Discovering Network Neighborhoods Using Peer-to-Peer Lookups

Li-wei Lehman and Steven Lerman
Center for Educational Computing Initiatives
Massachusetts Institute of Technology

Abstract— **In many distributed applications, end hosts need to know the network locations of other nearby participating hosts in order to enhance overall performance. Potential applications that can benefit from the location information include automatic selection of nearby Web servers, proximity routing in a peer-to-peer system, and loss recovery in reliable multicasting.**

We focus in this paper on the network neighborhood discovery problem in large-scale distributed systems. In these systems, the number of participating nodes can be very large, and the membership can dynamically change. Our goal is for each node to discover other "nearby" participating nodes in a completely decentralized manner, where each node probes only a small subset of other nodes in the system. This approach will lead to improved overall performance by matching client requests for services with participants in the peer-to-peer service system that are, on average, nearby in the network sense.

Recent works in distributed peer-to-peer systems, such as Chord, CAN, Tapestry and Pastry, provide efficient distributed lookup structures. In this paper, we investigate a rendezvous-based scheme for a node to discover other nearby participating nodes using a peer-to-peer lookup system such as Chord. Given a key, the Chord protocol maps the key onto a node. Our idea for network neighborhood discovery is for each host to compute a key that characterizes its network location on the Internet. We call such a key the location key, and the nodes that these location keys are mapped to the Rendezvous Points. To lookup other nearby participating nodes, a node seeking some service queries its corresponding rendezvous point using its location key.

We focus on the issue of how to generate the location key in a distributed fashion such that nodes that are close to each other in the actual network will have similar location key values, and therefore be mapped to nearby locations on the Chord ring. In this paper, we examine the performance tradeoffs of such a rendezvous scheme using the Global Network Positioning (GNP) approach to generate the location keys. In GNP, each node measures its network distances to a few landmark nodes to derive its coordinates in a D-dimensional geometric space. We generate a host's Chord location key from its 1-dimensional GNP coordinate, and use coordinates from a higher dimensional space to refine the searching process for the closest node. We evaluate our scheme in the context of the nearest neighbor discovery problem. Using data from the Active Measurement Project of the National Laboratory for Applied Network Research (NLANR), we compare its performance with a random mapping scheme, where location keys are randomly generated. Using our coordinate-based rendezvous scheme, 66% of the nodes found their actual closest network neighbor by pinging only a small number of nodes.

I. Introduction

In many distributed applications, end hosts need to know the network locations of other nearby participating hosts in order to enhance their performance. Example applications include automatic selection of nearby Web server, proximity routing in a peer-to-peer (P2P) system, and loss recovery in reliable multicasting. In this paper, we focus on the network neighborhood discovery problem in large-scale distributed peer-to-peer systems. There are several challenges to the network neighborhood discovery problem in these systems.

- The first challenge is to measure and estimate the network locations of nodes. Although the most accurate way to discover nearby nodes is for each node to explicitly measure the round-trip latency to every other node in the system, this scheme does not scale for systems with hundreds and thousands of nodes. Besides, this approach requires each node to have complete membership information, which is impractical to maintain in a large, dynamic peer-to-peer system. Therefore, we need a scheme to estimate and characterize the network location of a node by using only a small number of round-trip time measurements.
- Secondly, once we have an estimate of a node's relative network location, we need a compact representation of the location information so that nodes can exchange the locality information efficiently.
- The third issue is the distributed membership discovery problem; after we have a mechanism to estimate, characterize, and represent the network locations of nodes, how can nodes discover other nodes in the same network neighborhood? One way is to use a centralized directory lookup system, where all nodes deposit their location information at a centralized server. The disadvantage of this scheme is that the centralized server becomes the bottleneck and the single point of failure in the system. Another common way for membership discovery in a distributed environment is for members to use a gossip-like protocol[1]. While gossiping can be a robust way to discover membership information, it does not localize the information discovery process to only relevant nodes.

Our goal is for each node to discover other "nearby" participating nodes in a completely decentralized manner, where each node probes only a small subset of other members in the system. We use round-trip latency as a measurement of nearness between network hosts.

In this paper, we introduce a rendezvous-based scheme for distributed topology discovery using a peer-to-peer lookup

system such as Chord[2]. Given a key, the Chord protocol maps the key onto a node. Our approach is for each host to compute a key which characterizes its network location on the Internet. We call such a key the location key, and the nodes that these location keys are mapped to the Rendezvous Points (RPs). We use a coordinates-based approach to characterize a host's location on the Internet. In particular, we use the GNP (Global Network Positioning)[3] scheme to derive some D-dimensional coordinates for each host. To lookup other nearby participating nodes, a node seeking some service queries its corresponding Rendezvous Point using its location key.

One issue is how one can map Internet network locations to Chord identifier space such that nodes that are close to each other in the actual network will be mapped to similar key values in the Chord identifier space. On one hand, we would like to exploit the natural clustering among nodes so that nodes that are close to each other are mapped to the same rendezvous point. On the other hand, the location keys should not be clustered too tightly in order to avoid overloading any rendezvous points.

II. Related Work

Rendezvous-based communication is not a new idea. IP multicast and its later variants [4], [5] are examples of how senders and subscribers of a multicast group can use routers in the network as rendezvous points. Recently developed distributed peer-to-peer lookup structures, such as the original Plaxton scheme[6], Chord[2], Pastry[7], Tapestry[8], and CAN[9] are all well-suited to implement such rendezvous-based communications. Applications of these distributed lookup schemes include CFS[10], which is a distributed file system built on top of Chord; SCRIBE[11], which is a topic-based publish-subscribe infrastructure for content distribution using Pastry; and Bayeux[12], which is an application level multicast system built on top of Tapestry.

More recently, the Internet Indirection Infrastructure (I3)[13] presents a new Internet rendezvous-based communication architecture using Chord. The I3 infrastructure is designed to provide solutions for mobility, anycast and multicast on the Internet. In [13], I3 was examined in several application contexts, including server selection, service composition (e.g, applying transcoding in the network), heterogeneous multicast, and large-scale multicast. Its server selection application most resembles our work in that it also uses Chord as a way for nodes to discover the nearby servers. The difference is that I3's focus is on flexible and application-specific routing using something called "triggers" as Chord keys. It did not focus on the network neighborhood discovery problem.

Several peer-to-peer routing schemes implement some form of closest neighbor search in order to support proximity routing. In [14], Castro *et al.* presented a comprehensive study of network locality properties in Pastry. Upon joining, a node in Pastry needs to lookup a nearby seed node in order to initialize its routing tables to satisfy certain proximity requirements. To discover a nearby neighbor, a node performs a series of probes of other nodes' proximity tables. At each iteration, a node picks the nearest node from the other node's proximity table and uses that node as the probing target in the next iteration. The probing stops when no more "progress" is made in terms of the network distances. Castro *et al.'s* simulation results show that the scheme takes a large number of probes to discover the nearby node, since they start probing from a randomly selected node in the system.

In CAN, proximity routing is supported by having nodes that are close-by in network space map to nearby CAN key space. Similar to our scheme, they assume the existence of a set of landmarks in the Internet. They use some form of "distributed binning" scheme to allocate nodes into CAN key space based on their relative distances to those landmarks. This distributed binning scheme is later explored in detail in [15].

There are two major differences between their work and ours. First, in the distributed binning scheme, the location of an end host i is characterized by the ordering of landmarks in terms of their distances to i. The CAN key space is then divided into regions based on the $k!$ different orderings of the k landmarks. A node is randomly mapped to a point within the region in CAN key space (a d-dimensional Cartesian space) with corresponding landmark ordering. The potential drawback of their scheme is that the mapping depends on a fixed k. Changing the number of landmarks would change the way the key space is divided into regions, which will in turn change where the nodes are mapped to in the key space. In contrast, our scheme is insensitive to the exact number of landmarks used as long as the landmarks provide a consistent set of reference coordinates to other nodes. Secondly, their work did not focus on using the P2P lookup structure in the context of the distributed topology discovery problem. As a result, they did not address issues such as distributions of the bin size.

III. Rendezvous-Based Topology Discovery Using Structured P2P Systems

In this section, we describe a rendezvous-based scheme for distributed topology discovery. As mentioned earlier, we use GNP[3] to compute the coordinates of each host in a D-dimensional geometric space to characterize its network location. In GNP, a set of infrastructure nodes, called landmarks, are used to provide reference coordinates. Each host measures the distance between itself and the N landmarks to derive its coordinates in a D-dimensional geometric space. The GNP system computes the coordinates of a node by minimizing some error measurement function (which can be the simple squared error or more sophisticated error measurement function). As the GNP paper by Ng *et al.*[3] has shown, the coordinates-based system is an effective and compact way to represent a node's location relative to others.

The question next is how one can map GNP coordinates to Chord id space such that nodes that are close-by in terms of the actual network latency can discover each other quickly through the Chord lookup mechanism. Our approach is to simply compute the 1-D coordinate for each node based on its distances to the N landmarks, and then linearly scale the

1-D coordinate to the Chord identifier space. The scaled 1-D coordinate then becomes the key for looking up other nodes using Chord. After nodes rendezvous with each other using their coordinates-based location key, they can use coordinates with higher dimensions to refine their searches. More details of this mapping scheme will be presented in the next section.

A. Coordinates-based mapping scheme

For simplicity, we assume that all nodes in our peer to peer system maintain a Chord lookup structure amongst themselves. Each node is assigned a random identifier. Identifiers are ordered in an identifier circle modulo 2^m as in Chord.

Note that the above assumption is not required to guarantee the correctness of the scheme. Our topology discovery scheme does not require all nodes in a peer-to-peer system to maintain a Chord look up structure amongst themselves. In fact, the nodes which are querying their respective network neighborhoods can be a completely different and independent set of nodes from the Chord nodes. For example, a rendezvous service can be provided by a set of distributed, dedicated nodes running Chord. In this case, a querying node simply needs to know the identities of the landmark nodes for deriving its coordinates, and the identity of any Chord node to do the topology lookup.

Let L be the number of landmarks in the infrastructure. Each node will measure its RTT to each of the L landmarks. Using the L RTT measurements, each node then generates its D-dimensional coordinates, and a 1-dimensional coordinate. The 1-dimensional coordinate value is linearly scaled to the Chord identifier space. Each node i then uses its scaled 1-D coordinate value, V_i as a Chord key to look up its rendezvous point. Since in a distributed environment, the actual range of the 1-D coordinate value is unknown, each node needs to use a pre-defined, estimated coordinate range to map its 1-D coordinate value to the Chord key space.

The successor node of key V_i on the Chord identifier circle is the Rendezvous Point (RP) of node i. Node i sends a message, containing its D-dimensional coordinate values, to its RP to look up other nearby nodes. Each rendezvous point maintains a "pool" of nodes that are rendezvousing at itself. The "pool size" of a rendezvous point refers to the number of nodes that are rendezvousing at that RP.

We assume that each RP periodically exchanges its "pool" list with its immediate successor and predecessor on the Chord ring. The combined "pool" list is then called the "augmented pool" list. In response to each node i's query for i's nearest network neighbor, an RP sorts its augmented pool list based on each node's Euclidean distance in the D-dimensional space to node i. It returns the closest k nodes in the sorted list to node i as a response to its query. To complete the search process, node i then pings each of the k nodes returned by its RP, and declares the one with the closest RTT measurement its closest network neighbor.

IV. Experimental Results

In this section, we give a preliminary evaluation of the above coordinate-based rendezvous scheme for discovering nearby nodes. We compare our coordinate-based mapping with a randomized mapping scheme and show that the coordinate-based scheme is much more effective in locating nearby nodes. The randomized mapping scheme works the same way as the coordinate-based scheme, except that each node, instead of using the 1-D coordinate as a key to locate its RP, uses a randomly generated key to locate its RP.

The Active Measurement Project (AMP) at the National Laboratory for Applied Network Research (NLANR) collects network measurements between over 100 active monitors distributed over the Internet[16]. We use the round trip time (RTT) measurements between 110 of such monitors on July 16, 2002 for our experiments. The RTTs are the round trip ping time between each pair of hosts measured at a frequency of once every minute over a 24 hour period (i.e., total of 1440 round trip times reported between each pair of hosts). We take the average RTT over the 24 hour period between each pair of hosts, and generated a 110x110 RTT matrix to represent the pairwise distances between hosts.

To make minimum assumptions about the landmark placement, we randomly chose 10 nodes from the 110 nodes as the landmarks for each experiment. Ten experiments in total were performed, each with a different random selection of the 10 landmarks.

We use the RTTs between each host and the 10 randomly selected landmarks as input to GNP, and generated two sets of coordinates for each host: coordinates in 5-Dimensional and 1-Dimensional space. We use the simplex downhill implementation in GNP to compute the coordinates to minimize the following normalized squared error function:

$$\epsilon(d_{ij}, D_{ij}) = (\frac{d_{ij} - D_{ij}}{d_{ij}})^2$$

where d_{ij} is the actual distance measurement between host i and j, and D_{ij} is the computed distance (Euclidean in our case) between host i and j using the coordinates. The initial range we use to the simplex downhill method for locating the minimum in the above error function is -250 to 250. We use the same range for scaling the 1-D coordinates to the Chord identifier space. For ease of implementation, we chose the identifier space to be in the range of 0 to $2^{16} - 1$.

We examine our rendezvous algorithm in terms of 3 performance metrics: distance ratios, poolsize, and augmented poolsize. The distance ratio R_i of a node i is defined as, $R_i = \frac{RTT_f}{RTT_a}$, where RTT_f is the RTT measured between node i and its closest found node, and RTT_a is the RTT measured between node i and its actual closest network neighbor in the system.

A. Distance Ratio Comparison

In this section, we examine the distance ratios of both the coordinate-based and the random mapping schemes. We declare a node to have found its actual closest neighbor if the distance ratio is < 1.0001. The results presented here

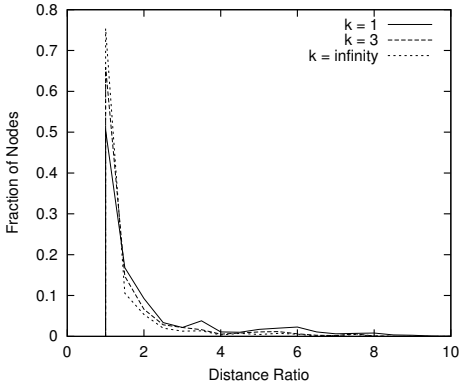

Fig. 1. Coordinates-Based Mapping Distance Ratio Distribution, k = 1,3, and infinity.

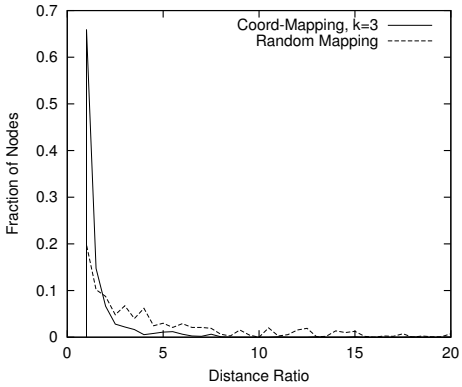

Fig. 2. Distance Ratio Distribution: Random vs. Coordinates-Based Mapping, k = 3.

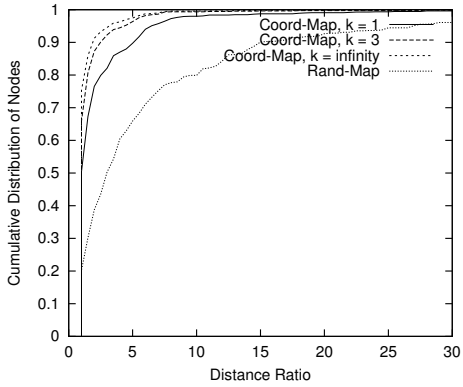

Fig. 3. Distance Ratio Cumulative Distribution: Random vs. Coordinates-based mapping

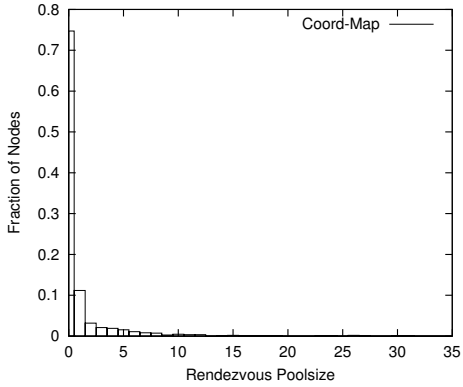

Fig. 4. Coordinates-Based Mapping Poolsize Distribution

are combined distributions of distance ratios from all 10 experiments (with different landmark selections).

We examine the distance ratios of the coordinates-based mapping scheme when k = 1, 3 and infinity. Recall that, in response to a node i's query to its closest node, an RP will return the k closest nodes to i in its augmented pool. When k = infinity, the RP returns the entire augmented pool list to node i, which in turn pings each node to locate the closest neighbor.

From Figures 1 and 3, we note that a node derives most of the benefits by pinging only the 3 closest nodes (in terms of 5-dimensional Euclidean distance) in its RP's augmented pool. Figure 2 and 3 show that the coordinate-based scheme significantly outperforms the randomized-mapping scheme. For example, from figure 3, using the coordinate mapping algorithm with a scaling range in (-250, 250), 66% of the nodes found their actual closest neighbor, and 80% of the nodes yield distance ratios less than 1.5 by pinging the closest 3 nodes from its RP augmented pool. In comparison, using the randomized mapping scheme, less than 20% of the nodes found their actual closest neighbor, and 30% of the nodes yield distance ratios less than 1.5.

B. Poolsize Distribution

Next we examine the poolsize distribution, i.e., for each node, we take the number of nodes rendezvousing at that node, and plot the overall distributions from the 10 experiments. This section presents the RP's pool size before augmenting with its Chord neighbors' pool list. In the next section, we present the distribution for the augmented pool.

As expected, the coordinates-based mapping scheme has more skewed poolsize distribution because the nodes are clustered in the actual network space. Before augmenting their pools, about 75% of the nodes have no nodes rendezvous at them. In contrast, 50% of the nodes in the random mapping scheme have no nodes mapped to them.

C. Augmented RP Poolsize Distribution

Finally, we examine the augmented RP poolsize distribution. For each node i in the system, we record the augmented pool size of the RP of node i and plot the overall distribution from 10 experiments.

Note that since we are taking the distribution from only RP nodes, there are no nodes with augmented poolsize of 0. The randomized mapping shows that about 16 percent of the nodes rendezvous at RPs have only 2 nodes in its augmented pool. In the coordinate-based mapping scheme, the

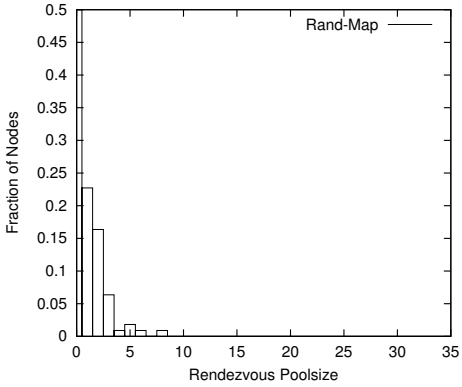

Fig. 5. Random Mapping Poolsize Distribution

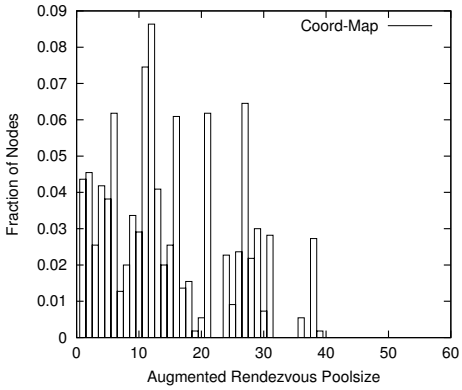

Fig. 6. Coordinates-Based Mapping Augmented RP Poolsize Distribution

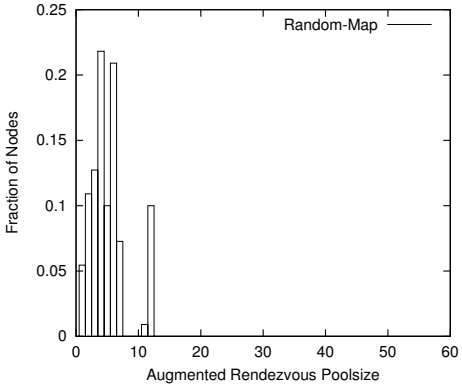

Fig. 7. Random Mapping Augmented RP Poolsize Distribution

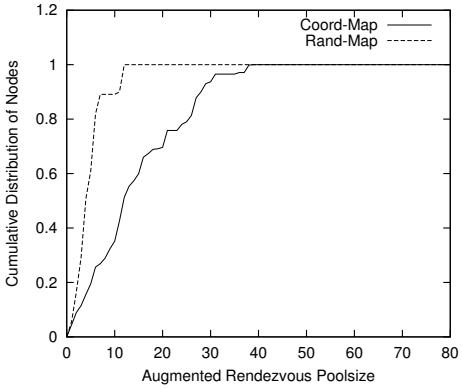

Fig. 8. Augmented RP Poolsize Cumulative Distribution: Random vs Coordinates-Based Mapping

maximum augmented poolsize is close to 40. Further research is necessary to examine the effects of scaling ranges on the performance tradeoffs of the system.

V. Conclusion and Future Work

In this paper, we have presented some initial evidence that a coordinate-based rendezvous scheme can be an effective way for distributed lookup of network neighborhood. The behavior of the algorithm needs to be evaluated further in a large-scale simulation environment. Further, nodes which are isolated in network space will likely be mapped to a rendezvous point with no other nodes in the pool. In this case, we need an efficient lookup algorithm for these nodes to locate its "closeby" nodes on the Chord ring. As part of the future work, we will explore alternative mapping functions to map Internet network locations to Chord key space. In particular, we will focus on the tradeoffs between the rendezvous point poolsize and the lookup efficiency of the rendezvous mechanism. We will also investigate the use of the coordinates-based rendezvous scheme in other scenarios, such as discovering network clusters among distributed peer-to-peer hosts and construction of efficient overlay networks.

References

[1] M. Harchol-Balter, T. Leighton, and D. Lewin, "Resource discovery in distributed networks," in *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC'99)*, 1999.

[2] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *SIGCOM*, 2001.

[3] T. E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *INFOCOM*, 2002.

[4] S. Deering and D. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems*, pp. 85–111, May 1990.

[5] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "An architecture for wide-area multicast routing," in *SIGCOMM*. ACM, 1994.

[6] C. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," in *Theory of Computing Systems*, no. 32, 1999, pp. 241–280.

[7] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *International Conference on Distributed Systems Platforms*, November 2001.

[8] B. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-resilient wide-area location and routing," UCB/CSD, Tech. Rep., 2001.

[9] S. Ratnasamy, P. Francis, M. Handley, and R. Karp, "A scalable content-addressable network," in *SIGCOM'01*, San Diego, CA, 2001.

[10] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS," in *SOSP'01*. Banff, Canada: ACM, October 2001.

[11] A. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel, "Scribe: The design of a large-scale event notification infrastructure," in *Third International Workshop on Networked Group Communications*, 2001.

[12] S. Zhuang, B. Zhao, and A. Joseph, "Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination," in *Eleventh International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Port Jefferson, New York, June 2001.

[13] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *SIGCOMM'02*, 2002.

[14] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, "Exploiting network proximity in peer-to-peer overlay networks," in *Proceedings of the International Workshop on Future Directions in Distributed Computing*, Bertinoro, Italy, June 2002.

[15] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *INFOCOM'02*. New York: IEEE, 2002.

[16] T. Hansen, J. Otero, T. Mcgregor, and H.-W. Braun, "Active measurement data analysis techniques," http://amp.nlanr.net/, 2002.