmark in Gaithersburg from this test set and used the remaining 19 landmarks (see Table 8.1), since there was no other landmark in Gaithersburg.

Our landmarks on Comcast network have a download bandwidth of 10 Mbps and upload bandwidth of 2 Mbps. To ensure that we do not create bottlenecks in the last hop of one of the landmarks, we chose to send noise at a rate of 1 Mbps from each noise generator to each landmark. We first chose one candidate city to send noise to. The probe node sent probe packets to the target at a rate of 5 packets per second for 25 seconds. Meanwhile the noise generators sent 256 byte noise packets at a rate of 500 packets per second to each landmark. Noise was sent 5 times starting at time instants $t = 4, 8, 12, 16, 20$ seconds, for a duration of 2 seconds each time, to landmarks in one of the chosen candidate cities. The experiment was then repeated with noise sent to landmarks in the other candidate city. By sending noise multiple times, we decreased the probability of picking up a false signature. We analyzed the target's RTT sequence for noise signature and declared the target to be in the city which gave the strongest signature.

Table 8.5 shows the geolocation results obtained with PAPBGv1. As shown, we can geolocate the targets in Greenbelt and College Park with high confidence

Table 8.5: PAPBGv1 Results

Target's True Location	**Target's Estimated Location Match (Percent)**				
	Greenbelt	College Park	Hyattsville	Germantown	No Match
Greenbelt	**92**	0	0	0	8
College Park	0	**89**	0	0	11
Hyattsville	0	0	**52**	0	48
Germantown	0	0	0	**61**	39

($\approx$ 90% of the times). However, for targets in Hyattsville and Germantown, this technique fails to perform well. We believe this is because our landmarks in these cities are very sparse (Table 8.2), which results in effectively low noise intensity in the vicinity of the target. In contrast, for Greenbelt and College Park, where the landmarks are relatively densely distributed, PAPBGv1 results in a clear stronger signature in the target's RTT sequences.

Finally, we note that in all of our experiments, PAPBGv1 never resulted in a misclassification. The target was either geolocated to the correct city or the result was indeterminate. Thus, given a sufficiently dense deployment of landmarks, the noise injection technique can geolocate a target to the correct city with very high confidence, thereby providing a resolution of approximately 1 mile.

8.2.3 PAPBGv1 vs PAPBG

PAPBGv1 sends high intensity noise traffic in the network, albeit for a very short interval. Be inducing and detecting the pattern of this noise traffic in the target's RTT sequence, PAPBGv1 can geolocate a target to correct location (city) with high confidence (provided sufficient number of landmarks are present in the vicinity). However, this approach raises concerns of denial of service attacks and can possibly disrupt traffic near the landmarks. Further, by sending a large amount of traffic near the target can make the target aware that is being probed. Compared to this, our final version of PAPBG uses perturbation traffic at a much lower intensity (10-50 Kbps per destination node). Instead of directly inducing a signature in the target's RTT sequence, PAPBG aims to slightly enhance the background traffic's signature so that this can be more effectively captured in the resultant PMF comparisons. Nevertheless PAPBGv1 shows that it is possible to perform high-resolution geolocating using high-intensity traffic. This can serve as a good technique for critical applications like e-911, where the resultant benefit far outweighs concerns of traffic disruption in the network.

Chapter 9

Future Work

Our geolocation algorithms, PBG and PAPBG, capture, detect and match patterns in the RTT sequences for geolocating a target IP address in a metropolitan area. Our *pattern recognition* based geolocation strategy is a first of a kind approach and we presented results on a real network. In this chapter we will discuss some open problems in this research which can followed up for future work.

9.1 Adding Landmarks

Our geolocation approach matches a target to one of the landmarks in our testbed. We assume that sufficient number of landmarks are available for comparing the RTT sequences of the target with. Adding landmarks to the testbed is an open problem and an area of research in its own. One possibility to encourage users to volunteer as landmarks is by providing incentives, like an unlimited storage for their email accounts, or a discount on internet bills, etc. In addition to these direct incentives, users can also be encouraged to volunteer as landmarks by providing indirect incentives like Google Latitude and Facebook Check-in. While adding more landmarks, we also need to ensure that no privacy concerns are raised.

9.2 Scalability

Having more landmarks in the testbed will improve the resolution of our geolocation algorithms. However increase in the number of landmarks raises scalability issues for both PBG and PAPBG.

9.2.1 sPBG

For PBG having more landmarks will proportionately increase the total number of probe packets, and hence the overall traffic, sent from each probe node. The increase in traffic can lead to possible congestion at the first hop near the probe node itself which results in artifacts in the RTT sequences. These artifacts can lead to mis-classification of a target's PMF to a distant landmark.

A possible solution to this problem is to develop a 'smart PBG' (sPBG) that collects PMFs of the landmarks at different times beforehand and characterizes these PMFs to get 'representative PMFs' for each landmark at each probe node. These representative PMFs constitute a PMF bank. We have shown in Chapter 7 that a simple 'pick and choose' PMF bank does not work. Instead we will need to build adaptation models that can suitably adapt one or more of the representative PMFs from the PMF bank using a few RTT values observed from each landmark. Given a target IP address, each probe node instead of measuring 500 RTT values for each landmark, now measures fewer number of RTT values (say 50) per landmark, and uses these to adapt and estimate the 'most suitable' PMF of the landmark from its PMF bank. Note that the landmarks are a part of our testbed, and hence, the

PMF bank for each landmark can be built in advance. However, we do not have any prior information about the target, and hence we need to measure the full 500 sample RTT sequence for the target to construct its PMF. Thus, sPBG reduces the amount of traffic sent from probe node at a given time, and makes PBG scalable.

9.2.2 sPAPBG

Constructing a PMF bank can reduce the amount of traffic sent from probe nodes in case of PAPBG also. But in this case, we have an additional perturber node which sends signal traffic at a much higher intensity. With an increase in the number of landmarks the total traffic sent by the perturber, thus, increases with a larger proportion than that sent by the probe nodes. To solve this issue, we plan to use 'smart PAPBG' (sPAPBG) on a smaller subset of landmarks. PBG can be first used to get a subset of landmarks which give low values of divergence with the target. And then PAPBG can send perturbation signal selectively to these landmarks to get a higher confidence in the location estimate. Thus, PBG and PAPBG can form a two-step approach to geolocation in a metropolitan area.

9.3 Feedback loop for PAPBG

PAPBG can be further refined by adding a feedback loop that modulates the signal intensity based on the network topology conditions. Our current protocol sends signal to all landmarks and target at a time. An alternative strategy is to send signal to only one landmark (and not the target) at a time, and detect the

signature of the signal in the RTT sequence of the target. The geographically closest landmark should induce the strongest signature. This "one-landmark-at-a-time" approach may provide a much higher resolution in target's location with high confidence. However, this will involve sending signal traffic at a high intensity (> 1 Mpbs), which can raise issues of network traffic disruption. Further experiments are needed to evaluate the effect of high intensity signal on the network traffic to investigate the feasibility of this idea.

9.4 Unresponsive targets

We assume that targets respond to pings. This need not be the case as many routers and hosts are configured not to respond to ICMP messages. We could use a slightly modified form of perturbation to geolocate such "non-responsive" target. We could send signal to the target and measure the variation in landmarks' RTTs.

9.5 Better Classifiers

Our algorithms currently use *nearest neighbor classification* [11], which is sub-optimal. The performance may be improved using better classifiers. Using Support Vector Machines (SVM) [9] is one possible solution. However, the standard kernel functions of SVMs cannot be directly applied for PMF classification, since the PMFs do not follow Euclidean distance metrics. As part of future work, (suitably modified) divergence based kernel functions proposed for SVMs [24, 20] can be investigated.

In addition to the two algorithms discussed in this paper, we also explored

Wavelet Analysis to detect 'sharp singularites' in the RTT sequences [21], which may be the result of some network activity in the vicinity. By detecting and matching these singularities we hoped to find the landmark which is closest to the target. However, this approach also did not perform well, suggesting that usually there is no detectable temporal pattern in the RTT sequences.

9.6 More ISPs

Our current experiments have been confined to landmarks on the Comcast and Verizon networks. As part of future work, these experiments can be extended to other ISPs.

Chapter 10

Conclusion

In this thesis, we have presented two algorithms for geolocation in a metropolitan area based on pattern recognition. Existing geolocation techniques that either use static or measurement-based approaches fail to geolocate a target in metroploitan area, and in fact perform worse than a 'random selection' technique. Compared to these, our algorithms geolocate a target IP address to within a few miles of its actual location.

We have explored the use of Probability Mass Functions (PMFs) as a feature for geolocating a target and have proposed a new shift-invariant distance metric for comparing the PMFs. We show that PBG based on PMF comparison has negligible network overhead and can estimate the target's location with a resolution of approximately 5 miles. With an increase in the density of landmarks the performance of PBG improves.

For improved resolution, we introduce PAPBG that uses additional 600 Kbps aggregate traffic to obtain finer location estimates. This approach is more intrusive and involves sending traffic to the network, albeit at a low intensity per destination node ($\leq$ 50 Kbps). With a small amount of perturbation, PAPBG can geolocate the target to within 3 miles of the nearby landmark.

Appendix A

Selection of PMF Computation parameters

For PMF computation we had two free parameters to decide: Sampling Rate of RTTs and Observation Duration. We followed the following strategy to select the values of these parameters.

A.1 Observation Duration

We collected 50 sets of validation data from the 20 landmarks on our testbed - 12 landmarks on Comcast network and 8 on Verizon network (See Tables 7.1 and 7.2). We used Probes 0 and 1 in this data collection (Table 8.3). Each data set consists of RTT sequences collected at a frequency of 10 samples per second per destination node collected for 500 seconds. To simulate different observation durations, we picked samples corresponding to the first 25, 50, 75, 100, 150, 250 and 500 seconds from each RTT sequence. We ran PBG experiments for each observation time on the 50 datasets. PMFs were compared with d_{SSD} (See Appendix C) as the distance metric and results from multiple probe nodes were combined using minimum mean divergence (See Appendix D). Figure A.1 shows the mean error vs observation duration for the two networks.

As can be seen from this plot, the mean error initially decreases with an increase in observation duration. However, after the observation duration increases

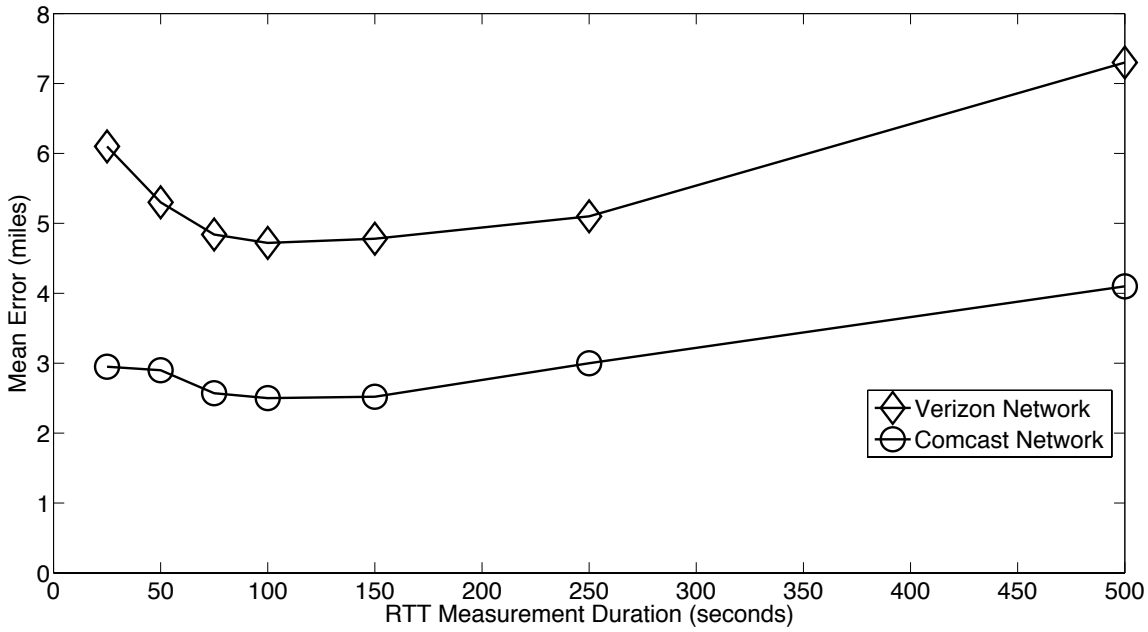


Figure A.1: PBG performance vs Obsevation Duration

to more than 150 seconds, the performance deteriorates for both networks. The performance remains more or less comparable for observation durations of 75, 100 and 150 seconds. Based on these results, we chose an observation value of 100 for our subsequent experiments.

A.2 Sampling Frequency

To obtain the best values for sampling fequency, we fixed the observation duration as 100 seconds and collected additional datasets for different sampling frequencies. We explored 8 values of sampling frequency - 0.5, 1, 2, 5, 10, 20, 50 and 100. For each value we collected 30 datasets from 20 landmarks in our testbed

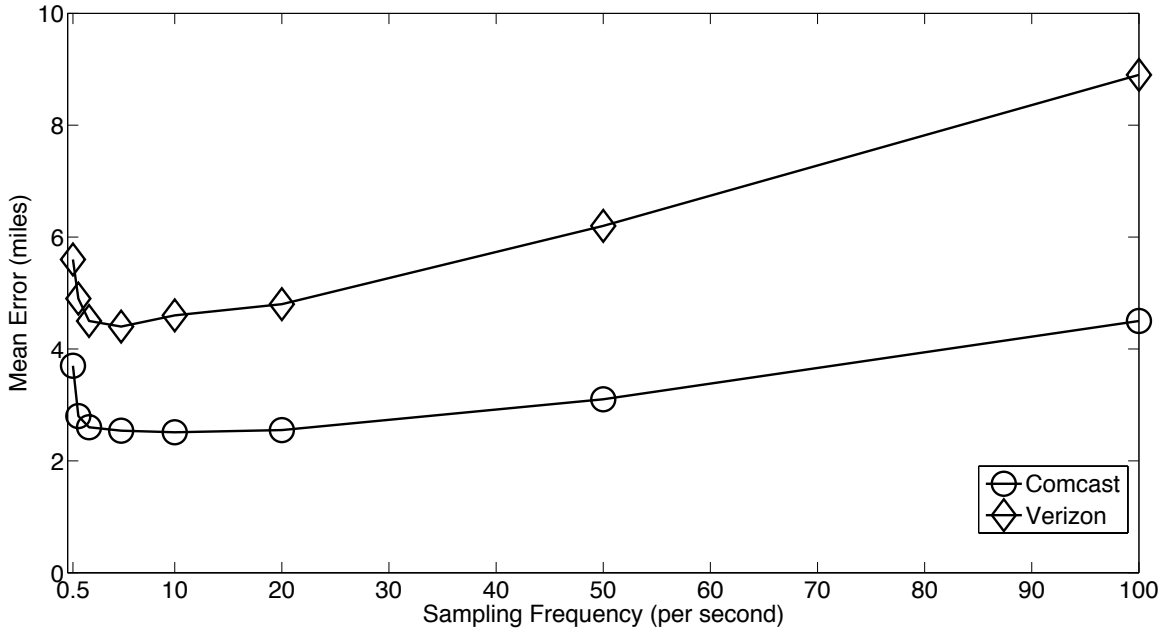


Figure A.2: PBG performance vs Sampling Frequency

from two probe nodes. We ran PBG on these datasets. Figure A.2 shows a plot of mean error versus sampling frequency using PBG for the two networks.

The best performance is obtained for sampling frequencies $\in \{2, 5, 10\}$. As expected, a too low or a too high value results in a deterioration of performance. We chose a frequency of 5 samples per seconds for our data collection.

Appendix B

Selection of PMF Comparison parameters

For comparing Probability Mass Functions (PMFs), PBG and PAPBG use two parameters :- the weight factor a and the penalty function ϕ. To find optimal values of a and ϕ, we collected 30 additional sets of data and evaluated the performance of PBG on this dataset for different values of the parameters. These datasets, thus, serve as training data for the PMF parameters. All evaluations were done separately for the landmarks on the Comcast and Verizon networks. Given landmarks on a new service provider, we would collect training datasets for those landmarks and run this training procedure to get the optimal values for the new setup.

PMFs were compared with d_{SSD} (See Appendix C) as the distance metric and results from multiple probe nodes were combined using minimum mean divergence (See Appendix D).

We explored the following three monotonically increasing penalty functions.

1. **Logarithmic Penalty:**

   $\phi(s_{min}) = \max\{0, 1 + \log_2(s_{min})\}$

2. **Linear Penalty:** $\phi(s_{min}) = s_{min}$

3. **Exponential Penalty:** $\phi(s_{min}) = 2^{s_{min}}$

For each of the above penalty functions, we explored different values of $a \in$

$[0, 1]$. To evaluate these parameter values we followed the *leave-one-out* [19] approach on the 30 sets of data. With a fixed penalty function and a, we chose one landmark as the target and tried to geolocate it using the rest of the landmarks in one dataset. We converted the RTT sequences collected from each probe node into PMFs and compared the PMF of the target to those of the landmarks to get divergence values for each landmark. (see Equation 5.3). We repeated this computation for each probe node. The final result was the landmark with the minimum mean divergence over all probe nodes. We iterated this step over all landmarks, with one landmark serving as the target in each iteration. We repeated these steps over all datasets for the three penalty functions with 100 values of a uniformly spaced on $[0, 1]$. Figure B.1 shows the mean error in geolocating a target on the Comcast network for the three penalty functions over different values of a. And Figure B.2 shows the statistics for the Verizon network.

As seen from Figures B.1 and B.2, the best performance is obtained for the exponential penalty function with $a = 0.9$ for Comcast network and $a = 0.95$ for Verizon network. We chose these as our optimum parameter values for PBG. PAPBG uses the same PMF computations and comparisons as PBG. So we use the same set of parametere values for PAPBG as well.

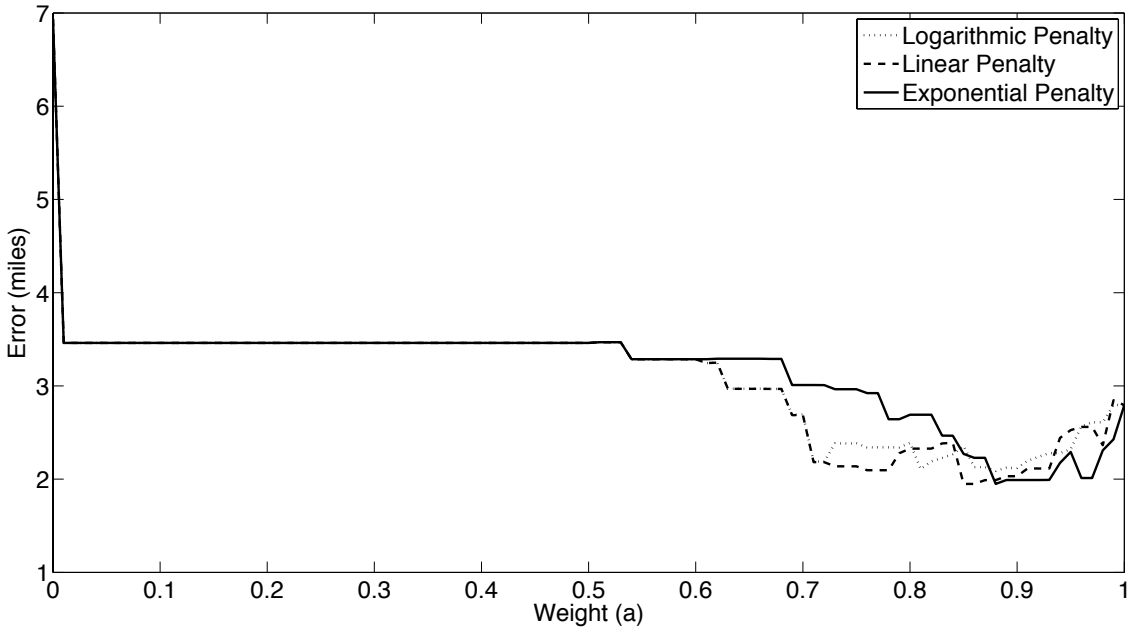


Figure B.1: Performance of Penalty Functions vs a for Comcast

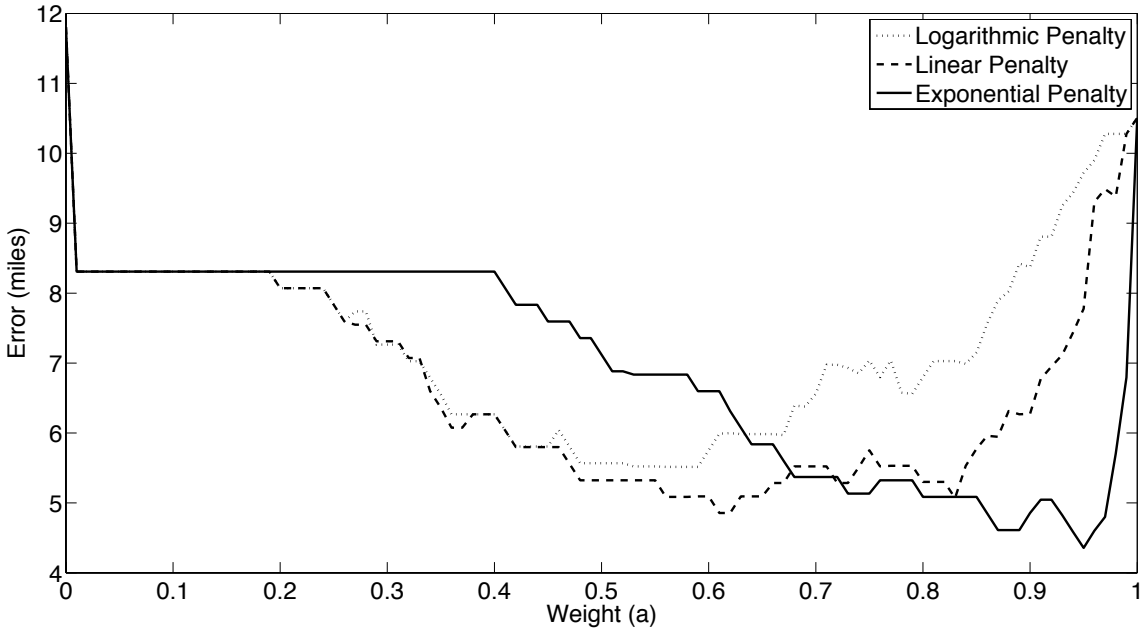


Figure B.2: Performance of Penalty Functions vs a for Verizon

Appendix C

PMF Distance Metric

We explored two distance metrics to compare PMFs for PBG:- Shifted Symmetrized Divergence, d_{SSD} (Equation 5.3), and Total Variation, V(Equation 5.4). In Appendix B we empirically derive the 'best' values for weight factor a and penalty function ϕ for computing d_{SSD}. We used the same dataset to evaluate performance of PBG using total variation as distance metric. Table C.1 lists the mean errors obtained from the two metrics for the two networks.

Table C.1: Mean Error (in miles) for Shifted Symmetrized Divergence and Total Variation

Network	d_{SSD}	V
Comcast	2.6	6.1
Verizon	4.7	9.2

Shifted Symmetrized Divergence gives better results as compared to Total Variation.

Appendix D

Multi-Probe PBG

For combining results from multiple probe nodes in multi-probe PBG we evaluated two decision rules: 'minimum mean divergence' and 'min max divergence'. The results in Appendix B were derived using 'minimum mean divergence'. We used 'min max divergence' to combine results from multiple probe nodes on the same dataset. Table D.1 compares the performance of the two decision rules.

Table D.1: Mean Error (in miles) for Minimum Mean Divergence and Min Max Divergence

Network	Minimum Mean	Min Max
Comcast	2.6	13.2
Verizon	4.7	15.3

Bibliography

[1] GeoNetMap, Geobytes, Inc. [Online]. `http://www.geobytes.com/GeoNetMap.htm/`.

[2] GeoURL [Online]. `http://www.geourl.org/`.

[3] IP Address to Latitude/Longitude [Online]. `http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll/`.

[4] MaxMind GeoIP City Database. [Online]. `http://www.maxmind.com/` .

[5] Net World Map [Online]. `http://www.networldmap.com/`.

[6] Hirotugu Akaike. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21:243–247, 1969. 10.1007/BF02532251.

[7] R Benzi, A Sutera, and A Vulpiani. The mechanism of stochastic resonance. *Journal of Physics A: Mathematical and General*, 14(11):L453, 1981.

[8] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. van Mieghem. Analysis of end-to-end delay measurements in Internet. In *Proc. Passive and Active Measurement Workshop (PAM 2002)*, Fort Collins, CO, USA, 2002.

[9] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[10] David D. Clark, Craig Partridge, Robert T. Braden, Bruce Davie, Sally Floyd, Van Jacobson, Dina Katabi, Greg Minshall, K. K. Ramakrishnan, Timothy Roscoe, Ion Stoica, John Wroclawski, and Lixia Zhang. Making the world (of communications) a different place. *SIGCOMM Comput. Commun. Rev.*, 35(3):91–96, 2005.

[11] Thomas M. Cover and P. Hart. Nearest Neighbor Pattern Classfication. *IEEE Transactions on Information theory*, 13(1), Jan, 1967.

[12] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[13] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[14] Michael J. Freedman, Mythili Vutukuru, Nick Feamster, and Hari Balakrishnan. Geographic locality of IP prefixes. In *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 13–13, Berkeley, CA, USA, 2005. USENIX Association.

[15] Luca Gammaitoni, Peter Hänggi, Peter Jung, and Fabio Marchesoni. Stochastic resonance. *Rev. Mod. Phys.*, 70(1):223–287, Jan 1998.

[16] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. Constraint-based geolocation of Internet hosts. *IEEE/ACM Transactions on Networking*, 14(6):1219–1232, December 2006.

[17] Tomasz Imieliński and Julio C. Navas. GPS-based geographic addressing, routing, and resource discovery. *Commun. ACM*, 42:86–92, April 1999.

[18] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. Towards IP geolocation using delay and topology measurements. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 71–84, New York, NY, USA, 2006. ACM.

[19] Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. pages 1137–1143. Morgan Kaufmann, 1995.

[20] Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37:145–151, 1991.

[21] S. Mallat and W. L Hwang. Singularity Detection and processing with wavelets. *IEEE Transactions on Information theory*, 38(3), March,1992.

[22] D. Moore. Where in the world is netgeo.cardia.org? In *Proceedings of INET 2000*, Stockholm, Sweden, June, 2000.

[23] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 173–185, New York, NY, USA, 2001. ACM.

[24] Pedro Moreno Purdy, Pedro J. Moreno, Purdy P. Ho, and Nuno Vasconcelos. A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications. In *In Advances in Neural Information Processing Systems 16*. MIT Press, 2003.

[25] Stefan Steiniger, Mortiz Neun, and Alistair Edwardes. Foundations of Location Based Services. *Lecture Notes on LBS*, 2006.

[26] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. Geolocalization on the Internet through constraint satisfaction. In *WORLDS'06: Proceedings of the 3rd conference on USENIX Workshop on Real, Large Distributed Systems*, Berkeley, CA, USA, 2006. USENIX Association.

[27] Artur Ziviani, Serge Fdida, José F. de Rezende, and Otto Carlos M. B. Duarte. Improving the accuracy of measurement-based geographic location of Internet hosts. *Comput. Netw.*, 47(4):503–523, 2005.