Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 29 (2006) 3295–3305
computer communications
www.elsevier.com/locate/comcom

Estimation of network distances using off-line measurements

Prasun Sinha [a,*], Danny Raz [b], Nidhan Choudhuri [c]

[a] Department of CSE, Ohio State University, USA
[b] Department of Computer Science, The Technion, Israel
[c] Department of Statistics, Case Western Reserve University, USA

Received 14 January 2006; received in revised form 14 May 2006; accepted 15 May 2006
Available online 19 June 2006

Abstract

Several emerging large-scale Internet applications such as Content Distribution Networks, and Peer-to-Peer networks could potentially benefit from knowing the underlying Internet topology and the distances (i.e., round-trip-time) between different hosts. Most existing techniques for distance estimation either use a dedicated infrastructure or use on-line measurements in which probe packets are injected into the network during estimation. Our goal in this paper is to study off-line techniques for distance estimation that do not require a dedicated infrastructure. To this end, we propose a metric termed "depth" and we observe that together with a quadratic function on the geographic distance, it can predict the network distance with high accuracy using multi-variable regression. When used for closest server selection, our approach performs much better than random server selection, and similar to the on-line metrics. Our approach incurs low overhead and can be deployed easily with some DNS extensions.

Keywords: Network distance estimation; Round trip time estimation

1. Introduction

In recent years more and more popular Web items are being replicated to facilitate their fast retrieval. The web caching schemes in which web items are cached and delivered to users from the local cache are evolving into global Content Distribution Networks (CDN). A Content Distribution Network [1–3] is an infrastructure that distributes the content of popular items to multiple geographically dispersed servers. When a client requests an item, the CDN directs it to the "best" replica, that is, the closest replica in terms of user observed latency. One of the main factors in finding this "good" replica is the network distance in terms of RTT (round trip time) between the client and the possible content servers. Thus, it is important for content distribution service providers to know these distances in order to make good choices.

Other examples where multiple copies of the same item are stored in the Internet are the peer-to-peer file sharing applications such as Gnutella and Napster. Here again, while searching for a specific file, the user can be directed to one out of several currently on-line copies, but it will be beneficial to direct the user to the "closest" copy. New emerging peer-to-peer overlay network applications can also use distance information in order to make the overlay network "distance aware". But, in this overlay network it will make very little sense to connect clients from say San Francisco, to peers in, say New York. Thus it is important to accommodate distance information when constructing peer-to-peer overlay networks [4].

The importance of retrieving network distance information as indicated by the above examples has resulted in several approaches toward collecting such information. Due to the substantial overhead both in terms of delay and network traffic, a number of projects aimed at collecting network distance information and distributing it to various applications are gaining popularity [5–7]. The information provided by such mechanisms is

* Corresponding author. Tel.: +1 614 659 9712.
E-mail addresses: prasun@cse.ohio-state.edu (P. Sinha), danny@cs.technion.ac.il (D. Raz), nidhan@nidhan.cwru.edu (N. Choudhuri).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.05.010

somewhat less accurate [8,6], and they require a special dedicated complex infrastructure. To address these drawbacks a light-weight approach of using "landmarks" has been recently proposed [4,9]. In this approach the distance information is computed from measuring distances from the client to a small number of well-known landmarks. Although this approach provides good distance information, it also requires the construction and maintenance of dedicated infrastructure of landmarks, and some on-line measurements. Techniques that require messaging during distance estimation are termed "on-line" and the others are referred to as "off-line". On-line techniques are typically time consuming and require high overhead of messages.

Our goal in this paper is to propose and study a network distance estimation technique that does not involve on-line measurements or a dedicated infrastructure. The main idea is to use *static* information for this purpose. One well-known type of such information is the geographic location of the host. As geographic distances are believed to be insufficient in predicting network latency [10], we propose a new off-line metric called DEPTH, which is the average RTT from a given network element to the nearest backbone network (precise definition is given later in the paper). Our approach is to use multi-variable regression techniques that exploit the combined information contained in these metrics to predict the minimum RTT. To contrast our results with on-line metrics, we use the number of hops and the number of autonomous systems on the route, both of which require on-line traffic for measurement.

The first step towards answering our question was to collect real Internet data. This was done using traceroute servers. Overall we performed more than 600,000 traceroute operations using more than 3000 hosts and 24 servers (10 sets of traceroutes between each host-server pair), worldwide. The data was split into training and test sets. The training data was used for performing the regression analysis and the test data was used to study the performance of closest server selection. Based on the statistical analysis we design an algorithmic framework that computes an approximated network RTT from the given static data. The metrics were evaluated to measure their performance in the context of performing topologically aware operations such as server selection. *We observe that DEPTH together with quadratic distance provides the best off-line metric and it performs similar to the best on-line metric*.

It is important to note that our technique does not require any infrastructure, and it can be easily deployed using existing extensions of the DNS service known as DNS-LOC [11]. This is basically a format defined in RFC1876 [12] that allows addition of location information to the DNS service. Using similar methods or extensions to the fields defined in RFC1876, one can add additional information such as geographic location and depth. Then, either the host, the local DNS server, or the application (CDN, P2P) server can perform our algorithm (which is a very simple computation) and choose the desired server.

The main contributions of this work are threefold.

- We identify the network depth as an important off-line metric that provides (in combination with location information) enough information to enable accurate topologically aware operations.
- We provide a detailed statistical analysis of various off-line and on-line metrics, and provide an analytical study of the correlation of these variables.
- We show that topologically aware server selection can be done without any need for dedicated infrastructure or on-line measurements. Furthermore, the accuracy of such server selection is similar to the best on-line methods, and the deployment of our scheme is immediate.

The rest of this paper is organized as follows. In the next section, we further discuss the different approaches for topological distance data and explain our proposed framework. Then in Section 4 we explain our data collection process, and in Section 5 we provide the statistical analysis of the data. In Section 6 we present and discuss the performance of our method for the problem of closest server selection. Finally, we conclude with pointers to future research in Section 7.

2. Related work

The correlation between different network metrics such as the number of hops on the route and RTT has been lately revisited due to the availability of new measurement infrastructures and tools. Skitter [13] is a tool that measures the forward path and round trip time (RTT) to a set of destinations by sending probe packets through the Internet. Based on this tool, several reports have been published lately that address correlation between different network parameters [14–16]. However, as far as we are aware, none of these reports provide a rigorous statistical analysis of the data with respect to multi-variable correlation and the joint ability of off-line metrics to predict the RTT.

The work of Carter and Crovella [10] studied the benefits of static vs. dynamic server selection. They report that dynamic selection is much better than static one. However, their study deals with server network load which is itself dynamic. Also, the static approach they considered did not use the combination of various metrics like in our approach. Moreover, we are introducing the new metric – DEPTH – which has not been studied before.

Realizing that dynamic server selection may cause significant network overload, several efforts have been launched for performing network measurements in order to provide network distance estimation. IDMAPS [8,6] is one such effort. In the IDMAPS architecture, dedicated tracers placed in the network perform on-line measurements. A set of information servers use these measurements to compute and disseminate distance information to hosts that request such information.

Recently, in [4] a off-line approach to the distance estimation problem based on network landmarks was proposed. This scalable approach can provide estimations that are good enough for topologically aware operations such as server selection. In their proposal the authors do not distinguish between servers that are within 10 ms of each other for computation of server selection accuracy. This is a reasonable approach (in our studies we have tried deviations of 10–50 ms) as it matters very little which server we choose if they are sufficiently close in terms of network distance. As opposed to [4], our approach is easier to deploy as it does not require the maintenance of a dedicated infrastructure of landmarks.

Several variations of the landmark approach has been explored recently. Ref. [17] explores the accuracy of embeddings of virtual coordinates in Euclidean space using the concept of virtual landmarks. The distance to a virtual landmark is defined as a linear combination of distances to actual landmarks. A similar mechanism for dimensionality reduction using principal component analysis has been proposed in [18]. Lighthouses [19] is a random strategy to select landmarks for computing a new nodes' coordinates. Mithos [20] is a strategy that uses the closest landmarks to compute the coordinates of new nodes for better accuracy of computing short distances. A hybrid mechanism was proposed in [21] along with an efficient strategy for computing the closest servers. It has been also shown to be robust against malicious users. The authors in [21] observe that maintenance of explicit landmarks is an overhead and study the effect of using locations of other nodes as landmarks. However, off-line techniques based on linear regression have not been studied before.

3. Distance estimation metrics: off-line vs. on-line

We classify the techniques for Internet distance measurement into *on-line* and *off-line*. On-line techniques require messaging during the computation of network distance, whereas off-line techniques only rely on measurements taken at other times. By definition, off-line techniques can not be aware of dynamic factors such as server load and network congestion. However, the low message complexity and the quick computation of network distances makes off-line estimation an attractive choice. This paper seeks to design an off-line technique that is comparable in performance to on-line techniques.

3.1. On-line distance measurement

On-line measurement using ping, traceroute, or application specific probing packet is performed whenever distance information is needed [10]. This technique generates information on the fly, upon request, but it creates both a significant amount of traffic overhead and delay. In addition, the retrieved information may be subject to dynamic local conditions, i.e., the RTT information is valid only for the exact time of the measurement and it may reflect local temporary congestion. This may be good if we want to use this type of information, but in most cases the introduced overhead traffic may be high.

For providing the distance information to a large number of clients without significant delay, the IDMAPS [8,6] project that deploys special measuring infrastructures has been proposed. It suggests the use of a set of "tracers" that continually measure the network, and a set of information servers that can provide the requested distance information for applications that might need it. It has been demonstrated [8,6] that such techniques generate much less overhead traffic and are capable of providing fairly accurate information. However, there is a need to create and maintain the special infrastructure of tracers and servers. In addition, monitoring traffic is still generated in the network.

3.2. Off-line distance measurement

To address the issues with the other approaches, a new light-weight technique has been proposed lately. A number of "landmark" servers are placed at well-known points in the network. Each client (server) locates itself according to the relative distances to these landmarks. Several techniques have been proposed to carry out the exact computation [22,4]. Yet, one needs to place and maintain the landmarks, and as pointed out in [4] one should make sure that these servers are capable of handling all the probing traffic. In fact, in [4] the authors estimated that each landmark should be able to handle as much as 2700 pings per second. Several variations of the landmarks based approach has been proposed in recent years [18–20,17,21]. Of these off-line approaches, [21] is the only one that attempts to compute the coordinates without using explicit landmarks. Our approach of using linear regression and the novel metric – DEPTH – is significantly different from it.

3.3. Our approach

We propose a solution that generates very little on-line measurements like in [22,4] but does not require a dedicated infrastructure. It can be deployed very easily and quickly using existing protocols and it provides fairly accurate information as we show in Section 5.

The main idea is that each network element will be associated with local static information. An example of this information is the geographic location, i.e., its latitude and longitude. The main requirements from this information are that it will be static (may change only over several days or weeks) and that it can be made available to each machine. Another critical requirement is that there should be an efficient way to compute the network RTT between two network elements given their static information. We will show later in this paper that such information is indeed available.

Now this information can be made part of the DNS system using the extension to the protocol as defined in

RFC1876 [12] (see [11] for a similar implementation of the RFC known as DNS LOC). Basically, for each host, the DNS server will have a field that represents the geographic location of the host together with its name and IP address(es). This, of course, can be expanded to contain additional local static information as required. For users connecting to the Internet using mechanisms such as dial-up, DSL, ADSL or cable modem, the DHCP (Dynamic Host Configuration Protocol) protocol is typically used for dynamically assigning IP addresses. All IP addresses available to a DHCP server can be associated with a single geographic location. Later (in Section 6), we show that for the problem of closest server selection, the off-line metrics of the client are not needed. Hence, the inaccuracy in geographic location corresponding to some IP addresses will effect the accuracy of server selection only when such users behave as content providers. Moreover, the inaccuracy will typically be limited to a few tens of miles, as most service providers have at least one DHCP server in each city.

There are several possible ways to use this information in order to perform server selection. The host that needs to select a server can retrieve the data related to all the possible servers, compute the distances, and select the best one. Another possibility is that some of the DNS servers will have the ability to perform the computation based on the host information that is provided with the query and send the preferred server name to the host. A third possibility is that the computation will be done by special application servers. For example in the CDN case, this could be a specific server, sometimes called the Network Director, that redirects client requests to the chosen replica. Note that once our approach is adopted, all three schemes can be used and each application can choose its best method to use the data.

3.4. The depth metric

As pointed out before, geographic location information appears to be insufficient to allow accurate enough topology-aware operations such as server selection. We want to define another static variable that can be used to improve the accuracy. To this end, we define the *DEPTH* of a node to be the average RTT from a node to the backbone. In order to make this definition more rigorous we define the set of Class 1 carrier networks, worldwide, to be the Internet backbone. This definition is backed by the recent findings in [23] that defines a set of 20 AS's to be the Internet "dense core". We actually used the networks of the AS's as defined in [23].

Now for each user we calculate the average RTT to this dense Internet core. If the traffic from a host passes through more than one such network, then the DEPTH is defined as the weighted average of the RTT to these networks, where the amount of traffic to each network is the weight. Based on all the minimum RTT traceroutes from the servers, the average depth of the server as well as the average depth of each client is computed. A depth is therefore a single number associated with every host. When computing the network distance, we also refer to the summation of the client and the server depths as the DEPTH, and the use will become clear from the context.

The key idea behind using this variable is that in addition to the geographic distance that clearly effects the RTT, the number of sub-networks in the route from the host to the Internet core plays a significant role in its network distance. A bay-area host that is connected directly to one of the backbones will have a considerably shorter RTT to the East-cost compared to another host, that is located at the same geographic area, but is connected through a 3rd level ISP that is connected through a local ISP to the Internet backbone.

4. Data collection

4.1. Methodology

Since US is highly populated with machines and networks compared to the rest of the world, we wanted to study the US data in isolation. The measurements that we collected between end hosts in the US are collectively referred to as the US-data. However, to get a broader picture and observe relations across the globe, we also collected and studied a second set of data, called the World-data, for which the hosts are spread across the globe outside of US.

We used *traceroute* as our tool for data collection. Several web-servers (see list in [24]) can execute *traceroute* to any machine on the Internet. These web-servers are referred to as servers in the remaining paper. The US-data is based on measurements from 8 servers and the World-data is based on 16 servers. Only those machines, that execute *traceroute* and for which we know the geographic location precisely (at least up to the city level), were picked up as servers. Similarly, the client set used in our study are only those hosts for which the location information[1] is known.

Since our focus in this work is on off-line metrics (as discussed in Section 3), we tried to eliminate the effect of network congestion and network dynamics (such as route updates and failures) from our measurements. All measurements between a server and a client are based on the minimum RTT traceroute from a set of 10 traceroutes performed during various times of the day. Fig. 1 shows how the total summation of minimum RTT changes when a new set of traceroutes is taken into account. For the US-data, we observe that beyond 6th set of traceroutes, the percentage improvement in sum total of all minimum RTT measurements drops below 1%. This shows that additional measurements have insignificant impact on the value of the minimum RTT indicating that the measured value of

[1] Location accuracy is more important for the few servers than the clients.

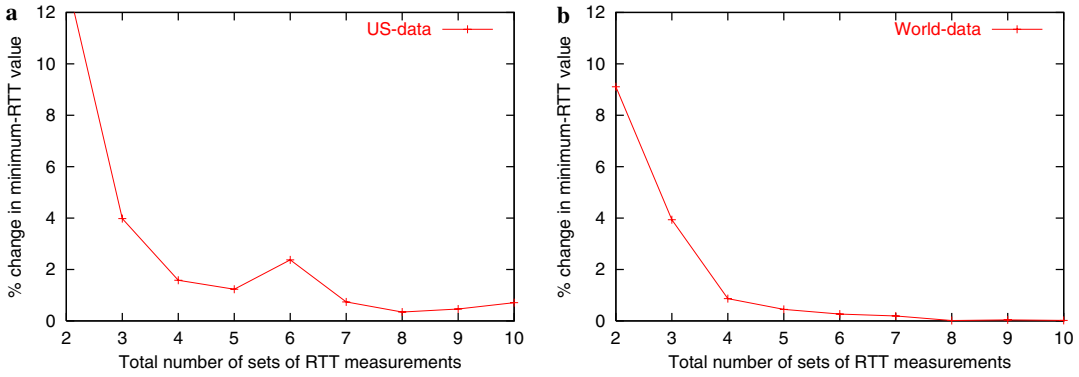

Fig. 1. Percentage change in minimum RTT between all the clients and servers. (a) US-data; (b) World-data.

RTT is highly accurate. For the World-data the sum total of all minimum RTT measurement changes only by less than 1% after the 3rd set of traceroutes. This shows that 10 sets of traceroutes for every client-server pair produces a good approximation for the minimum RTT between them.

To obtain the output of traceroute performed at the servers, we used the *wget* utility on Linux to fetch the corresponding website. The HTML output produced by *wget* is then parsed to obtain the relevant metrics for the study.

Our study is based on both on-line and off-line metrics. We study on-line metrics and combinations of on-line and off-line metrics only for reasons of comparison with off-line metrics.

- *HOPS (On-line)*: It is the number of hops between two nodes, obtained from the output of traceroute.
- *AS (On-line)*: It is the number of Autonomous Systems on the route. The names and the IP addresses of the intermediate routers are available from the output of traceroute. A change in IP network prefix and also a change in the network domain name indicates that the traceroute probe has very likely entered a new AS.
- *DIST (Off-line)*: It is the geographic distance between two nodes while taking the Earth's curvature into account.
- *DEPTH (Off-line)*: The depth for each node (client and server), as defined in Section 3, is computed by processing all the traceroute data.

4.2. Data sets

The details of the World-data set and the US-data set are described below.

- *World-data*: The names and locations of the 16 servers that are used for the World-data are presented in Table 1. The set of 664 clients was a mix of web-servers for universities and libraries from around the world outside US. We use the Class 1 carrier networks [23] as the set of core networks for computing the depth.A quick look at the plot of RTT vs. the geographical distance in Fig. 2(a) reveals three clusters. The first cluster with the lowest distance mostly comprises of measurements from clients and servers in Europe. The second cluster around a geographical distance of 9000 km reflects mostly measurements between hosts in Europe/Australia and Asian countries such as China, Japan, India, etc. The third cluster around 16,000 km is mostly measurements between hosts in Australia and Europe.
- *US-data*: The set of 8 servers that we used for measurements were geographically separated in the US. Their names and the locations are presented in Table 2. A set of 1578 web-servers for US libraries was used as clients [25]. The geographic information was in terms of the name of the city which was converted to latitude and longitude using a conversion tool provided by the US Census Bureau [26]. Once again the Class 1 carrier networks [23] were used as core networks.

Table 1
World-data: names and locations of servers

1.	http://www.mclink.it	Italy
2.	http://www.noc.itgate.net	Italy
3.	http://www.mutugoro.or.jp	Japan
4.	http://www.media-m.co.jp	Japan
5.	http://www.helios.de	Germany
6.	http://www.informatik.rwth-aachen.de	Germany
7.	http://www.cbl.com.au	Australia
8.	http://proxy1.syd.connect.com.au	Australia
9.	http://members.iinet.net.au	Australia
10.	http://ulda.inasan.rssi.ru	Russia
11.	http://www.csc.fi	Finland
12.	http://www.switch.ch	Switzerland
13.	http://www.eye.ch	Switzerland
14.	http://www.belnet.be	Belgium
15.	http://www.ee	Estonia
16.	http://bijt.net	Netherlands

We randomly split the US-data and World-data sets into training and test sets. The training set is used for performing the regression analysis (Section 5). The test set is used to study the performance of closest server selection (Section 6).

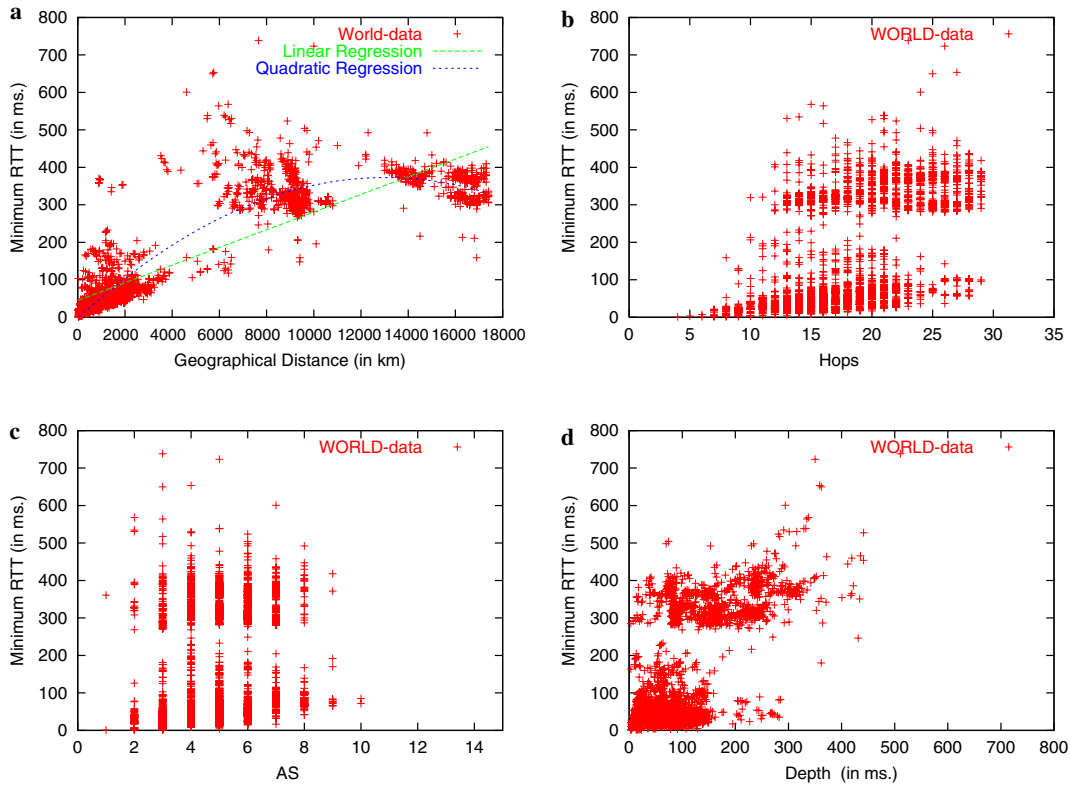

Fig. 2. World-data (Training): scatter plot of minRTT against the four predictors. (a) DIST; (b) HOPS; (c) AS; (d) DEPTH.

Table 2
US-data: names and locations of servers

1.	http://www.telcom.arizona.edu	Arizona
2.	http://www.net.berkeley.edu	California
3.	http://www.sdsc.edu	California
4.	http://www.comnetcom.net	Indiana
5.	http://www.imbris.com	Idaho
6.	http://customers.hispeedhosting.com	New Jersey
7.	http://www.cobaltrack.com	Virginia
8.	http://www.vineyard.net	California

5. Regression analysis

In this section we analyze the training data with the goal of finding a predictive model for minimum RTT. First, we build a model based on DIST, HOPS, and AS as the predictors. Second, we build a model based only on the off-line metrics, i.e., DIST and DEPTH. Then we compare these two models from a statistical aspect. The statistical tool Splus [27] has been used for all the analysis presented in this section. A model is called off-line if it involves off-line metrics only, otherwise it is called an on-line model.

5.1. World-data

A scatter plot of RTT against AS in Fig. 2(c) reveals very little relationship between them, while a scatter plot of RTT against HOPS in Fig. 2(b) shows an increasing relationship. A plot of RTT against DIST in Fig. 2(a) shows an increasing relationship up to 11,000 km (approximately) and a decreasing relationship thereafter. Thus, a quadratic equation on distance seems to be a better fit. However, distance has a very high correlation with RTT (see Table 3), Table 4 and thus even a linear equation on distance will perform well. AS has very small correlation with RTT while the correlation between HOPS and RTT is moderate.

Table 3
Correlation matrix for the World-data (Training)

	HOPS	AS	DEPTH	RTT
DIST	0.4887	0.2656	0.5871	0.8691
HOPS		0.5235	0.3197	0.5569
AS			0.0081	0.2304
DEPTH				0.6486

A number closer to 1 indicates higher correlation.

Table 4
World-data (Training): R^2 values of various models

On-line Models	DIST + HOPS	0.7782
	DIST + HOPS + AS	0.7848
	DIST + DISTSQ + HOPS	**0.8768**
Off-line Models	DIST + DISTSQ	0.8727
	DIST + DEPTH	0.7845
	DIST + DISTSQ + DEPTH	**0.8841**

For performing linear multi-variable regression, it is critical to ensure that the metrics are not highly correlated with each other. All outliers have been removed using Cook's distance measure. To check the presence of *multicollinearity* among the predictors, we computed the *condition number*, which is the square root of the ratio of the largest Eigen-value to the smallest Eigen-value of the correlation matrix of the predictors. A *condition number* larger than 15 is considered to be an indication of presence of *multicollinearity*. The corresponding number here is found to be equal to 2.4422, which is small enough to suggest the absence of any *multicollinearity*. Thus, Akaike Information Criteria (AIC) [28], the likelihood version of Mallows C_p statistics, may be used for model selection.

First, a linear regression model is fitted with DIST, HOPS, and AS as the predictors. Then the function 'step' in 'Splus' is applied to obtain the AIC statistics. The R^2 statistic indicates the fraction of variation that can be explained by a given prediction model. By removing AS, the AIC statistics increased from 12,575,848 to 12,946,555, and the multiple R^2 dropped from 0.7848 to 0.7782 (see Fig. 4). Though an increase in the AIC statistics suggests that the variable being removed from the model is

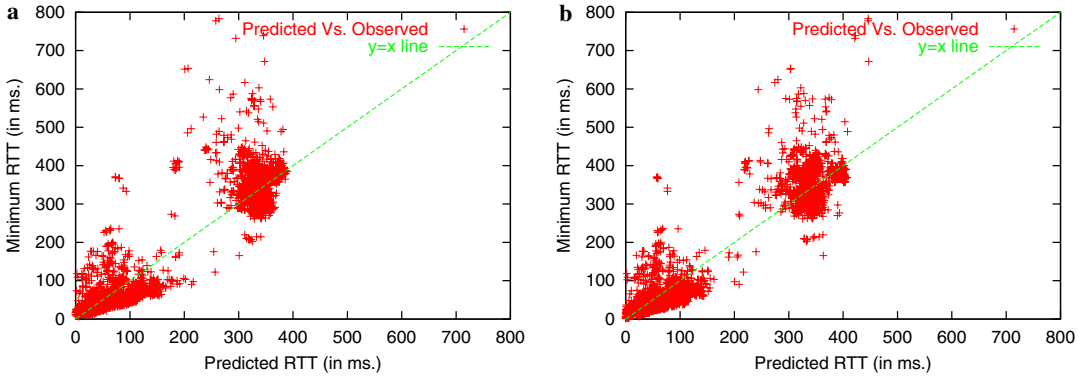

Fig. 3. World-data (Training): minRTT against the predicted RTT from the two competing models. DISTSQ refers to a square/quadratic term in distance. (a) DIST + DISTSQ + HOPS; (b) DIST + DISTSQ + DEPTH.

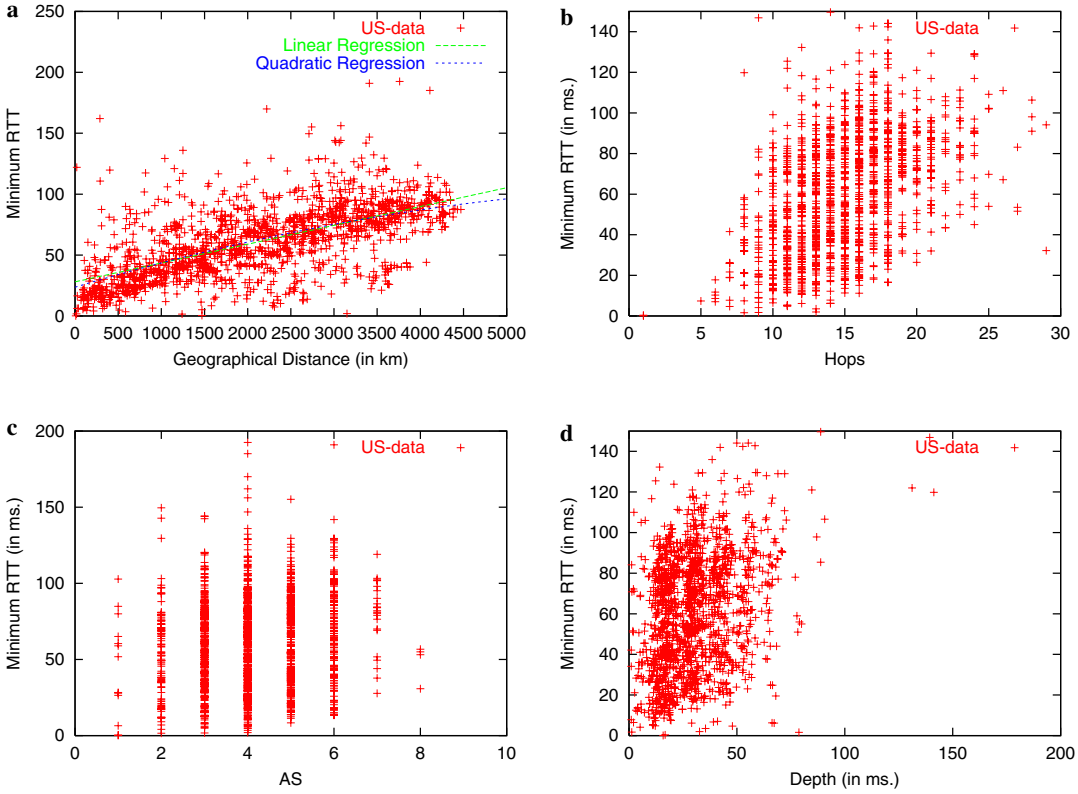

Fig. 4. US-data (Training): scatter plot of minRTT against the four predictors. (a) DIST; (b) HOPS; (c) AS; (d) DEPTH.

important, but the amount of change in the AIC statistics and the R^2 values indicate that by removing this predictor, we lose very little. On the other hand, the estimation error in the coefficient of this variable may inflate the prediction error for RTT. Running the function 'step' on the reduced models, the AIC values indicate that no more variables should be removed. Now we add a quadratic term on distance to this model and the R^2 value increased significantly to 0.8768. Thus, our first model is to regress RTT with a quadratic term in distance and a linear term in hops.

Now we build the best off-line model using the metrics DIST and DEPTH. The scatter plot of RTT against DEPTH in Fig. 2(d) shows an increasing linear relationship. A linear regression on DIST and DEPTH produces an R^2 value of 0.7845. Running the function 'step' on this model, the AIC statistics indicates that none of the variables should be removed. Adding a quadratic term in distance to this model improves the R^2 value to 0.8841. Thus our final off-line model is to regress RTT on DIST with a quadratic term, and DEPTH. Fig. 3 shows that both of the above models provide good prediction accuracy, although the R^2 values indicate that DIST and DEPTH (quadratic) performs slightly better.

Table 5
Correlation matrix for the US-data (Training) A number closer to 1 indicates higher correlation

	HOPS	AS	DEPTH	RTT
DIST	0.3023	0.1137	0.0648	0.6113
HOPS		0.3064	0.2462	0.4539
AS			0.1329	0.1285
DEPTH				0.3658

Table 6
US-data (Training): R^2 values of various models

On-line Metrics	DIST + HOPS	0.4534
	DIST + HOPS + AS	0.4539
	DIST + DISTSQ + HOPS	**0.4559**
Off-line Metrics	DIST + DISTSQ	0.3772
	DIST + DEPTH	0.4805
	DIST + DISTSQ + DEPTH	**0.4851**

5.2. US-data

In the US-data, the scatter plots of RTT against DIST, HOPS and DEPTH (Figs. 4(a), (b), and (d)) reveal a linear relationship with RTT. No visual relation is found with AS. The correlation matrix (Table 5) Table 6 suggests that DIST is the best predictor for RTT, while AS has very little influence on RTT. First, we consider only three predictors – DIST, HOPS, and AS. The correlations between these predictors are not big enough to cause any multicollinearity. By removing the predictor AS, the AIC statistics increased from 680,182 to 679,864, and the multiple R^2 dropped from 0.4539 to 0.4534 (see Fig. 6). The small decrease in multiple correlation and an increase in the AIC clearly suggest that AS should be removed from the model. Running the function 'step' on the reduced models, the AIC values indicate that no more variables should be removed. Now we replace the predictor HOPS by the predictor DEPTH, and the R^2 value increased to 0.4805.

To be consistent with the world data, we add a quadratic term in distance. This improves the R^2 value marginally for both the models – (1) distance with hops, and (2) distance with depth, indicating that the quadratic term is not that important for the US-data. However, we kept the quadratic term in our final models, as suggested by the AIC statistics and to be consistent with the world data.

The regression equation based on a quadratic in DIST and a linear term in HOPS explains 45.6% of the variance in RTT ($R^2 = 0.4559$), while the regression equation based on a quadratic term in distance and a linear term in depth explains 48.5% of the variance in RTT ($R^2 = 0.4851$). The observed vs. predicted plots for both the models (Fig. 5) show that the two models perform alike. Thus, through completely off-line metrics we achieve a comparable RTT prediction model.

A linear term in distance is adequate enough for the US-data, while we require a quadratic term in the same variable for the World-data. However, it will be misleading to conclude that the relationship of RTT on distance is different for the two cases. A closer investigation of the estimated quadratic function of the distance in the World-data reveals that the change in the gradient is prominent when

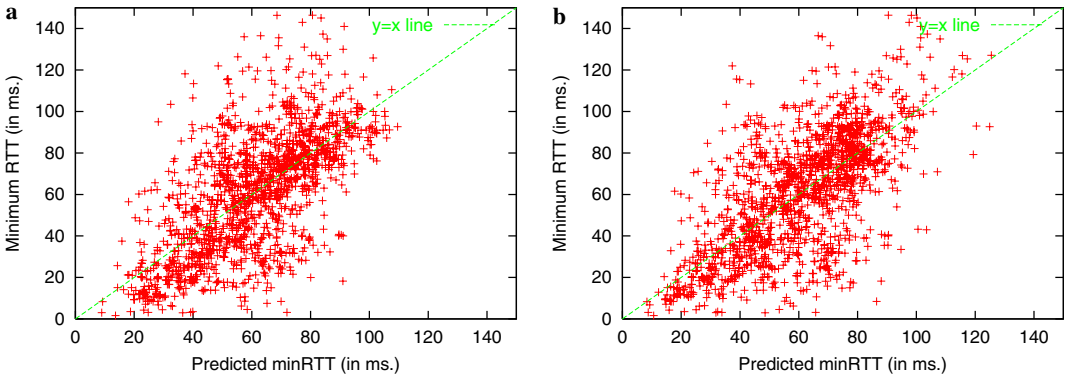

Fig. 5. US-data (Training): minRTT against the predicted RTT from the two competing models. (a) DIST + DISTSQ + HOPS; (b) DIST + DISTSQ + DEPTH.

the distance is around 12,000 km. The function is almost linear for small values of distances. All the distances in the US data are less than 4500 km and thus the effect of the quadratic part, if any, is unobservable. Hence, distance (in quadratic form) and depth are found to be reasonably good predictors for RTT and the models are found to be consistent for both the data sets.

6. Closest server selection

In the previous section we analyzed the data statistically to come up with a predictive model for RTT. Based on the regression results, we found that DEPTH along with DIST (quadratic) is the best off-line model for distance estimation. Now we want to evaluate the quality of these models with respect to the "goodness" of the topology-aware operations, particularly the closest server selection. Here, we are interested in identifying the closest server from a given set of servers without taking into account the dynamic variables such as network congestion and server load. We also seek a solution which will compute the closest server quickly with minimum extra traffic.

Our algorithm works as follows. Given a client and a set of possible servers, each with its location and depth information, we estimate RTT from the client to each of the servers by first computing the geographic distance from the location information and the total depth (obtained by adding the client depth with the server depth), and then plugging in these two numbers into the corresponding regression equation. Then we choose the server with the smallest estimated RTT. Since the regression equations are linear in the total depth, there is no need to use the client depth, as it adds a constant to the network distances of the servers from this particular client. Therefore, for our server selection algorithm, the only information needed is the client's geographical location and both the geographical location and depth of the servers. This makes our approach highly scalable, as only the servers need to compute the depth. In addition, as the DEPTH of the clients is not needed for server selection, this approach easily applies to mobile clients as well.

In our collected traceroute data, we observed that several clients were not able to reach all the servers. The accuracy results that we present here only take those clients into account that reached all the 16 servers in the case of the World-data and all the 8 servers in the case of the US-data. In addition, the data set used in this evaluation was obtained from the test set (not the training set that was used to derive the regression models). For the World-data-test set (Fig. 6), 132 clients were able to reach all the 16 servers and we have used only those clients for computing the accuracy of server selection. For the US-data-test set (Fig. 7), the corresponding number is 150 clients.

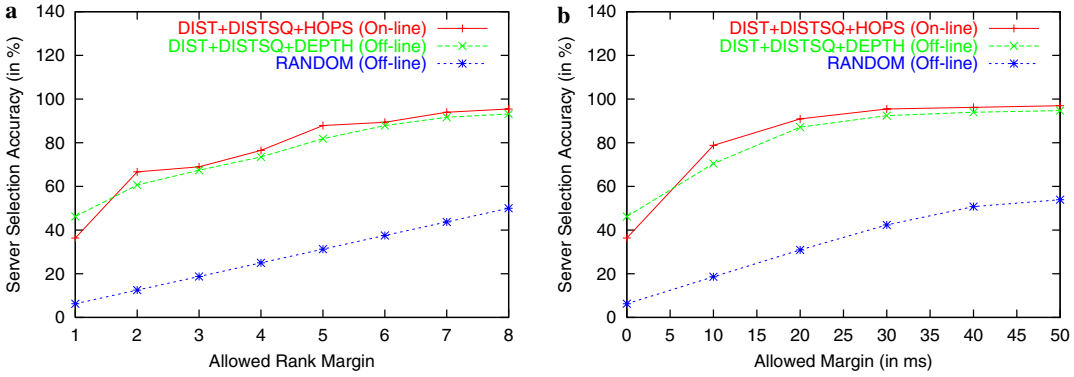

Fig. 6. World-data (Test): deviation of RTT of chosen server from the closest server. (a) Accuracy vs. allowed rank deviation; (b) accuracy vs. allowed delay deviation.

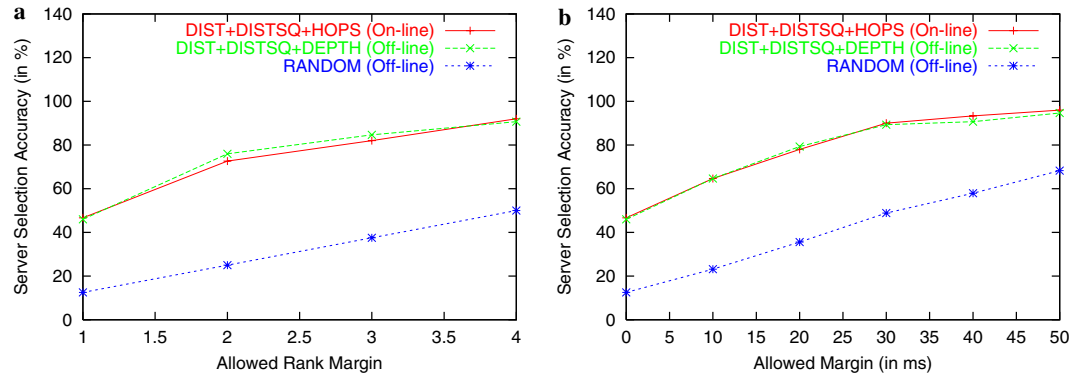

Fig. 7. US-data (Test): deviation of RTT of chosen server from the closest server. (a) Accuracy vs. allowed rank deviation; (b) accuracy vs. allowed delay deviation.

Following recent approach of forgoing a small amount of deviation of RTT [4], we explore the accuracy of server selection for different allowable deviations from the optimal choice. We also study deviation from optimal in terms of ordered ranking of servers. Note that for purposes of server selection there is no difference between DIST and DIST + DISTSQ, as both the metrics use the same independent variable and both are monotones. We use the model derived from the training half of US-data (World-data) to study performance on the test half of the US-data (World-data). We have observed that the model of World-data can also be applied to the US-data to obtain similar results.[2]

The most striking observations from Fig. 6 and Fig. 7 are threefold. First, the accuracy of using the best off-line model (DIST + DISTSQ + DEPTH) for choosing the closest server is very high, 90% accuracy for an allowed deviation of 30 ms. Second, the best off-line model performs much better than the RANDOM approach, where the server is chosen randomly. Third, the accuracy of the best off-line model is not significantly different from the best on-line model, which requires higher probing overhead.

7. Conclusions

In this paper, we studied the ability to perform accurate topology-aware operations such as server selection using only static information. Our study was based on four metrics, namely Geographic distance, Number of hops, Number of Autonomous Systems (AS), and Depth, out of which only the Geographic distance and Depth are off-line parameters. The other two were included for the sake of comparison. Based on our detailed study of various combinations of these parameters using regression analysis, we found that Depth along with geographic distance in quadratic form is the best off-line model for distance estimation. From the study on server selection we observe that our off-line approach performs as well as the best on-line approach. Thus by deploying our distance estimation technique in the Internet, low overhead and low latency distance estimation can be readily performed with high accuracy. We have shown that with support from DNS system for including location information, the Depth and location measurements can be readily made available to clients for enabling fast distance estimations.

One of the limitations of our research deals with the need to measure the depth of the server using traceroute. This needs to be repeated whenever there is significant change in traffic which could be several days to weeks. We are exploring solutions to further reduce the frequency of traceroutes and trying to find alternate techniques for Depth measurement as some routers may not support traceroute queries. Our proposed approach requires support from the DNS system for maintaining location and Depth information of the servers. One of the challenges that has not been addressed in the present work is related to mobile servers. For such servers computing Depth and updating the DNS periodically requires novel approaches that are being explored. Clearly, in some cases there might be a need for higher levels of accuracy than the one provided by the off-line scheme. In such cases one can incorporate our methods along with on-line, or semi on-line techniques, achieving better server selection at the cost of increased network overhead. The off-line approaches [18–21] based on the idea of landmarks [17] has been shown to reduce the number of dimensions (or metrics) needed for estimating the RTT, without losing any accuracy. We are investigating techniques for combining our approach with other such off-line approaches for improving the overall performance.

[2] Not included in this paper due to lack of space.

References

[1] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl, Globally distributed content delivery, in: Proceedings of IEEE Internet Computing, Sept. 2002, pp. 50–58.
[2] B. Krishnamurthy, C. Willis, Y. Zhang, On the use and performance of Content Distribution Networks, in: Proceedings of the 1st ACM Workshop on Internet Measurement, Nov. 2001, pp. 169–182.
[3] C. Gkantsidis, P. Rodriguez, Network coding for large scale content distribution, in: Proceedings of INFOCOM, vol. 4, Miami, Mar. 2005, pp. 2235–2245.
[4] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, Topologically-aware overlay construction and server selection, in: Proceedings of INFOCOM, New York, NY, June 2002, pp. 1190–1199.
[5] P. Francis, S. Jamin, V. Paxson, L. Zhang, D.F. Gryniewicz, Y. Jin, An architecture for a global internet host distance estimation service, in: Proceedings of INFOCOM, New York, NY, Mar. 1999, pp. 210–217.
[6] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, L. Zhang, IDMaps: a global internet host distance estimation service, IEEE/ACM Transactions on Networking 9 (5) (2001) 525–540.
[7] S. Srinivasan, E. Zegura, Network Measurement as a Cooperative Enterprise, Lecture Notes In Computer Science, 2002 166–177.
[8] E. Cronin, S. Jamin, C. Jin, A.R. Kurc, D. Raz, Y. Shavitt, Constrained mirror placement on the internet, IEEE Journal on Selected Areas in Communications 20 (7) (2002) 1369–1382.
[9] T.S.E. Ng, H. Zhang, Towards global network positioning, in: ACM SIGCOMM Internet Measurement Workshop, Academic Press, New York, 2001, pp. 25–29.
[10] R.L. Carter, M. Crovella, Server selection using dynamic path characterization in wide-area networks, in: INFOCOM, Kobe, Japan, Apr. 1997, pp. 1014–1021.
[11] C. Davis, H. Rose, DNS LOC: geo-enabling the domain name system, <http://www.ckdhr.com/dns-loc/>.
[12] C. Davis, P. Vixie, T. Goodwin, I. Dickinson, A means for expressing location information in the domain name system, RFC 1876, Jan. 2001.
[13] Skitter, Tool for topology and performance analysis for the internet, <http://www.caida.org/tools/measurement/skitter/>.
[14] B. Huffaker, M. Fomenkov, D. Moore, k. claffy, Macroscopic Analyses of the infrastructure: measurement and visualization of internet connectivity and performance, in: Proceedings of PAM (A Workshop on Passive and Active Measurements), Amsterdam, Netherlands, Apr. 2001.

[15] B. Huffaker, D. Plummer, D. Moore, k. claffy, Topology discovery by active probing, in: Proceedings of Symposium on Applications and the Internet (SAINT), Nara, Japan, Jan. 2002, pp. 90–96.
[16] B. Huffaker, M. Fomenkov, D. Plummer, D. Moore, k. claffy, Distance metrics in the internet, in: Proceedings of IEEE International Telecommunications Symposium (ITS), Natal RN, Brazil, Sept. 2002.
[17] L. Tang, M. Crovella, Virtual landmarks for the internet, in: Proceedings of IMC, Miami Beach, FL, Oct. 2003, pp. 143–152.
[18] H. Lim, J.C. Hou, C.-H. Choi, Constructing internet coordinate system based on delay measurement, in: Proceedings of IMC, Miami Beach, Florida, Oct. 2003, pp. 129–142.
[19] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, S. Bhatti, Lighthouses for scalable distributed location, Lecture Notes in Computer Science, Peer-to-Peer Systems II: Second International Workshop, IPTPS, vol. 2735, Feb. 2003, pp. 278–291.
[20] M. Waldvogel, R. Rinaldi, Efficient topology-aware overlay network, SIGCOMM Computer Communication Review 33 (1) (2003) 101–106.
[21] M. Costa, M. Castro, A. Rowstron, P. Key, PIC: practical internet coordinates for distance estimation, in: Proceedings of ICDCS, Tokyo, Japan, Mar. 2004, pp. 178–187.
[22] T.S.E. Ng, H. Zhang, Predicting internet network distance with coordinates-based approaches, in: Proceedings of INFOCOM, New York, NY, June 2002, pp. 170–179.
[23] L. Subramanian, S. Agarwal, J. Rexford, R.H. Katz, Characterizing the internet hierarchy from multiple vantage points, in: Proceedings of INFOCOM, New York, NY, June 2002, pp. 618–627.
[24] List of Traceroute Sites, Traceroute.org, <http://www.traceroute.org/>.
[25] Skitter, Tool for topology and performance analysis for the internet, <http://sunsite.berkeley.edu/>.
[26] US Census Bureau, Data tools, <http://www.census.gov/>.
[27] Splus, Tool for exploratory data analysis and statistical modeling: Splus, <http://www.splus.com/>.
[28] K.P. Burnham, D.R. Anderson, Model selection and inference: a practical information-theoretic approach, Springer, New York, 1998.

Prasun Sinha received his PhD from University of Illinois, Urbana-Champaign in 2001, MS from Michigan State University in 1997, and B. Tech. from IIT Delhi in 1995. He worked at Bell Labs, Lucent Technologies as a Member of Technical Staff from 2001 to 2003. Since 2003 he is an Assistant Professor in Department of Computer Science and Engineering at Ohio State University. His research focuses on design of network protocols for sensor networks and mesh networks. He served on the program committees of various conferences including INFOCOM (2004-2006) and MOBICOM (2004-2005). He has won several awards including Ray Ozzie Fellowship (UIUC, 2000), Mavis Memorial Scholarship (UIUC, 1999), and Distinguished Academic Achievement Award (MSU, 1997). He received the prestigious NSF CAREER award in 2006.

Danny Raz received his doctoral degree from the Weizmann Institute of Science, Israel, in 1995. From 1995 to 1997 he was a post-doctoral fellow at the International Computer Science Institute, (ICSI) Berkeley, CA, and a visiting lecturer at the University of California, Berkeley. Between 1997 and 2001 he was a Member of Technical Staff at the Networking Research Laboratory at Bell Labs, Lucent Technologies. In October 2000, Danny Raz joined the faculty of the computer science department at the Technion, Israel. His primary research interest is the theory and application of management related problems in IP networks. Danny Raz served as the general chair of OpenArch 2000, and as a TPC member for many conferences including INFOCOM 2002-2003, OpenArch 2000-2001-2003, IM-NOMS 2001-2005, and as an Editor of the IEEE/ACM Transactions on Networking (ToN).

Nidhan Choudhuri received his PhD from Michigan State University, his M. Stat. and B. Stat. degrees from Indian Statistical Institute, Calcutta, India. He was a visiting faculty at University of Michigan Ann Arbor from 1999 to 2000. Since 2000 he has been an assistant professor at Case Western Reserve University. His areas of interest are nonparametric function estimation, Bayesian methods, empirical likelihood, and statistical computing.