Distributed Distance Measurement for Large-Scale Networks

Jiangchuan Liu[1], Xinyan Zhang[2], Bo Li[3], Qian Zhang[4], and Wenwu Zhu[5]

[1, 3] Department of Computer Science,
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{csljc,bli}@cs.ust.hk

[2] Department of Information Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
xyzhang2@ie.cuhk.edu.hk

[4, 5] Microsoft Research, Asia
3/F, Sigma Centre, No. 49, Zhichun Road, Haidian District
Beijing, 100080, China
{qianz,wwzhu}@microsoft.com

Abstract

There is an increasing trend in the Internet that a set of replicated providers are qualified for a service or resource request from a client. In this case, it is advantageous to select the *best* provider considering some distance measures, such as hop count or path latency. In this paper, we present a *Group-based Distance Measurement Service* (GDMS), which estimates and disseminates distance information of node-pairs in large-scale wide area networks. GDMS is fully distributed and does not rely on any centralized servers; thus is particularly suitable for the rapidly popularized peer-to-peer applications. The key concept in GDMS is *Measurement Groups* (MGroups). Nodes are self-organized into MGroups to form a hierarchical structure. A set of algorithms are proposed to handle network dynamics and optimize the group organization to reduce system costs as well as improve estimation accuracy. Moreover, a novel multicast-based algorithm is used for both intra- and inter-group performance measurements. Performance evaluation over different network topologies shows that GDMS is scalable and provides effective distance information to upper-layer applications at a relatively low cost.

Bo Li's research is supported in part by grants from Research Grant Council (RGC) under contracts AoE/E-01/99 and HKUST6196/02E. J. Liu's work is partially supported by a Microsoft fellowship. Part of this work was done when J. Liu and X. Zhang are interns at the Wireless and Networking Group, Microsoft Research, Asia.

I. INTRODUCTION

In current networks, given a service or resource request from a client, there are usually a set of qualified providers, for example, mirrors of a FTP server or a cluster of Web servers [5,29,31]. Thus, a key issue in this case is how to efficiently discover and deliver a required resource with a specific Quality-of-Service (QoS). This requires two basic services: 1) a *resource discovery service* to locate the candidate providers; and 2) a *distance measurement service* to measure the network performance between node-pairs, so that the *best* provider can be selected based on some distance measures, such as path latency. Even though the network distance may not be the dominating consideration in some scenarios, it is still useful to include the distance to each candidate provider as a factor in making a selection [5,29].

For large-scale wide area networks, distance measurement using individual probing for each access is clearly not an efficient method. Many existing studies suggest that several measurement servers can be deployed over the global Internet; these servers perform distance measurement on behalf of its local clients, and a client can obtain distance estimations by querying its measurement server. These system are based on the extensively-studied client/server model. However, it is known that the client/server model has some typical drawbacks, such as expensive (to deploy servers for specific purposes) and vulnerable (since there is a single point of failure) in large scale networks.

Recently, a new communication model, *peer-to-peer* communication has emerged at the forefront of Internet computing [1,2]. The rapid and widespread deployment of peer-to-peer applications, such as Napster [3] and Gnutella [4], suggests that there are several advantages of this model over the traditional client/server model. The most important one is its decentralized or distributed nature. That is, resources such as multimedia files are stored in end users' machines (hosts, or *peers* in this paper) rather than in a central server and, as opposed to the client/server model, resources are transferred directly between peers. Therefore, with this model, resources stored or replicated in the global Internet can be fully utilized. Moreover, the potential bandwidth or processing bottlenecks at the server's end can be alleviated, and the hazard caused by the failure of a server is reduced.

Several pure decentralized resource discovery services have been proposed for peer-to-peer applications, such as Chord [23] and CAN [24]. However, to our knowledge, there are few decentralized network distance measurement services that are specifically designed for peer-to-peer applications in a wide area network environment.

In this paper, we propose a decentralized peer-to-peer distance measurement service for large-scale wide area networks, GDMS. This service does not rely on any centralized server, but uses a self-organizing infrastructure. The key concept in GDMS is *Measurement Groups* (MGroups)[2]. We distinguish the peer-pair performance information into *intra-* and *inter-group measures* to achieve a scalable and efficient solution. Peers are self-organized into MGroups and a group leader is dynamically elected which acts as a representative for inter-group distance measurement. We devise a set of distributed group forming algorithms to handle network dynamics, balance the workload of different peers, and minimize the overall measurement cost. Moreover, a novel multicast-based measurement algorithm is used for both intra- and inter-group measurements. The algorithm is highly scalable and incurs much lower overheads compared to traditional unicast-based measurement algorithms.

We envision GDMS as an underlying measurement service that provides peer-pair distance performance information to upper-layer applications. The performance of GDMS has been evaluated under various network topologies. The results show that GDMS can indeed provide useful distance information for QoS-aware peer-to-peer applications at a reasonable cost.

The rest of the paper is organized as follows. In Section II, we present some related work and discuss our design objectives. Section III describes the system model of GDMS and makes some basic assumptions. Section IV presents the group forming algorithm for GDMS and its optimizations. An efficient and scalable multicast-based algorithm for both intra- and inter-group measurements is described in Section V. The cost and scalability of GDMS is analyzed in Section VI, and its performance is evaluated in Section VII through simulations. Finally, Section VIII concludes the paper.

II. RELATED WORK AND OUR DESIGN OBJECTIVES

There has been extensively work on distance measurement for wide area networks, specifically, the Internet. There is also much work on neighbor discovery or locating the best service provider given some QoS measures [5,27,29,31,32]. The proposals generally assume there is an underlying distance measurement service or can best be supported by such a service. To date, many of the distance measurement services are designed for the client/server based applications. An example is the anycasting service [5], in which a set of anycast resolvers measure the response times of replicated servers on behalf of clients, and direct a client's request to the "best" server. Other proposals like SONAR [6], HOPS [7], and IDMaps [13], though do not target applications with specific communication models, adopt the client/server model for measurement; that is, a set of servers are assumed to be placed over the global Internet; they perform measurements on behalf of clients, and the clients query these servers to obtain the distance information. A key design issue for such systems is where the measurement servers should be placed. This placement problem have been studied in [10,13,17] with the objective of minimizing average

[2] In the rest of this paper, unless explicitly specified, a *group* means an *MGroup* .

measurement errors, and a set of heuristic algorithms are proposed for placement with a given number of servers and a specified network topology.

The design objective of GDMS is to provide a measurement service based on the peer-to-peer communication model yet retains high measurement accuracy. In the following, we first discuss some basic design considerations in GDMS.

- *Network Performance Metrics of Interest.* GDMS is designed as an underlying measurement service to provide peer-pair performance information to upper-layer applications. To this end, the information provided by GDMS should be generic enough, for example, the "raw" measures between peer-pairs: hop-count, path delay, and bandwidth. In this paper, we present the use of GDMS with a set of additive distance metrics, because these metrics are relatively easy to measure and, fortunately, typical distance metrics, such path delay, are very useful for QoS-aware applications [10,13]. However, GDMS does not restrict itself to any specific performance metric; its architecture can accommodate other metrics.

- *Scalability and Efficiency*. Peer-to-peer systems have become quite popular in the last two years. In July, 2001, Gnutella had about 40,000 simultaneous users, and many more potential users grow with time [4]. Thus, any measurement service for such systems should solve the scalability problem. Efficiency is also an important consideration because the overhead of measurement in a large system could be very high without a proper design. For example, a simple method to obtain peer-pair performance information is for the initiating host to measure it itself, using tools such as *ping* or *traceroute*. While these tools are easy to use, their utility is generally limited by their overhead. For instance, the latency of running a single *traceroute* can exceed the latency of a discovery query itself. More important, a large number of hosts making independent and frequent measurements could have a severe impact on the Internet.

Ideally, measurements made by one host should be shared, with low overheads, by other hosts. This is essential for peer-to-peer applications because any peer-pair in the network may have conversations at any time. Previous work on network performance measurements has shown that a hierarchical infrastructure is a highly scalable and efficient solution for large networks regarding information sharing [13]. Therefore, GDMS adopts a two-level hierarchy based on the concept of MGroup.

- *Decentralization and Adaptivity.* As mentioned before, a peer-to-peer system generally prefers distributed control where there is no centralized server. In addition, a peer-to-peer system is usually dynamic where distances vary over time and nodes (peers [3]) could join or leave the system or move to other locations at will. Therefore, the measurement service should be adaptive to handle such dynamics. To this end, in GDMS, nodes are self-organized into measurement groups, and the organization of the groups is

[3] In the rest of this paper, we use *node* and *peer* interchangeably.

dynamically adjusted over time. A similar architecture is shown in [32], where a term *bin* is used instead of group. Their work is mainly on constructing an overlay that is aware of the underlying network latency. The objective of GDM, however, is on designing the low-cost distance measurement methods and providing 'raw' distance information for diverse upper-layer applications.

Note that the self-organizing group infrastructure introduces errors caused by approximation, such as the use of distances between group leaders to estimate peer-pair distances. These errors are sensitive to the organization of the groups. Thus, an important objective of this paper is to optimize the group forming procedure and, to find a trade-off between the efficiency and accuracy of measurements in a dynamic environment.

III. System Model and Assumptions

We consider a peer-to-peer communication system which consists of a set of nodes (peers) belonging to a wide area network. There is no centralized server in the system. A node can communicate to any other node at will. In addition, a node can join and leave the system or move to another location of the network with the support of Mobile IP [26] at any time.

To facilitate further discussions, we define the following notations:

- N : The number of nodes (peers) in the system;
- k : The number of groups formed in GDMS;
- G_i : The i th group. The group ID of G_i is *i*.
- $|G_i|$: The size (number of nodes) of group G_i ;
- L_i : The leader of group G_i ;
- (x, y) : The exact distance between nodes x and y .

Two nodes, *x* and *y*, are said to be *nearby* if $(x, y) < T$, where T is a threshold. We define a MGroup as a set of nearby nodes in which a specific node will be elected as the *group leader*. In GDMS, the network nodes will be self-organized into MGroups using a group forming algorithm.

As mentioned before, the performance metric of interest here is an additive distance. In [13] and [17], a series of experiments were conducted on the global Internet to study the properties of such distances. The results show that, in most cases, they satisfy the *triangle inequality*; that is, for three nodes x, y, z in the network, inequality $(x, y) \le (x, z) + (z, y)$ holds. This is because the Internet routing protocols try to find the shortest paths, and the routes used by two nearly hosts usually do not drastically differ from each other. Though this property does not necessarily hold over all parts of the Internet, it is feasible to use triangulation to estimate distances, as shown in [13]. From the triangle inequality, we can further derive that

$$|(x,z)-(z,y)| \le (x,y) \le |(x,z)+(z,y)|. \tag{1}$$

Assume nodes x and z are in different groups, and y is the leader of x's group. From inequality (1), we have $|(x,y)-(y,z)| \le (x,z) \le |(x,y)+(y,z)|$. As a result, if T is very small, we have $(x,z) \approx (y,z)$, and we can thus use (y,z) to estimate (x,z). Hence, distance measurements can be divided into two parts:

- *Intra-group Measurement.* The full-mesh distance information of all peer-pairs in a group. It is measured by the nodes in the same group.

- *Inter-group Measurement.* The full-mesh distance information between the group leaders in the system. It is measured by the group leaders, and a leader will disseminate this information to all the members in its group. When a non-leader node needs to measure the distance to another node in a different group, it can use the distance between the two group leaders as an approximation according to the property of nearby nodes.

We assume that the network supports IP multicast [25]. Multicast messages will be used for group formation. Furthermore, an efficient multicast-based method will be used for both intra- and inter-group measurements to obtain the full-mesh peer-pair performance information. Nevertheless, multicast is not an indispensable requirement for GDMS, but it will greatly improve the performance of GDMS compared to unicast-based measurement, as we will show in Section V.

This self-organizing hierarchical structure has several advantages. First, it is very efficient for multiple nodes to share measurement information. Second, note that both a group and its leader are dynamically formed or elected, this makes the architecture very suitable for a peer-to-peer system with dynamically joining and leaving nodes, or mobile nodes. Finally, by using a hierarchical organization and an efficient multicast-based measurement algorithm, it provides a scalable solution and incurs low overheads for measurement. In this paper, we focus on a 2-level hierarchy; however, by using more levels, this system can scale to much larger networks.

IV. FORMATION AND OPTIMIZATION OF GROUPS

At the bootstrapping stage of GDMS, a group forming algorithm is used to form MGroups. When a node joins or leaves the system, or moves to another location, the algorithm is also executed to adjust the group organization. In this section, we first present a basic group forming algorithm, and then devise a set of heuristics to reduce its estimation errors as well as to improve its efficiency.

A. The Basic Group forming Algorithm

Note that the fundamental error in GDMS is the use of group-based estimations, i.e., using the distance between the group leaders to approximate the distance between two non-leader nodes. Therefore, the primary optimization objective of group formation is to minimize the expected estimation error for all peer-pairs belonging to different groups, or so-called *k-Average Error* given *k* groups are formed [10,13], as follows (Problem P1):

$$\text{(P1) Minimize } \frac{1}{N'}\sum_{\substack{i,j\in[1..k]\\ i<j}}\sum_{\substack{x\in G_i\\ y\in G_j}}\frac{|(x,y)-(L_i,L_j)|}{(x,y)}, \text{ where } N'=\sum_{\substack{i,j\in[1..k]\\ i<j}}|G_i\|G_j|. \tag{2}$$

To achieve this objective, a node should join the group whose leader is nearest to it. In GDMS, a multicast group with a known IP address of M is used for exchanging control messages for group formation. When node x joins the system or moves to a new location, the following basic forming algorithm is executed to decide whether to create a new MGroup or to let x join an existing group.

1. Node *x* joins the multicast group *M*, and sends a request with a specified Time-to-Live (TTL) value to find its nearby group leaders.
2. If a leader receives the request, it will reply by providing its group ID.
3. If node *x* receives a reply within a latency of T_{Nearby}, it will join the corresponding *MGroup*. Here T_{Nearby} is a specified threshold.
4. Otherwise, *x* should create a new *MGroup* and act as the leader.

When a leader leaves the system or is failed (this can be detected if it does not distribute messages to its group members for a long time), all non-leader nodes in that group should also perform the above algorithm to join other groups or, possibly, to create new groups and act as the leaders. Note that, too many nodes simultaneously execute the forming algorithm may cause message implosion in such a multicast environment [14]. To avoid this hazard, a node should set an exponentially distributed delay timer if it wants to join the system, and execute the above algorithm after the timer expires [14].

B. Heuristics for Group Optimization

The basic group forming algorithm is simple for implementation. However, it has several limitations. Specifically, the leader of a group is fixed unless this leader leaves the system. Similarly, a node is fixed to the group it first joins unless the leader of that group leaves the system. Since the system is dynamic in that distances vary over time and nodes join, leave or move, the original selections may not be optimal which may cause significant estimation errors in the inter-group measurements.

To achieve better performance, it is necessary to reform groups during their life time. Note that the optimal solution to problem P1 requires the global and precise distance information between any two nodes. However, this information is not available to individual nodes in the distributed GDMS system. Specifically, if a node *x* is in group G_i, it only knows (L_i, L_j), *i≠j*, by inter-group measurements, and (x, L_i) by intra-group measurements. Furthermore, to our knowledge, there is still no polynomial-time solution to problem P1 even if the global and precise information is available. Therefore, in the rest of this section, we propose a set of heuristics to find a near-optimal formation with local, imprecise and dynamic distance information.

C. Group Re-selection (GR)

With the GR algorithm, if node x finds its current group G_i is not suitable any more, it should select a better group and join it. Let L' denote the leader of the group to be selected, the selection criterion should be

$$L' = \arg_{L_j} \ \min\{(x, L_j) : (x, L_j) < (x, L_i), j \in [1, \ldots, k]\}, \tag{3}$$

which basically returns the nearest leader to node *x*. Note that node *x* can not directly use this criterion to find a better group because (*x*, *L*$_j$), *j≠i*, is not available to it. However, according to the triangle inequality, we have $|(x, L_i) - (L_i, L')| \le (x, L')$. Therefore, *L'* given by Equation (3) should satisfy $|(x, L_i) - (L_i, L')| \le (x, L_j) < (x, L_i)$, and hence

$$(L_i, L') < 2(x, L_i). \tag{4}$$

This gives a loose criterion for the selection of the best group and is based on a node's local information. If node *x* finds some leaders in other groups satisfy the above criterion, it can initiate probes to these candidates to find the best one.

D. Leader Re-Election (LR)

With the LR algorithm, the nodes in a group will periodically re-elect a better leader among them. Similar to that in group re-selection, the criterion for leader election should be based on a node's local information. We have studied various criteria for leader re-election. Through both experiments and analysis, we find that the following criterion exhibits superior performance in most cases,

$$L_i = \arg_{x} \ \text{median}\left\{ \sum_{y \in G_i} (x, y) : x \in G_i \right\}. \tag{5}$$

The rationale of using this median-based criterion can be found in Appendix A. Note that this criterion only relies on intra-session distances which are available to all the nodes in a group. A node can thus determine the new leader locally, yet resulting in the same selection as others. However, in an unreliable network where messages could be dropped, we recommend the following process.

```
1. The current leader $L_i$ find a leader candidate $L_i'$ by (5);
2. If $L_i = L_i'$, goto 1. Otherwise, $L_i$ should contact $L_i'$ to see whether
   it is willing to be the new leader. If $L_i'$ agrees, goto 3;
   otherwise, goto 1;
3. $L_i'$ acts as the new leader and announces this handover to all
   other leaders and all other nodes in $G_i$. Goto 1.
```

E. Group Number Regulation

The basic algorithm does not consider the number of groups in the system. However, it is an important control parameter relating to the cost and accuracy of measurements. We will formally study the choice of the number of groups in Section VI. Practically, to limit the group number to a given threshold k, we assume that when the group number reaches k, a flag in each inter-group measurement message is set to 1. When a new node tries to join the system, it should first join multicast group *M* and listen to at least one inter-group measurement message. If the flag is 0, it can use the basic (or with GR and LR) algorithm. Otherwise, it will use an expanded ring search to find a nearest leader and join that leader's group.

V. INTRA- AND INTER-GROUP MEASUREMENTS AND SHARING

In this section, we first propose a novel multicast-based measurement algorithm for both intra- and inter-group measurements. We then discuss the dissemination and utilization of the measurement information.

A. Multicast-based Measurement Algorithm

The fundamental requirement of intra- and inter-group measurements is to measure the distance between any two nodes (a peer-pair). To this end, a simple way is to let each node measure its distances to all other nodes by unicast-based probing. This method, though straightforward and simple, has two shortcomings. First, for a node in group *G*, it should send probing packets to all other nodes, i.e., send $O(|G|)$ packets. Second, a node can obtain only the distances between itself and other nodes while not the full-mesh distance information of all the peer-pairs.

To solve these problems, we devise a novel multicast-based measurement algorithm. This algorithm has two phases. In the first phase, the group leader initiates a probe by multicasting a packet to all group members. Each node that receives this probe will give a reply which is also multicasted. Thus, the reply is received not only by the leader but also by all other nodes in the group. From these replies, each node obtains a set of local (or partial) measurements. In the second phase, each node multicasts its local measurements to the whole group. By manipulating the local information from all other nodes, each node can calculate the full-mesh distances.

For illustration, we use the measurement of the Round-Trip Time (RTT) as an example. We assume that there is a group G consisting of 4 nodes, a,b,c,d , and a is the leader. First, node a initiates a multicast probe, and all other nodes reply to this probe using multicast as well. Figure 1 shows the routes of the packets sent and received by each node in this phase. The routes of the packets received by each node are also listed in Table 1.

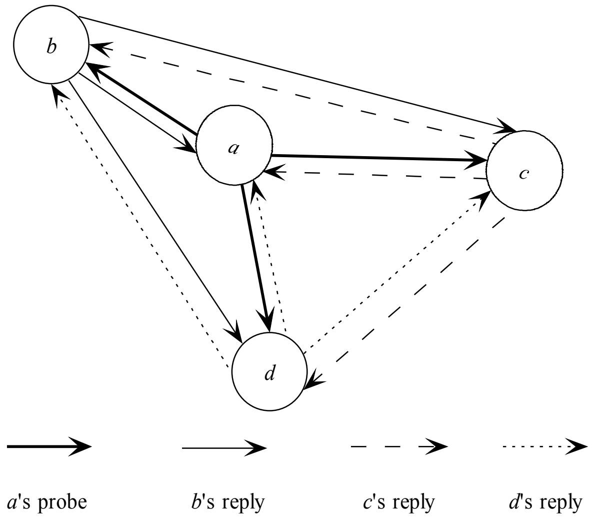


Figure 1: Routes of the packets sent and received by each node in the first phase. Note that all packets are transmitted by multicast.

Node *a*	Node *b*	Node *c*	Node *d*
	a-b	*a-c*	*a-d*
a-b-a		*a-b-c*	*a-b-d*
a-c-a	*a-c-b*		*a-c-d*
a-d-a	*a-d-b*	*a-d-c*	

Table 1: Routes of the packets received by each node in the first phase.

In the table, *a-x* means the probe from leader *a* to node *x*, and *a-x-y* means node *x's* reply to *a* has been received by node *y*. Define (*x*↔*y*) as the round-trip time between *x* and *y*, and (*x*→*y*) as the one-way trip time from *x* to *y*. After the first phase, the following local (or partial) measurements are available at the four nodes.

Node a:

$a.1$: $(a\rightarrow b)+(b\rightarrow a)=(a\leftrightarrow b)$

$a.2$: $(a\rightarrow c)+(c\rightarrow a)=(a\leftrightarrow c)$

$a.3$: $(a\rightarrow d)+(d\rightarrow a)=(a\leftrightarrow d)$

Node b:

$b.1$: $(a\rightarrow c)+(c\rightarrow b)-(a\rightarrow b)$

$b.2$: $(a\rightarrow d)+(d\rightarrow b)-(a\rightarrow b)$

Node c:

$c.1$: $(a\rightarrow b)+(b\rightarrow c)-(a\rightarrow c)$

$c.2$: $(a\rightarrow d)+(d\rightarrow c)-(a\rightarrow c)$

Node d:

$d.1$: $(a\rightarrow b)+(b\rightarrow d)-(a\rightarrow d)$

$d.2$: $(a\rightarrow c)+(c\rightarrow d)-(a\rightarrow d)$

In the second phase, each node sends the above local information to all other nodes using multicast. After that, each node can locally calculate the full-mesh RTT information. For example, we have

$$\begin{aligned} & b.1+c.1 \\ & =[(a\rightarrow c)+(c\rightarrow b)-(a\rightarrow b)]+[(a\rightarrow b)+(b\rightarrow c)-(a\rightarrow c)] \\ & =(c\rightarrow b)+(b\rightarrow c)=(b\leftrightarrow c) \end{aligned} \tag{6}$$

Similarly, we have $b.2+d.1=(b\leftrightarrow d)$ and $c.2+d.2=(\mathrm{c}\leftrightarrow d)$.

Here we assume that each node replies to the probe immediately. However, in large groups, this method may cause the well-known *feedback implosion* problem [14]. To avoid implosion, an exponentially distributed delay timer [14] should be set for each reply, and the delay should be included in the reply for final adjustment.

It can be seen that, in our algorithm, each non-leader node sends two replies and receives $2|G|$ replies. This is much lower than the cost of using unicast-based probing. More important, the full mesh distance information is available in each node after these two phases.

B. Dissemination and Utilization of Measurement Information

In GDMS, the above multicast-based algorithm is also used for inter-group measurement, where L_1 serves as a leader for all the group leaders. A group leader will also disseminate the inter-group measurement results, i.e., the full-mesh distances between leaders, to all other nodes in its group.

When a node intends to access a resource, it first calls a QoS-aware resource discovery service. The discovery service locates all the qualified nodes (providers) that have the resource of interest, and then uses GDMS to decide which one should be selected as the provider. The selection is based on their distances to the requesting node. If there exist candidates that are in the same MGroup of the requesting node, the one with the smallest intra-group distance will be selected. Otherwise, the one with the smallest inter-group distance will be selected. Finally, this selection will be returned to the requesting node, and it will then initiate an access request to the selected provider for peer-to-peer delivery.

VI. Analysis of Cost and Scalability

In GDMS, there is a tradeoff between the measurement cost and accuracy. In general, GDMS achieves high precision when there are enough number of groups. Specifically, if there are N groups in the system, i.e., each group has only one member (the leader), the accuracy is 100% if the error of probing is not taken into account. Another extreme is the use of only one group. In both cases, GDMS is reduced to a non-hierarchical system whose cost could be very high.

In this section, we formally analyze the cost and accuracy of GDMS, and try to find a balance between them. Since GDMS is an overlay service implemented in the application level, we mainly focus on the cost or workload of end systems, which is defined as the average number of packets transmitted and received per node per measure. Note that, given a group G, the cost of obtaining the full-mesh distance information using our multicast-based measurement algorithm is $O(|G|)$. Therefore, in GDMS, for a non-leader node in group $|G_i|$, its cost $C_i^{NonLeader}$ is $O(|G_i|)$. For leader L_i, its cost C_i^{Leader} is $O(|G_i|+k)$, where k is the number of groups in the system. The average cost for a node, C^{GDMS}, is thus given by

$$C^{GDMS} = \frac{1}{N}\{\sum_{i=1}^{k}\sum_{x\in G_i, x\neq L_i} O(|G_i|) + \sum_{i=1}^{k} O(|G_i|+k)\}$$

$$= \frac{1}{N}\{\sum_{i=1}^{k}\sum_{x\in G_i} O(|G_i|) + \sum_{i=1}^{k} O(k)\} = \frac{1}{N}\sum_{i=1}^{k} O(|G_i|^2) + \frac{1}{N}O(k^2) \quad (7)$$

Note that $\sum_{i=1}^{k}|G_i| = N$. Therefore, for a given k, the first item of C^{GDMS} is minimized when $|G_i| = N/k$, $i = 1,2, ...,k$. In this case, the costs of all non-leaders are identical, so are all leaders. Thus, the cost C^{GDMS} can be represented as a function of N and k, as follows,

$$C^{GDMS} = \frac{1}{N}\{\sum_{i=1}^{k} O[(\frac{N}{k})^2] + O(k^2)\} = O(\frac{N^2+k^3}{Nk}) \quad (8)$$

It can be shown that C^{GDMS} is minimized when $k = \frac{1}{\sqrt[3]{2}}N^{2/3}$. In this case, the costs (workload) of a non-leader, $C^{NonLeader}$, and a leader, C^{Leader}, are $O(N^{1/3})$ and $O(N^{2/3})$, respectively. Hence, a leader's workload is $N^{1/3}$ times of a non-leader's. However, in the peer-to-peer communication model, it is desirable to distribute the workload more evenly. Therefore, we add another constraint as follows,

$$\frac{C^{NonLeader}}{C^{Leader}} = \lambda, \text{ where } \lambda \text{ is a constant } >1. \quad (9)$$

It can be shown that $k = \sqrt{(\lambda - 1)N}$. Specifically, when $\lambda = 2$, we have k=| G_i|=$\sqrt{N}$,i=$1,2,...,$ $\sqrt{N}$, and the average cost is $O(\sqrt{N}\,)$. This suggests that GDMS achieves better scalability compared to a non-hierarchical measurement system. In Figure 2, we show the costs as functions of k for a 6000-node network. Practically, the choices of k can be made by considering the characteristics and requirements of specific network environments.

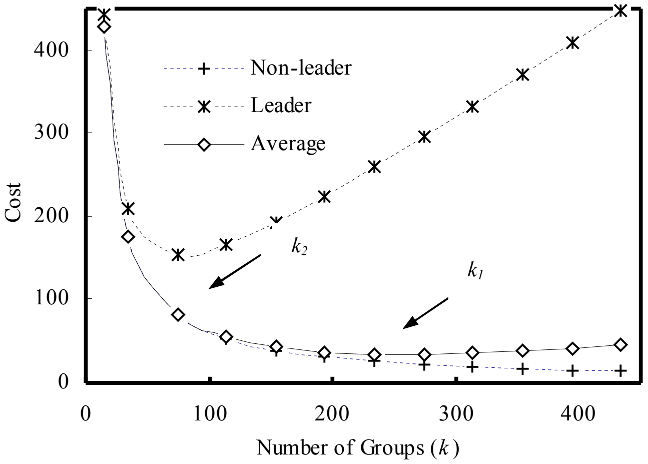


Figure 2. Costs as functions of the number of groups (k). The cost of a non-leader is $O(N/k)$, the cost of a leader is $O(\frac{N}{k} + k)$, and the average cost is $O(\frac{N^2 + k^3}{Nk})$. In this figure, N=6000, $k_1 = \frac{1}{\sqrt[3]{2}} N^{2/3} = 262$, and $k_2 = \sqrt{N} = 77$.

VII. Performance Evaluation

In this section, we evaluate the performance of GDMS through extensive simulations. Our main objective in the performance evaluation is to investigate whether GDMS provides effective measurement results to upper layer services. Note that the fundamental error of GDMS is the use of the MGroup-based approximation. Therefore, we first simulate different group forming algorithms on a variety of network topologies to investigate the estimation errors caused by grouping. We then investigate the correctness of using GDMS for QoS-aware discovery.

A. *Evaluation Methodology and Settings*

The topological structure of a network is typically modeled using a graph, with graph nodes representing routers and edges representing direct connections between routers. A host node (peer) is represented by a leaf connected to a single router node. We use three models to generate network topologies:

- Inet model [17] aims at generating graphs with power-law node degree frequency distribution of Internet topologies, as reported in [18].

- Waxman model [15] produces flat random graphs but includes network-specific characteristics such as using a distance-aware probability function to interconnect two nodes.

- Transit-Stub (TS) model [16] focuses on reproducing the hierarchical structure of the Internet topology by composing interconnected transit and stub domains.

We decide to use more than one topology generator because the actual topology of the Internet is still under research. The TS and Inet models reflect the hierarchical structure of the Internet from different aspects. Waxman model, though does not explicitly attempt to reflect the structure of the real Internet, is attractive for its simplicity and is commonly used to study networking problems. Detailed description of the topology generation procedures in our study can be found in Appendix B. In the following part, we will present the results of 9 networks, namely, networks of 3000-node, 6000-node and 12000-node from each model.

We assume router and links are persistent components in a network and the path delay between a peer-pair varies from 20 ms to 200 ms, following a 10-state Markov Process [28]. A node dynamically joins and leaves the system. We represent the join/leave process of a host using an exponentially distributed ON/OFF model with average T_{ON}=300 seconds and T_{OFF}= 300 seconds, respectively. The node alternates between OFF and ON periods such that it activates and participates in the system during an ON period, and is off-line during an OFF period. In addition, to emulate the scenario where mobile IP [26] is adopted, a small portion of nodes (1%) changes its location in each OFF period.

We use shortest-path routing for unicast and shortest-path tree routing for multicast. Since these two methods are widely used in conventional unicast or multicast routing protocols, our conclusions are general while not restricted to specific routing protocols.

B. Accuracy of Inter-Group Measurements

In this set of experiments, we study the relative accuracy of the use of inter-group approximation in GDMS. The metric of accuracy is defined as follows:

$$1-\frac{1}{N'}\sum_{\substack{i,j\in[1..k]\\ i<j}}\sum_{\substack{x\in G_i\\ y\in G_j}}\frac{|(x,y)-(L_i,L_j)|}{(x,y)},\quad \text{where}\quad N'=\sum_{\substack{i,j\in[1..k]\\ i<j}}|G_i||G_j|. \tag{10}$$

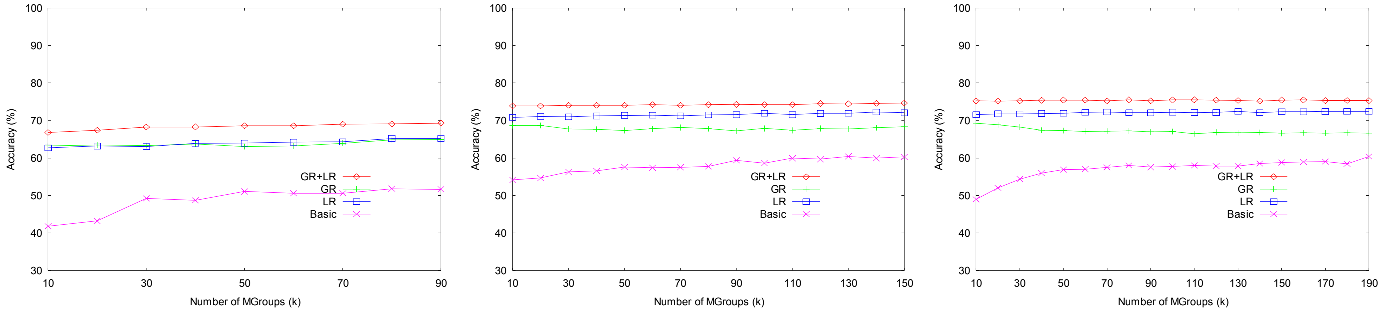
We evaluate the accuracy with all the group forming algorithms, including the basic algorithm, and its extensions with Group Re-selection (GR), Leader Re-election (LR), and GR+LR. For a given group number, *k*, we conduct 10 simulations on each topology. A simulation lasts 3000 seconds, and a node performs a measurement every 10 seconds. Note that our estimation method is based on the triangle relations, which, for one peer-pair estimation, involves only the initiated node and two group leaders. Therefore, the estimation errors will not accumulated, and we obtain the overall accuracy by averaging the accuracy of all the measurements.

Figure 3 shows the results under different topology settings. We find that, in all the settings, the accuracy of inter-group measurements generally increases with the increase of group numbers. This is because the expected group size decreases when the number of groups increases, and the error caused by using group leaders as representatives is thus reduced as discussed in Section III.

The use of different group forming algorithm also influences the measurement results. In all the settings, the accuracy of using the heuristic group forming algorithms (GR, LR, or GR+LR) is higher than that of the basic algorithm. Specifically, the basic algorithm often exhibits very poor performance with a small number of groups. On the contrary, the heuristic algorithms, especially GR+LR, still retain reasonable accuracy (70%) with a small number of groups.

Note that Figure 3 shows the accuracy of inter-group measurements only. The much more accurate intra-group measurement results are not included. Therefore, the overall accuracy of GDMS is expected to be higher and, in our experiments, we find it is more than 90 % in most cases with the GR+LR algorithm. Thus, we adopt the GR+LR algorithm as the group forming algorithm in GDMS.

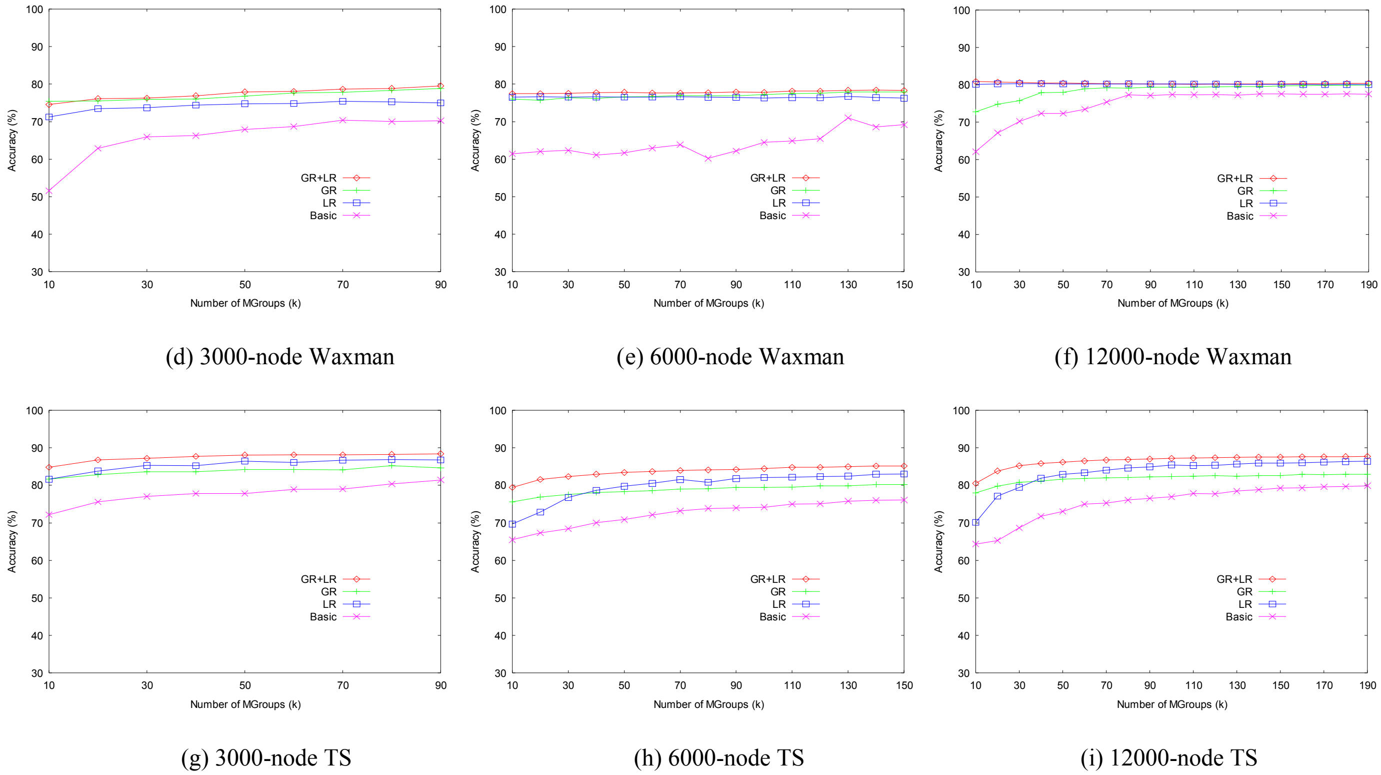
From these results, we conclude that the group-based measurement architecture with our heuristic group forming algorithm provides reasonably accurate and stable measurements in practice.



(a) 3000-node Inet

(b) 6000-node Inet

(c) 12000-node Inet



(d) 3000-node Waxman

(e) 6000-node Waxman

(f) 12000-node Waxman

(g) 3000-node TS

(h) 6000-node TS

(i) 12000-node TS

Figure 3. Comparison of the accuracy of inter-group approximation with different algorithms.

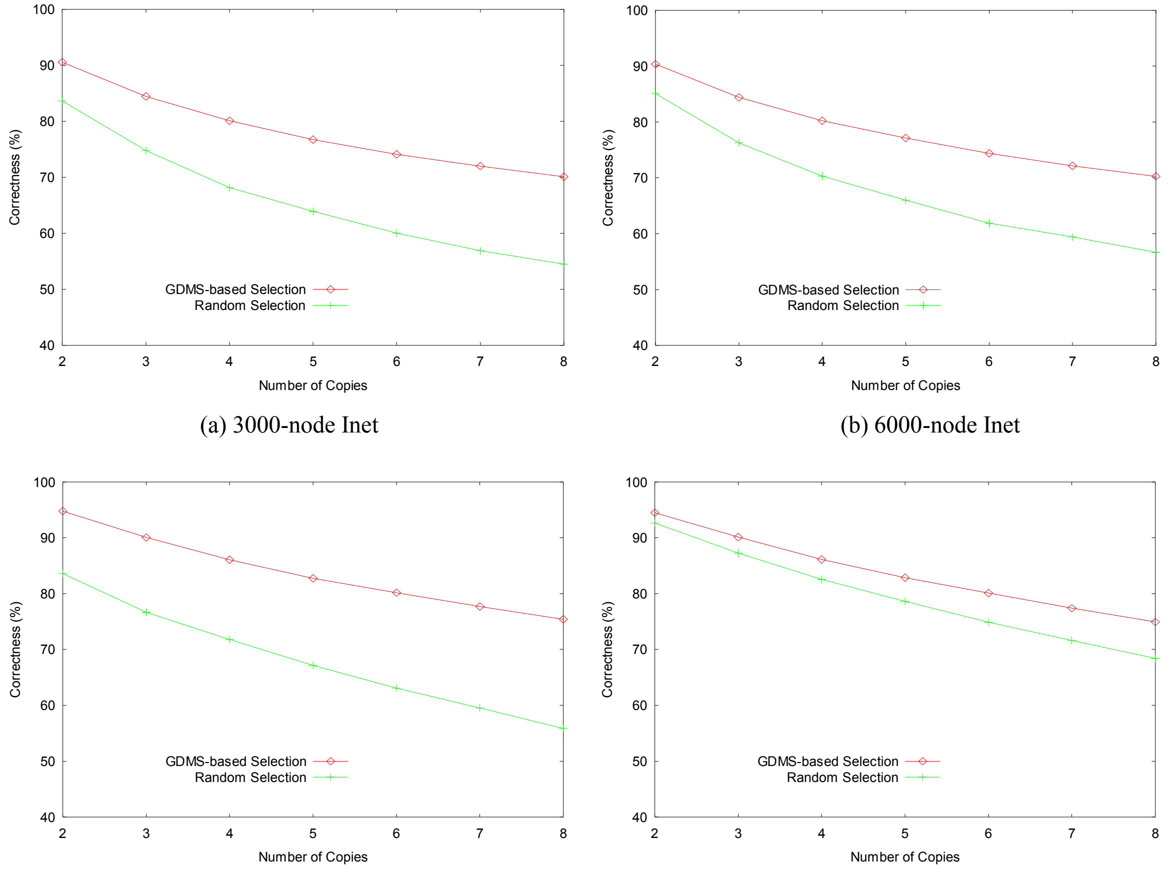
C. *Effectiveness in QoS-aware Resource Discovery*

As we stated before, the ultimate objective of GDMS is to provide useful distance information to higher-level services, such as resource discovery. Therefore, in the second set of experiments, we study the effectiveness of GDMS for QoS-aware resource discovery.

In our experiments, we generate 1000 different types of resources. Each type of resource has R replicated copies which are stored in R randomly selected nodes (providers). Therefore in the network each node, on average, has $1000R/N$ resources. We assume that each node initiates one query for resource discovery every 30 seconds. The types of the queried resources are randomly generated, and no provider is in the same group as the querist.

The GDMS service is used for resource provider selection. For comparison, we also implement a system which uses random selection. Here, the performance of interest is the percentage of correct selections. We consider a selection is correct as long as the distance between the querist and the selected provider is within γ times of the distance between the querist and the nearest provider. In this study, we use $\gamma = 1.5$, by which a querying host will not experience a perceptible difference in the Internet environment [13].

Figure 4 shows the correctness of provider selection in resource discovery. We can see that GDMS exhibits reasonably good performance in the experiments. The correctness of the GDMS-based selection is over 80% in most cases, and outperforms the random selection in all the cases. The differences are about 20%-60%. Moreover, the correctness of GDMS-based selection does not decrease drastically with the increase of the number of replications. On the contrary, the random selection exhibits very poor performance when more than 3 replications are placed. Specifically, in Figure 4(e), 3000-node TS topology, when they are 8 replications, the correctness of GDMS is still higher than 95%, where the correctness of the random selection drops to less than 40 %. Since the TS model reflects the hierarchical infrastructure of the Internet, we believe that GDMS is particularly suitable in such a scenario as it itself uses hierarchical organization.

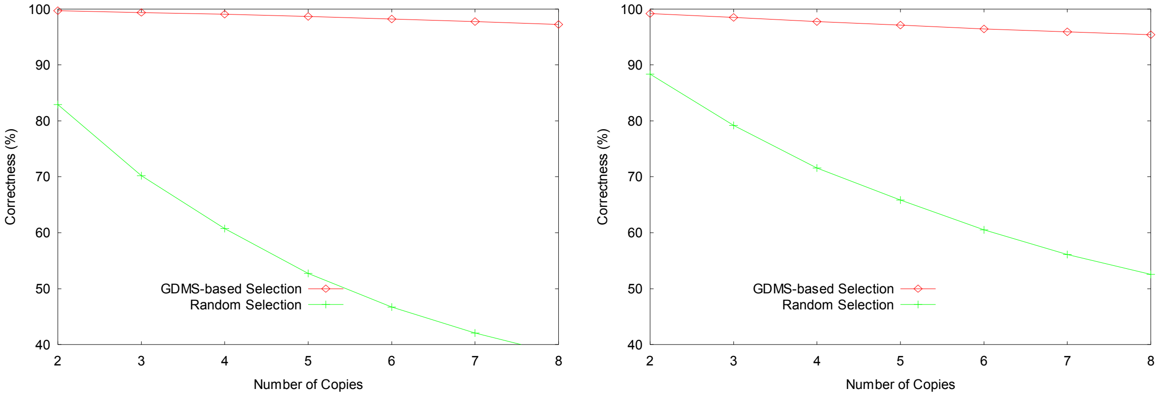


(a) 3000-node Inet

(b) 6000-node Inet

(c) 3000-node Waxman

(d) 6000-node Waxman

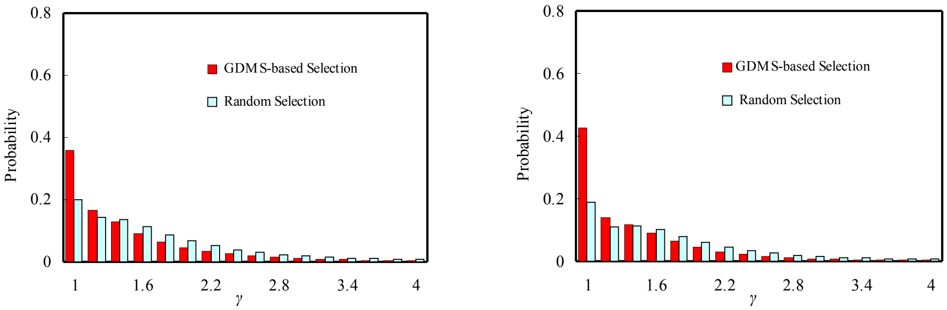


(e) 3000-node TS

(f) 6000-node TS

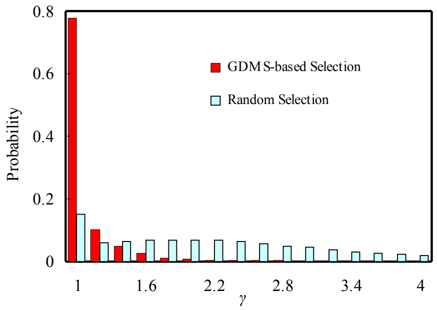
Figure 4. Percentage of the correct answers in QoS-aware resource discovery for peer-to-peer applications. In GDMS-based selection, $\sqrt{N}$ groups are used for a network with N nodes.

In Figure 5, we show the probability distributions of $\gamma\,(=\dfrac{\text{Distance to the selected provider}}{\text{Distance to the nearest provider}})$. It can be seen that GDMS can locate the nearest provider with much higher probability than the random selection. In addition, by GDMS, γ is usually controlled in 2, that is, 2 times the best selection. However, by the random selection, γ exhibits a heavy-tail distribution and sometimes exceeds 4, especially with the TS topology. Therefore, the hazard of selecting a ‘bad’ provider is greatly reduced by using GDMS.



(a) 3000-node Inet

(b) 3000-node Waxman



(c) 3000-node TS

Figure 5. Probability distributions of γ, with a sampling interval of 0.2. Here, $\sqrt{N}$ groups are used for a network with N nodes.

The comparison results basically show that GDMS can provide useful distance information in QoS-aware resource discovery for peer-to-peer applications, where the random selection is not satisfactory. In summary, our main results in this evaluation can be concluded as follows:

- Using our heuristic group forming algorithm (GR+LR), the estimation errors caused by grouping is well-controlled. In most settings (topologies, network sizes, etc.), GDMS offers satisfactory measurement results.
- The GDMS-based provider selection works well in most settings, and offers noticeable improvement over random selection in QoS-aware resource discovery for peer-to-peer applications. It also greatly reduces the hazard of selecting providers of much longer distances than the best candidate.

VIII. CONCLUSIONS

In this paper, we have proposed a decentralized network distance measurement service for large-scale wide area networks. This *Group-based Distance Measurement Service* (GDMS), provides a 2-level hierarchical measurement framework by using self-organized measurement groups. We have devised a set of distributed group forming algorithms to handle network dynamics, balance the workload of different peers, and minimize the overall measurement cost. Moreover, a novel multicast-based measurement algorithm has been used for both intra- and inter-group measurements. The algorithm is highly scalable and incurs much lower overheads compared to traditional unicast-based measurement algorithms. Through analysis and simulations, we showed that GDMS can indeed provide useful peer-pair distance information at a reasonable cost.

Appendix A: The Rationale of Median-based Leader Re-election

For leader re-election, a straightforward idea is to chose the node that has the minimal average distance to other nodes in the group, as follows,

$$L_i = \arg\min_x \left\{ \sum_{y \in G_i} (x, y) : x \in G_i \right\} \tag{11}$$

However, our experiments show that this *min*-based criterion does not give the optimal leader selection, and even gives the worst. On the contrary, the *median*-based leader election gives stable and near-optimal results. In this appendix, we give a brief explanation of this phenomenon.

For simplicity, we use the average absolute error as the error measure. Moreover, we assume that only one group, G, will re-elect its leader, L', and we only consider the error of the distance measurement to the leader node r of another group. The measurement error with a chosen leader L' of group G can be expressed as follows,

$$\frac{1}{|G|} \sum_{x \in G} |(x,r) - (L',r)|. \tag{12}$$

For illustration, we consider a group G which has four members *a*, *b*, *c*, *d*, and one reference node r which is outside of G. In Figure 6, we line up the nodes of G according to their distances to r. From this figure, it can be easily proven that the measurement error is minimized by choosing the node with the median distance to r as the leader.

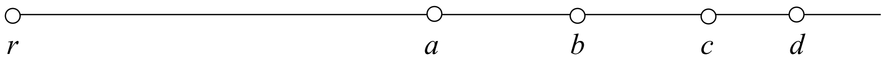


Figure 6. The distances from the reference node r to the nodes in *G*.

Practically, we cannot use a node outside of the group as a reference because the exact distance between r and a non-leader node in G is unknown. However, we can use a node inside G as a reference to find a local optimum. Figure 7 shows the distances from a reference node to other nodes when a,b,c,d are chosen as the reference, respectively. As mentioned before, the best leader for each line is the median node among the three non-reference nodes, that is, b,d,b, and b, respectively. Hence, we can conclude that node b is likely the best leader candidate. A heuristic to obtain this result is just the use of the median-based criterion for leader election.

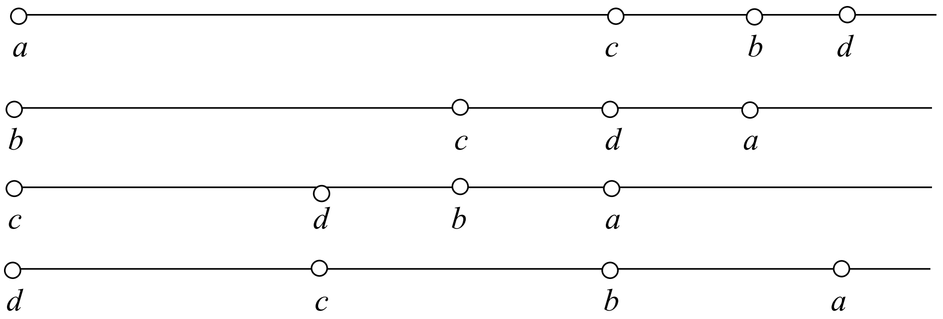


Figure 7. The distances from a reference node to other nodes when a, b, c, d act as the reference, respectively.

In this example, if min-based criterion is used, node c will be chosen as the leader. It is, however, a rather bad choice because most times, node c is the nearest one to the reference, and hence results in great errors.

Though our heuristic algorithm uses only local information, experimental results show that it is quite stable and works well for leader election.

APPENDIX B: CONFIGURATIONS FOR TOPOLOGY GENERATION

All three topology generation models work by first placing a given number of nodes, N, on a two-dimension plane and then generating connections (edges) between nodes. In this paper, we choose the plane size of 10000×10000 distance units.

For the Inet model, we use the default settings of the generator to generate the networks [13].

In the Waxman model, the probability of having an edge between nodes u and v is given by $\alpha \cdot e^{-d(u,v)/\beta L}$, where $d(u,v)$ is the Euclidean distance between u and v, L is the maximum distance between any two nodes, and α and β are two control parameters. Similar to that in previous studies [17], we use $\alpha = 0.0015$ and $\beta = 0.6$. We also compute a spanning tree in the resultant network and add necessary edges to ensure that the final network is a fully connected graph.

The TS topologies are generated using the Georgia Tech Internetwork Topology generator *GT-ITM* [22]. The parameters for this model include: T, the number of transit domains; N_t, the average number of nodes per transit domain; K, the average number of stub domains per transit node; N_s, the average number of nodes per stub domain; E_t, extra transit-stub links; and E_s, extra stub-stub links. We list the settings for our study in Table 2. Note that with these settings, the numbers of nodes generated by *GT-ITM* are actually

3010, 6000 and 12420, respectively. To obtain the specified number of nodes, we randomly choose some leaf nodes and remove them.

	3000-node	6000-node	12000-node
T	10	30	30
N_t	7	8	9
K	6	8	9
N_s	7	3	5
E_t	0	30	40
E_s	0	100	200

Table 2. Parameter settings for the TS topologies.

REFERENCES

[1] K. Kant, R. Iyer, and V. Tewari, "On the Potential of Peer-to-Peer Computing: Classification and Evaluation," Submitted for publication. Available at *http://kkant.ccwebhost.com/download.html*

[2] *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology*, Edited by A. Oram, O'Reilly & Associates, Inc. 2001.

[3] Napster. *http://www.napster.com*

[4] Gnutella. *http://gnutella.wego.com*

[5] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar, "A Novel Server Selection Technique for Improving the Response Time of a Replicated Service," in *Proceedings of IEEE INFOCOM' 98*, April 1998.

[6] K. Moore, J. Cox, and S. Green, "Sonar - a Network Proximity Service," *Internet Draft*, Available at *http://www.netlib.org/utk/projects/sonar/*, February, 1996.

[7] P. Francis, "A Call for an Internet-wide Host Proximity Service (HOPS)," *White Paper*, March 1997. Available at *http://www.ingrid.org/hops/wp.html*

[8] M. Stemm, R. Katz, and S. Seshan, "A Network Measurement Architecture for Adaptive Applications," in *Proceeding of IEEE INFOCOM 2000*, March 2000.

[9] C. Labovitz *et al.*, "Internet Performance Measurement and Analysis Project," Available at *http://www.merit.edu/ipma/*

[10] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "An Architecture for Large-Scale Internet Measurement," *IEEE Communications Magazine*, Vol. 36, No. 8, pp. 48–54, August 1998.

[11] A. Adams et al., "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior," *IEEE Communications Magazine*, pp. 152–158, May 2000.

[12] B. Krishnamurthy, J. Wang, and Y. Xie, "Early Measurements of a Cluster-based Architecture for P2P Systems", *ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[13] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: A Global Internet Host Distance Estimation Service," *IEEE/ACM Transactions on Networking*, October 2001.

[14] J. Nonnenmacher and E. W. Biersack, "Optimal Multicast Feedback," in *Proceedings of IEEE INFOCOM'98*, March 1998.

[15] B. M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, pp. 1617-1622, 1988.

[16] E. W. Zegura, K. Calvert, and S. Bhattacharjee, "How to Model an Internetwork," in *Proceedings of IEEE Infocom 96*, April 1996.

[17] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, L. Zhang, "On the Placement of Internet Instrumentation," in *Proceeding of IEEE INFOCOM'00*, April 2000.

[18] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," in *Proceeding of ACM SIGCOMM' 99*, August 1999.

[19] R. Wolski, "Forecasting Network Performance to Support Dynamic Scheduling," in *Proceedings of the 6th IEEE International Symposium on High Performance Distributed Computing*, 1997.

[20] A. Ballardie, "Core based Trees (CBT version 2) Multicast Routing," *RFC 2189*, September 1997.

[21] D. Estrin, *et al*., "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," *RFC 2362*, June 1998.

[22] Georgia Tech Internetwork Topology Generator *GT-ITM*, http://www.cc.gatech.edu/projects/gtitm

[23] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proceedings of ACM SIGCOMM' 01*, September 2001.

[24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A Scalable Content-Addressable Network," in *Proceedings of ACM SIGCOMM' 01*, September 2001.

[25] S. Deering, "Multicast Routing in a Datagram Internetwork, "Ph.D. Thesis, Stanford University, 1991.

[26] D. Solomon, "Mobile IP: The Internet Unplugged, " Prentice-Hall:Upper Saddle River, NJ, 1998.

[27] M. Harchol-Balter, F. T. Leighton, and D. Lewi, "Resource discovery in distributed networks," *ACM PODC,* 1999.

[28] L. Rogers and D. Williams, "Diffusions, Markov Processes, and Martingales : Foundations, " Cambridge Mathematical Library Press.

[29] Robert L. Carter and Mark E. Crovella, "Server Selection using Dynamic Path Characterization in Wide-Area Networks," in *Proceedings of IEEE INFOCOM'97*, April 1997.

[30] S. Savage *et al.*, "The End-to-End Effects of Internet Path Selection," in *Proceedings of ACM SIGCOMM'99*, September 1999.

[31] V. Cardellini, M. Colajanni, and P.S. Yu, "Dynamic Load Balancing on Web Server Systems," *IEEE Internet Computing*, pp. 28–39, May-June 1999.

[32] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-Aware Overlay Construction and Server Selection," *IEEE INFOCOM*'02, June 2002.