A PEER-TO-PEER NETWORK POSITIONING ARCHITECTURE

Oguz Kaan Onbilger, Shigang Chen, Randy Chow
Dept. of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611, USA

Abstract - Many network-based applications either rely on knowledge of the network topology or can benefit from dynamic adaptation to the changing network environments. This need is particularly essential for the emerging peer-to-peer distributed systems and mobile agent computing. To support efficient implementation of such applications, a network positioning model is necessary for capturing physical/abstract location information of network nodes. Some proposed solutions use a coordinate-based approach to estimate the logical distances among the hosts in the Internet. The coordinate/location information is either provided by a centralized server or derived in a distributed fashion by some landmarks. These approaches often suffer some deficiency in accuracy or scalability. This paper proposes a peer-to-peer architecture to address these problems by completely eliminating the need for landmarks.

1. Introduction

Applications that can benefit from being aware of the underlying topology and the network distance information are numerous. For example, in a web caching system, clients of the system may choose the best location among the alternatives to obtain the cache copies. Similarly, the location information is obviously useful in placing WWW/FTP mirror sites in different areas on the network. In such a replica system, clients may need an automated mechanism to choose the best mirror site based on topology information. The problem of addressing this requirement for such applications is known as the *server selection* problem. Another important application area is in the design of *content addressable networks* [1] and *overlay networks* [2]. These systems require transformation of logical network topologies from an underlying physical network. In general, any *content provider* or *peer-to-peer* (p2p) file sharing facility (i.e., Napster and Gnutella) requires topology information for the best performance, which otherwise may not be observed. Many other topology-aware applications have been explored in literature (please see [3] for an extensive list).

Different from the applications described above, our interest in developing a network positioning model came from the need for supporting a *multiple mobile agent system* (see Section 5).

The proposed positioning model is a p2p coordinate-based system, which maps a physical host location to logical coordinates for efficient on-the-fly logical distance (e.g., communication delay) estimation between any two hosts in the system.

2. Related Work

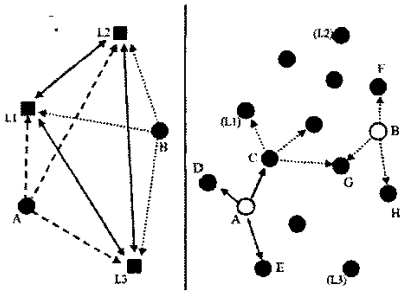
Much like time services in distributed systems, distance services can be provided by some servers or obtained collaboratively by some participating hosts. The IDMaps [4] is an example of the former case. This approach uses triangulation heuristics combined with a centralized, client/server architecture.

Examples of the latter case are the coordinate-based Global Network Positioning (GNP) [5], the Lighthouse [6] and the binning [3] approaches. The GNP and binning strategies use the concept of *landmarks*, which has its roots in [7]. In GNP, coordinates are computed by modeling the Internet as a geometric space. Some hosts in the system are identified as landmarks. These landmarks first measure their distances to each other and then compute their coordinates by minimizing the error between the measured distances and the calculated distances (Please see Step 4 in Section 3.1 for details).

This results in approximate distances as represented by the coordinates among landmarks. An ordinary host can measure its distance to these landmarks and use the landmarks' coordinates to compute its own with the same minimization technique. Fig. 1 (a) illustrates two hosts A and B, which compute their coordinates using the global landmarks L1, L2 and L3. From there on, a host needs only to ask the coordinates of the host of interest to obtain the distance to that host. The GNP is a p2p architecture as oppose to the IDMaps' central servers approach. In the binning strategy, rather than utilizing coordinates, hosts measure their actual distances to the landmarks and place themselves in a bin, which based on the sequence of sorted distances to the landmarks. The binning strategy is not as flexible as the coordinate approaches. This is due to the fact that in the binning strategy bins are tightly bound to the landmarks, whereas in coordinate-based approaches the coordinates and the landmarks are relatively loosely coupled.

The Lighthouse approach [6] shares the same idea of eliminating the landmarks to achieve scalability, as the proposed approach in this paper. The scheme uses a transformation matrix maintained by the hosts for making conversions between the global and the local bases. However, the approach does not address the practical implications, namely, it is not clear how the global basis is formed and the definition of the local basis is not consistent with the positions of the hosts since the hosts are picked arbitrarily.

0-7803-8783-X/04/$20.00 © 2004 IEEE



(a) Global (GNP) (b) Local (TPNP)

Figure 1. Network Positioning Models

Moreover, every host that wants to participate needs to create a local basis. While it is also not clear how to find a host that is currently participating in the system, it does not bring any advantage over GNP in terms of accuracy of the distances.

The results of GNP study [5] show that, the coordinate-based approach provides better estimation of distances than the IDMaps. It is also more accurate and robust due to its p2p architecture when compared to IDMaps. However, this approach has its own drawbacks. First, the landmarks must be globally distributed to the entire Internet to accurately measure the distances between two arbitrary distant hosts. But the distance estimation between two local hosts is biased with the coordinates of the widespread landmarks, which are not local to the given two hosts. Second, these landmarks are central points of failure and may become target of attacks. Third, where to locate the landmarks, how many landmarks there should be and how/where to move the landmarks or add new ones as the Internet topology changes and grows are important open scalablity problems. Finally, there is a security concern of the system. The authors point out that [5] this system may not be suitable in an uncooperative environment since there is nothing to prevent a host from lying about its coordinates for being chosen or not chosen depending on the application. To address these shortcomings we propose the coordinate-based *Pure Peer-to-Peer Network Positioning* (Triple-P NP or *TPNP*) architecture, which completely eliminates the landmarks and the problems they introduce.

3. TPNP APPROACH

The idea of TPNP is similar to the basic principle of dynamic distance vector routing, except that hosts do not need a global picture (i.e. a routing table) of the network. A host that is interested in participating the TPNP system first discovers the nearby hosts that are already in the system to use them as *reference hosts*, which replace the landmarks for the host, in question. Then, the host contacts these references to compute own coordinates (see Section 3.1). Different from Fig.1 (a), Fig.1 (b) illustrates the same hosts A and B's coordinate computation process. A uses three other ordinary hosts C, D, and E already in the system, and B uses F, G, and H locally. These hosts, which are already in the system, are not shown in (a) for clarity. As shown Host C may have used three other locally positioned ordinary hosts for its own coordinates.

Unlike GNP or any other global positioning system local measurements as of TPNP are likely to give more accurate results between local hosts. There are two important facts that support this argument. The first one is that the network traffic of hosts that are close to each other in terms of network distances tend to be routed the same way in most cases. For example, suppose we are interested in measuring the distances between two hosts in Asia. Landmarks, in Europe or North America would not contribute to the accuracy of the estimation. For instance, congestion happened in North America will not affect the end-to-end delay in Asia. In fact, those landmarks at distant locations may have a negative effect on the accuracy. In the global case, we show that the local measurements will have at least, comparable accuracy to that of GNP. For the other fact suppose there are two hosts in the same region (i.e. a country) and the network traffic originated from these two hosts to outside of the region follow different paths, which might have different characteristics (e.g., bandwidth, delay, average load). So, the coordinates of these two hosts that are computed relative to the globally distributed landmarks may seem totally unrelated (i.e. far away). However the hosts may be connected with a reasonably fast metropolitan area network, which makes the distance between the two very small.

Majority of the positioning systems including the ones, which are based on coordinates do not require on-line distance estimations. In fact this is one of the major advantages of coordinate-based systems. To figure out the distance between any two hosts in the network, all we need to know is the coordinates of these hosts. However, the Internet is an extremely dynamic ever-changing environment. Therefore, the coordinates should be recomputed by the hosts periodically in order to reflect the changes so as to use up-to-date distance/delay information. The next subsection presents the algorithm to join into the system. The one to be used to recompute the coordinates by an already participating host is very similar and is not given due to space considerations.

3.1 The Algorithm

The steps of the basic algorithm to join in the TPNP system followed by the hosts are as follows:

1) Discover nearby hosts that are already in the system.
2) Select a subset of the hosts discovered.
3) Measure distances (i.e. delay) to every selected host.
4) Compute own coordinates.
5) Store and advertise own coordinates.

Step 1: DISCOVERY

The discovery algorithm is given as Algorithm Listing 1. The algorithm uses IP-multicast as the discovery mechanism as explained in detail in Section 3.2. Two types of messages are used:

Multicast message: an IP packet indicating a TPNP inquiry with a specified Time-To-Live (TTL) value.

Response message: an IP packet, with a payload of: 1- coordinates of the responding host, 2- coordinates and IP addresses of the reference hosts of the responding host.

The algorithm gradually expands the multicast ring to find necessary number of reference hosts. The first phase of inquiry process takes place in lines 01 through 10. If the pre-determined number of hosts cannot be found and the threshold value for the multicast depth is reached, lines 14 through 18 are executed to check whether at least one reference host was found. If this is the case further references of the reference hosts are also to be used by the current host. If the current host which is willing to participate in the system is not lucky which means that there is no candidate reference hosts found so far, then multicast ring needs to be further expanded by using multicast depth values between d_{thresh} and d_{max}.

```
DEFINE
   Hr : set of hosts that replied to the current TPNP inquiry
   Hn: set of newly discovered hosts responded to the current TPNP
   inquiry
   N_TPNP : number of reference hosts to be used globally in TPNP
   d : multicast depth
   d_init : predetermined initial value of d
   d_thresh : predetermined threshold value of d
   d_max : predetermined maximum value of d
   t_inq : response wait time for the inquiries

01   Hr = φ;
02   d = d_init;
03   REPEAT
04      Hn = φ;
05      multicast an inquiry message with TTL = d;
06      wait(t_inq);
07      check responses;
08      Hr = Hr ∪ Hn;
09      increment(d);
10   UNTIL |Hr| ≥ N_TPNP   OR   d > d_thresh;
11   IF |Hr| ≥ N_TPNP
12      RETURN(Hr);
13   ELSE
14      FOR EACH (i) responding host in Hr
15         FOR EACH (j) reference host used by Hr_i
16            IF Hr_ij ∉ Hr
17               Contact Hr_ij and obtain reference host info;
18               Hr = Hr ∪ {Hr_ij}
19      IF Hr = φ
20         REPEAT
21            Hn = φ;
22            multicast an inquiry message with TTL = d;
23            wait(t_inq);
24            check responses;
25            Hr = Hr ∪ Hn;
26            increment(d);
27         UNTIL |Hr| ≥ 1   OR   d > d_max;
28         IF |Hr| ≥ 1
29            EXECUTE 14 through 18
30            RETURN(Hr);
31         ELSE
32            // No participating hosts found in the vicinity,
33            // Contact DNS for global host information.
```

Algorithm Listing 1

The same procedure is then repeated in lines 20 through 27. If there is still no response from any hosts, then the algorithm gives up. The DNS needs to be contacted (see Step 5) to figure out some global hosts, which are participating in the system, since there is no local participating host in the vicinity.

Step 2: SELECTION

The selection of reference hosts from the candidates determined by the discovery procedure above is presented in the Algorithm Listing 2. The algorithm uses a heuristics to perform its job as explained below.

IF $|Hr| = N_{TPNP}$
Hr is the final set of hosts to be used as references;

IF $|Hr| \geq N_{TPNP}$
apply the selection heuristics:

a) Pick the hosts, which reside in outside domains or use greatest number of hosts in different outside domains as their own reference hosts.
b) Pick the hosts, which are closest to origin in Euclidean space (i.e. the ones with absolute smaller coordinate values).
c) Pick the closest hosts (i.e. quick responding hosts).

and prepare the final set of hosts as references.

Algorithm Listing 2

(a) in Algorithm Listing 2 tries to ensure that the coordinates converge not only locally but also globally. (b) tries to ensure that the system/coordinates converges towards a fixed point to prevent chaos. Coordinates may go out of control over time due to absence of fixed hosts (e.g., landmarks), accumulated error terms and periodic recalculations. Similar to the concept of combining GMT and atomic clock for providing global time and preserving accuracy of time dissemination, we use a single point that never changes its coordinates to solve the similar problem. For example, this point can be set to *O(0, 0)* as the origin, assuming a two-dimensional space. A computer engineering department of a university, for example, may configure all its hosts to serve this purpose. Except hosts closest to the origin no other host needs to contact to this origin. When computing the coordinates for the fist time, a host can choose the reference points that are closer to the origin as compared with the others. In recomputing coordinates, they may also consider their own coordinates to select the reference points with respect to the origin.

(c) tries to minimize overhead into the network due to distance measurements. The discovery part of the algorithm (Step 1) guarantees (c) and therefore is already taken into account in the first two. So, it is applied when the first two cannot make a decision.

Step 3: MEASURING DISTANCES

The distances (delay) are measured using ICMP ping. However as it is very well known ping can easily be abused. The protocol below was designed to use ICMP ping in a secure manner for TPNP. The Source in the protocol is the host, which is willing to join in the TPNP system. The Target is one of the reference hosts (nodes) used by the source host.

Distance Measurement Protocol:

1) Source connects to Target and asks for ping permission.
2) If the Target grants the permission, it returns a random ephemeral port number to be used for ping along with a one-time password to Source. If Target is busy serving

other requests, it may simply respond with a "contact later" message.

3) Target opens the ping port by starting the ping server listening on the ping port.
4) Source pings the Target on the ping port. The ping packets include the one-time password.
5) Target checks the password and if valid responds to the request, otherwise the packet is discarded.
6) After the predetermined number of ping requests received or time-out occurred, Target closes the ping port.

Steps 4 and 5 are repeated for a predetermined number of times. Out of these attempts, the minimum of the round-trip time measures is taken as the distance between the two hosts.

Step 4: COMPUTING COORDINATES

A host which is willing to compute its own coordinates needs the coordinates of its reference hosts and distances to these hosts. All this information is now ready as the result of the previous steps. TPNP, for the time being, follows the coordinate computation process of GNP [5]. This process is a non-linear computation that involves a minimization function. It is subject to further research to use linear functions similar to the ones proposed in [8] and [9], in TPNP.

The coordinate computation process tries to find a solution to a multi-dimensional minimization problem. It is formally the minimization of the objective function

$$f(.) = \sum_{i,j} \varepsilon(m_{ij}, p_{ij})$$

where ε is an error measurement function, m_{ij} and p_{ij} are measured and predicted distances among the hosts.
The error measurement function is the normalized error measure, which was rated the best in [5]:

$$\varepsilon(m_{ij}, p_{ij}) = \left((m_{ij} - p_{ij})/m_{ij}\right)^2$$

The inputs to the process are measured distances obtained in the previous step in the form of a matrix and coordinates of the hosts obtained in Step 1. At each iteration minimization function is computed and new computed coordinates are assigned to as the host's coordinates and used by the following iteration. Iterations continue until the error is reduced below a predetermined minimal value. Final coordinate values minimize the overall error between the computed and measured distances from the current host to the chosen reference hosts.

Step 5: STORING AND ADVERTISING COORDINATES

A host, which completed the previous steps should store its coordinates to make them available to the outside world. The coordinates may be required by the other hosts for two purposes. First one is for applications, which would benefit and therefore use distance information between the hosts in the Internet. The second one is that other hosts that would like to join in TPNP may request coordinate information from the current host. The same is true for hosts, which may need to recompute their coordinates.

In addition to making the coordinates available for application purposes on the host, DNS can be used to provide host-name/location or IP-address/location resolution that would eliminate the need to contact hosts for coordinate information.

Lastly, the current host should join in to the IP-multicast group reserved for TPNP to receive and reply TPNP requests from other hosts, which might like to join in TPNP afterwards. The host should also prepare for handling ICMP ping requests as explained in Step 3.

3.2 Starting the system

One of the major issues in realizing TPNP is how to start up the system. In most server-based distributed systems, servers are first configured and then started running. Problems such as load balancing, scalability, proximity and fault-tolerance arise later and continue throughout the lifetime of the system. Within TPNP however, the opposite is true. Once the system is started the problems mentioned diminishes and even disappears over time with the increase in the number of participating hosts. There are two ways to start up the system, namely, sequential joining from the beginning and participation of a set of hosts in parallel. The former scheme is used in our simulations and explained in Section 4. The latter can be achieved by simultaneous participation from fixed sites (such as universities) with centrally and statically computed coordinates as in the GNP approach. However, in TPNP this is to be done only once, and these hosts might even cease to exist over time with no effect on the system. Furthermore, the number of such hosts needs not be greater than the number of reference points desired.

One way to figure out the participating hosts in the system is to use IP multicast as used by the algorithm given in the previous section. It would be safe to assume that the first responses should be coming from the closest hosts. This scheme fits nicely into the local distance estimation in TPNP because it is very easy to find the closest hosts using multicast. Otherwise, it would not make sense to talk about "closest hosts" when we are already discussing a distance estimation scheme. One nice side effect of the peer finding process in TPNP is the fact that every host discovered by a new participant reveals more hosts giving more than enough nearby hosts for references.

The alternative is to use DNS to explore the participating nearby hosts in the system by exploiting the Local and Authoritative Name Servers. However, although DNS hierarchy gives clues, it is not easy to find the distant hosts especially during the early stages of the system. Further research is necessary to determine how best DNS could be as an alternative with minimal changes and overhead to the existing system.

3.3 Scalability Issues

Contrast to the common scalability problem in distributed systems, system growth has a positive affect on the TPNP system due to its p2p architecture. The GNP approach is clearly more scalable when compared to a centralized client/server model like that of IDMaps. However landmarks,

which are cornerstones in GNP introduces other important scalability problems as explained in Section 2. Since the TPNP eliminates the landmarks completely, these problems disappear altogether.

However, use of multicast as the discovery mechanism introduces load into the underlying network. But this overhead diminishes and even disappears as the number of hosts participated in the system increases. In fact, in most cases, the overhead of multicast is observed when a new host wants to join in the system. Even then, if the number of the nearby hosts, which are already in the system, is no less than the required number of hosts, the multicast overhead will be minimal. Further simulation experiments are underway to provide quantitative analysis of this issue in addition to those of Section 4. For the same reason, hosts do not have to measure their distances to distant landmarks which will reduce the overhead introduced into the network. Therefore, it can be said that the approach is *positively scalable*, in terms of the load introduced into the network for both multicast and delay measurements.

3.4 Security Considerations

The TPNP approach also addresses the security of the system. The security aspect of TPNP is two-fold. First, as pointed out before, GNP cannot prevent hosts from lying about their coordinates. For TPNP, because hosts compute their coordinates relative to nearby hosts, and these nearby hosts have to provide their reference points along with the coordinates, it is possible to check using this information whether a host's coordinates are correct. Heuristics can be devised to verify the correctness of the host coordinates by asking the coordinates of the reference hosts of the target host. Several levels of security can be provided at the expense of communication overhead by increasing or decreasing the number of hosts to check.

The second security problem is related to landmarks, which may become target of attacks since they are to answer each ping request directed to them. The problem is more prevalent for TPNP since each participating host would have to answer any ping request coming from any host in the Internet. The solution we propose is to customize the use of ping. The Distance Measurement Protocol in Section 3.1 serves this purpose.

4. Experimental Evaluation

We have conducted simulation experiments to figure out how TPNP will perform, that is how accurate the predicted distances among the hosts will be in the system.

4.1 Simulation Environment

Our simulation system employs a hierarchical structure, which models individual Internet domains either as an Inet or a Waxman topology. Every domain in a given topology has an equal probability of having either a Waxman or Inet model, and has any number of nodes ranging from 20 to 200. The borders between domains use the Inet, which provides a power-law topology. Earlier studies have shown that the Internet follows a power-law topology [10], [11]. A more recent study [3] related with ours also confirms that results obtained with power-law topologies better match with the actual Internet traces.

To apply GNP in our simulation environment and to compare with TPNP we have integrated the GNP software by Ng and Zhang [5]. (GNP software is available from http//:www-2.cs.cmu.edu/~eugeneng/research/gnp.) We have used the software for GNP computations and for TPNP where possible, without modifications other than the ones necessary for integration.

4.2 Simulation Parameters

In our experiments, we used 10 landmarks for GNP and 10 reference nodes (peers) in TPNP (except the first 10 nodes to join). The number of dimensions is two for both. It is important to note that the choice of ten landmarks and two-dimensional coordinate system is for simulation simplicity and efficiency. Our aim is to compare the two approaches and therefore our first priority in selecting parameters is to ensure fairness. Using only two dimensions has the advantage of much less computation overhead. When the number of landmarks and reference points increases, both approaches benefit and GNP benefits more since the results with two-dimensional space are not very accurate and there is more room for improvement. As reported in [5], there is a saturation point for both number of dimensions and reference nodes. We confirm the results for GNP and report that the same is true for TPNP. For prediction accuracy, we use the same performance metric in [5], which is the following:

$$\text{Relative Error} = \frac{|\,measured - estimated\,|}{\min(measured, estimated)}$$

Our choice of the metric as opposed to the simple percentage or simple ratio metrics, which are used in [12], [3], [6] is due to the fact that while percentage error metric hides underestimates, ratio error is not suitable to compute average undirectional error due to the magnitude difference of over and underestimates.

4.3 Simulation Strategy

Our simulation strategy is to obtain the coordinates of all the hosts in the topology and therefore estimations for all the shortest path distances among every pair of nodes. For TPNP, we have used the downhill-simplex method [5], which used for GNP for minimizations.

To select the best possible nodes to be landmarks for GNP, we implemented the N-cluster-medians technique, which was rated the best [5]. We applied the technique to the whole network to find the nodes, which represent all the nodes with an aggregation based on proximity. It should be noted that this technique is only feasible in a simulation environment and in favor of GNP.

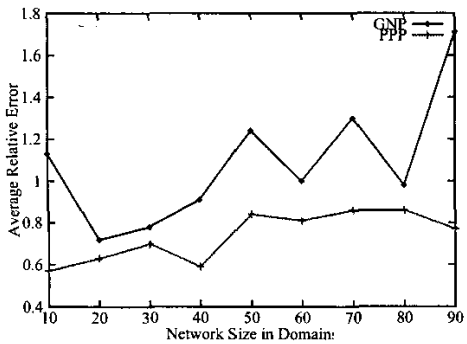


Figure 2. Average Relative Error

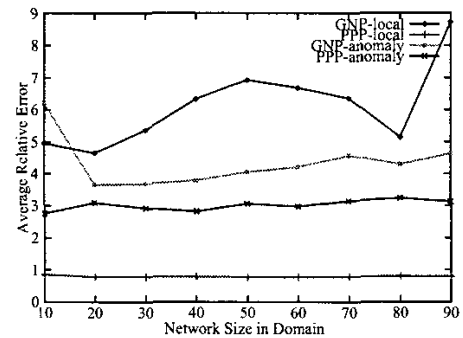


Figure 3. Average Local and Outliers Relative Error

All of the results presented in this paper were obtained by using three different topologies of roughly the same size and average values of the three simulation runs are reported. This is to make sure that the results are repeatable which is especially important for TPNP's stochastic behavior due to random node join order.

To simulate the actual formation of the TPNP system, we pick a random node on the whole network and assign the coordinates of origin *(0,0)* of the Euclidean system for convenience. Since we are using a 2D Euclidean system, only the second and the third nodes' coordinate computation process differs from the rest. For these two nodes, we measure distance of shortest paths between them and previously joined nodes. We place the second node on the x-axis and pick the positive value for the third node among the alternatives for convenience. For the rest of the nodes, the same minimization technique of GNP is used, however, 4^{th} through the 10^{th} nodes use the available number of reference points while the rest use exactly 10 of them.

4.4 Simulation Results and Analysis

As seen in Fig. 2, when the number of domains increases, prediction performance of GNP is negatively affected. This is the case even though the landmarks are chosen using the N-cluster-medians of the entire network.

It is clear from the figure that using a pure p2p approach without landmarks, better accuracy can be achieved. This is again due to the fact that global landmarks are not adequate for local distance estimations.

In Fig. 3 we have plotted the intra-domain distance predictions of the both approaches with increasing network sizes. Fig. 3 shows clearly that the local (intra-domain) relative error remains constant in TPNP regardless of network size and the number of domains. However GNP's intra-domain relative error is correlated with network size. Moreover, even with a fairly small network size of only one thousand nodes, which is the first point in Fig. 3 with 10 domains, GNP's accuracy is still not as good.

Although excluding outliers in statistical data has merit, in practice it is desirable for a system to be as fair as possible to its users or applications. Since we expect some form of uniformity from TPNP, we also plotted the graphs for average relative error of values greater than 1, which we call *anomaly* in Fig. 3. As expected, TPNP provided the uniformity of anomalies with also smaller error terms when compared to GNP. As it is clear from the figure, GNP has larger anomalies, which scale with the network size.

Within TPNP, due to cascading coordinate computation, one might expect that the predicted distances would be less accurate due to accumulated errors when the number of hops (peers) increases between any given two hosts. To provide an insight, we plotted the graphs in Fig. 4. These graphs have been obtained from 7000-node topologies.

Fig. 4 (a) shows relative error distribution to the path latency intervals. We created 200 ms intervals of path latency starting from zero and plotted the prediction errors of all the paths whose measured distances fall in that interval. Fig. 4 (b) shows the distribution of paths into these latency intervals in the

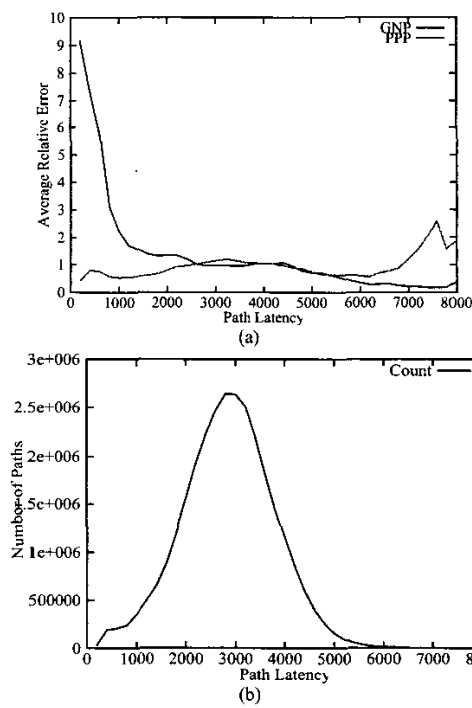


Figure 4. Distribution of Average Relative Error to Path Latencies

network. With an average of 100 ms of hop latencies in the network a slight increase arises at 7000 ms point for TPNP. This means that error accumulation due to many predictions across the network due to the p2p nature of TPNP, may not be significant. However, until the 2500 ms point, GNP predictions of shorter paths are not as good. GNP's accuracy clearly increases when the path latency increases due to very well-positioned landmarks in the simulation environment, but the number of paths in this upper range is fairly small as seen in Fig. 4 (b). Additional experiments are underway to provide a closer analysis for this issue.

5. Applying Tpnp To Mobile Agent Computing

Mobile Agent (MA) computing is a promising technology that offers many interesting applications in addition to traditional distributed client/server computing [13]. An MA is an autonomous entity, which is composed of code, data and state information. They visit hosts (e.g., servers) possibly using an itinerary, perform some execution on those hosts and migrate with their state information from host to host. However MAs are not lightweight entities.

The classical MA model uses a single agent to perform a single task. There are MA applications, which requires an itinerary but do not impose an order of the hosts to be visited in the itinerary, such as e-commerce, software distribution, and information retrieval especially from censors across a network. Our goal is to show that an MA with an itinerary, which is aware of the underlying network structure via distance measurements, can perform its task in much less time and preserve bandwidth. We have applied TPNP together with a simple, basic Nearest Neighbor (NN) heuristic to find a near-optimal tour (i.e. itinerary) for an MA, which is an instance of the well-known Traveling Salesperson Problem (TSP). The simulation results are given in Table 1. The first topology refers to a 50-domain 5174-node network. The second one is a 50-domain 4700-node network; both created using the same simulation environment of Section 4. We created an itinerary of randomly chosen 100 hosts for both topologies. The columns show the average of 100 measurements of tour lengths of random order, the near-optimal NN tour length with the TPNP predicted distances and the actual measured tour length of the corresponding near-optimal NN tour, respectively. The actual distance that the MA will travel is given in the last column. As it is shown in the table, the simple NN heuristic and TPNP predicted distances yield an actual tour length, which is roughly one third of the average tour length of randomly chosen tours (the second and last columns).

In the multiple MAs model a group of autonomous MAs perform a single well-defined task [13]. In this model, TPNP not only addresses the performance problem, but also provides context-awareness that could be extremely useful addressing security problems of MAs.

TABLE 1
Application Of Tpnp And Tsp Into Classical MA Model

Topology	Random Order (ms)	Predicted NN (ms)	Measured NN (ms)
I	277,323	48,764	96,165
II	285,301	59,701	94,697

6. Conclusion

In this paper we presented a p2p coordinate-based network position estimation scheme that does not rely on landmarks for coordinate computation as in previously proposed approaches. The goal of eliminating the landmarks is to achieve greater scalability. The overall architecture is presented and the goal justified. The simulation analysis also shows that the system performance in terms of local distance prediction accuracy can be significantly improved over GNP. Even with a two-dimensional Euclidean space its overall performance is within an acceptable range. The pure p2p nature of the architecture lends itself very well with the robustness of the system. Some security issues are also addressed. Finally, it is shown how MA computing can benefit from being network-aware as can be provided with TPNP.

References

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A Scalable Content-Addressable Network", *Proc. SIGCOMM 2001*, Aug. 2001.

[2] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", *Proc. SIGCOMM 2001*, Aug. 2001.

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

[4] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, L. Zhang, "IDMaps: a Global Internet Host Distance Estimation Service", *IEEE/ACM Trans. on Networking*, Vol. 9, No 5, pp. 525-540, Oct. 2002.

[5] T. S. Eugene Ng, H. Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches", *Proc. IEEE INFOCOM 2002*, pp. 170-179, June 2002.

[6] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, S. Bhatti, "Lighthouses for Scalable Distributed Location," *Proc. 2nd Int. Workshop on P2P Systems*, Feb. 2003.

[7] S. M. Hotz, "Routing Information Organization to support scalable interdomain routing with heterogeneous path requirements," 1994, PhD Thesis, University of Southern California.

[8] H. Lim, J.C. Hou, C-H. Choi, "Constructing Internet Coordinate System Based on Delay Measurement", *Proc. Internet Measurement Conference*, pp. 129-142, October 2003.

[9] L. Tang and M. Crovella, "Virtual Landmarks for the Internet", *Proc. Internet Measurement Conference*, pp. 143-152, October 2003.

[10] M. Faloutsos, P. Faloutsos, C. Faloutsos, "On power-law Relationships of the Internet Topology", *Proc. SIGCOMM'99*, Sep. 1999.

[11] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, W. Willinger, "Network Topologies, Power Laws and Hierarchy," Tech. Rep. TR01-746, Technical Report, University of Southern California, 2001.

[12] K. P. Gummadi, S. Saroui, S. D. Gribble, "King: Estimating Latency between Arbitrary Internet End Hosts", *Proc. (SIGCOMM) Internet Measurement Workshop*, Nov. 2002.

[13] O. K. Onbilger, R. Chow and R. Newman, "Remote Digital Signing with Mobile Agents", *Proc. Second International Workshop for Asian Public Key Infrastructures 2002*, pp. 123-130, Nov. 2002.