Improving Delaunay Triangulation for Application-level Multicast

Wan-Ching Wong S.-H. Gary Chan
Department of Computer Science,
The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon Hong Kong
Email: {wwilliam, gchan}@cs.ust.hk

Abstract— **In recent years, there has been increasing interest in *application-level multicast (ALM)*, where the multicast related functionalities are moved to end-hosts. One of the promising ALM protocols is *Delaunay Triangulation (DT)*, which constructs an overlay mesh using 2-D Delaunay Triangulation (DT) and makes use of compass routing to forward packets. However, DT protocol as it is originally proposed suffers from several weaknesses: 1) it requires users to input its geographic location, and assumes that the location correlates well with network distance; 2) it tends to form multiple connections across two domains, and hence has a high usage of long delay (interdomain) links; 3) it does not consider the fanout of a host, therefore some less-powerful hosts may serve too many users, leading to degradation of service. To address these problems, we propose to use *Global Network Positioning (GNP)* for host location estimation and *Forward Delegation* to limit the fanout of a host explicitly and efficiently trade off the network resource usage with latency. Using Internet-like topologies, we show that our scheme, as compared to the original DT protocol, can substantially reduce average relative delay penalty, physical link stresses and network resource usage while meeting the processing capability of the hosts in the network.**

I. Introduction

Despite its proposal more than a decade ago, IP multicast is still not widely deployed nowadays. This is mainly due to its many technical and implementation difficulties [1] [2]. In order to overcome these difficulties, researchers recently have been focusing on enabling multicast at the application layer, the so-called application-level multicast (ALM). In ALM, the multicast-related functionalities such as packet forwarding and reliability are shifted from the network layer to the end hosts in the multicast group. A recent approach based on *Delaunay Triangulation (DT)* emerges as a promising ALM protocol. In DT, a host only needs to maintain local information for packet forwarding, making it scalable to large groups [3]. DT protocol constructs an overlay mesh using 2-D Delaunay Triangulation based on the host locations, and makes use of compass routing to identify one's children and parent for data delivery [4], [5].

However, DT protocol as it is originally proposed still suffers several weaknesses:

- Inaccurate estimation of host location: DT protocol requires users to input their geographic coordinates, and assumes the differences between the coordinates correlate well with network latency. This is certainly true for wireless network, but often not the case for the Internet;
- High network resource consumption: In general, interdomain distances are much longer than intradomain distances. In fowarding packets from one host, say u, to two other hosts in another domain using compass routing, two long-haul connections usually would be set up (in which u forwards packets to the two other hosts individually).
- Unlimited fanout of a host: Compass routing does not take the fanout of a host into consideration. As a result, a host may forward a packet to many hosts and hence can be heavily loaded.

To address the above issues, we hence propose to use the following two algorithms:

- *Global Network Positioning (GNP)*: Instead of using geographic coordinates, we propose to use GNP which estimates the host locations in the GNP space with several well-known landmarks in the Internet [6]. It has been shown that the distances between hosts in GNP space are highly correlated with the latency in the Internet.
- *Forwarding Delegation:* It balances the loads among hosts by limiting the fanout of a host according to its capability. In order to reduce network resource usage, it also aggregates long-delay (inter-domain) paths and delegates the forwarding mechanism to another host.

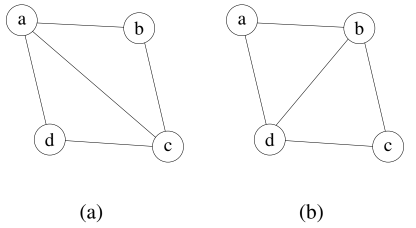
Many ALM protocols, such as HMTP [7], Narada [8] and Scribe [9] require hosts in the multicast group to periodically probe other hosts to gradually improve their performance in term of end-to-end delay or network resource usage. Our scheme, as opposed to these, requires a host to probe only a small number of landmarks in the Internet when it joins. Through these probing a host estimates its location in a *logical* space for efficient packet routing. Therefore, our scheme has lower overheads while keeping its performance comparable to that of other schemes. M-CAN is similar to our scheme in the sense that a host also estimates its location in a logical space through probing when it joins [10]. However, the number of possible host locations in MCAN is finite, while that in our scheme is infinite; therefore we can provide a much better estimation of host locations.

The paper is organized as follows. We first review the traditional DT protocol and compass routing in Section II. Then we discuss how the location of a host can be estimated based on GNP in Section III, and present the algorithm of forwarding delegation in Section IV. We finally present some illustrative simulation results in Section V and conclude in Section VI.

II. Review

In DT protocol, each host has a *geographical* coordinate and hosts first form an overlay mesh based on these coordinates. Compass routing is used to route a packet from one point to another. DT protocol connects the nodes together in a triangular manner so that the mesh satisfies the Delaunay Triangulation property, i.e., the minimum internal angle of the adjacent triangles in the mesh are maximized [5], [11]. To illustrate the triangulation process, consider that points a, b, c and d form a convex quadrilateral $abcd$. There are two possible ways to triangulate it as shown in Figure 1. Since the minimum internal angle of $\triangle abc$ and $\triangle acd$ (Figure 1(a)) is less than that of $\triangle abd$ and

This work is supported, in part, by the Competitive Earmarked Research Grant (HKUST 6155/03E) and Direct Allocation Grant (DAG02/03 EG41) of the Research Grant Council in Hong Kong.



(a) (b)

Fig. 1. a) Two adjacent triangles forming a convex quadrilateral $\triangle abd$ and $\triangle bdc$ violating the DT property. b) Restoration of DT property by disconnecting a from c and connecting b and d.

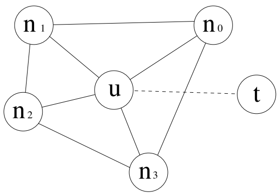


Fig. 2. When u receives a unicast packet with destination t, it forwards the packet to n_0. When u receives a multicast with source t, it forwards the packet to n_1 and n_3.

$\triangle bcd$ (Figure 1(b)), DT protocol transforms the former configuration into the latter. One strength of forming the mesh this way is that it connects those geographically close nodes together.

In traditional DT protocol, the joining process is bootstrapped by a DT server, which caches a list of joined host. A joining host u first queries the DT server for some already-joined hosts. u then sends via unicast join requests to these hosts, all of which in turn send back those joining requests *along the DT mesh* towards u until reaching a set of hosts nearby u. These nearby hosts then connect to u, forming a mesh. Note that the newly established connections may violate the DT property. To restore it, every host periodically tests its connections against the property, and drops those failing in the test. A host also discover nearby hosts through periodical exchange of control messages with its neighbors, and connects to any nearby hosts if this does not violate the DT property.

To illustrate these processes, suppose that initially hosts a, b, c and d are connected as in Figure 1(a). Because the connection ac violates the DT property, it is dropped when host a or c tests it. After that host b discovers d through control messaging, and connects to d since bd satisfies the DT property. Then the resultant overlay mesh (as shown in Figure 1(b)) satisfies the DT property.

To route a packet from one point to another in DT mesh, compass routing can be used. When host u receives a unicast packet with destination v, it first computes the slope between the two coordinate points u and v. Let the slope be s. u then computes the slopes of all its N neighbors with itself. Let these be s_i, $1 \leq i \leq N$. u then forwards the packet to the neighbor whose slope is the closest to s, i.e., u forwards to neighbor j where $|s_j - s|$ is the minimum for all s_i's. We illustrate the compass routing in Figure 2, where u needs to route a packet destined to t to one of its neighbors. Because the slope between u and n_0 is the closest to the slope between u and t, host u forwards the unicast packet to host n_0.

Reverse path forwarding algorithm is used to *multicast* packets in the DT mesh. When host u receives a multicast packet from source s, it forwards the packet to those neighbors if u is on the path from them to s. Refer back to Figure 2 again. When host u receives a multicast packet with source t, it forwards the packet to n_1 and n_2. This is because the slope of $t \rightarrow n_1$ is the closest to slope $u \rightarrow n_1$ among all the slopes of $u \rightarrow n_1$, $n_0 \rightarrow n_1$ and $n_2 \rightarrow n_1$, while the slope of $t \rightarrow n_2$ is the closest to slope of $u \rightarrow n_2$ among the slopes of $u \rightarrow n_2$, $n_1 \rightarrow n_2$ and $n_3 \rightarrow n_2$.

III. Accurate Estimation of Host Locations using GNP

GNP has been proposed in [6] to estimate the relative location of a host in the Internet based on measured network delays, such that the difference between the locations of two hosts correlates well with the round trip time between them. Our scheme makes use of GNP to estimate host locations to construct its mesh. To the best of our knowledge, it is the first time GNP is applied in this context.

In GNP, a number of infrastructure hosts, termed as *landmarks*, are used as reference points for measurement purposes. The landmarks, after measuring the round-trip time among themselves, forward the measurement results to one of the landmarks, which computes the landmark locations in the GNP (or *Internet*) space by minimizing and objective function based on the measurements. The landmark locations are then disseminated back to the respective landmarks. Specifically, to estimate the locations of M landmarks, the following objective function is minimized:[1]

$$J_{landmark}(L_1, L_2, \ldots, L_M) = \sum_{L_i, L_j \in \{L_1, \ldots, L_M\} | i<j} \left(\|L_i - L_j\| - RTT(L_i, L_j) \right)^2 , \qquad (1)$$

where L_i and L_j, $1 \leq ilj \leq M$, are the 2D coordinates of two landmarks in GNP space (i.e., $L_i = (x_i, y_i)$, and $L_j = (x_j, y_j)$) to be found and $RTT(L_i, L_j)$ is the round-trip time between landmarks L_i and L_j. Clearly, $J_{landmark}$ is the sum of the differences between the measured network distances (i.e., round-trip time) and the logical distances in the GNP space among landmarks. Therefore, we seek to find a set of landmark locations such that the sum is minimized. Note that although there are infinite sets of $\{L_1, L_1, \ldots, L_M\}$ to minimize $J_{landmark}$, any one of them would be equally good to represent landmark locations.

Given the landmark locations, a new host joining the mesh estimates its location by similarly minimizing another objection function given by:

$$J_{host}(u) = \sum_{L_i \in \{L_1, \ldots, L_M\}} \left(\|u - L_i\| - RTT(u, L_i) \right)^2 , \qquad (2)$$

where u is the desired host location and $RTT(u, L_i)$ is the measured round-trip time between host u and landmark L_i. Note that the landmarks do not have to be permanent. In fact, it is trivial to update Equation (2) lest a landmark fails by removing the landmark from the set. Furthermore, backup landmarks can be set up at any time. Their locations can be calculated according to Equation (2) and multicast to the other hosts in the overlay.

Overloading of a landmark also is not an a serios issue for a rather stable mesh because the workload of a landmark depends on how frequent a host joins the mesh. Furthermore, a landmark is not required to store any information or perform any computation upon the arrival of a new host as computation (Equation (2)) is done at the end-host.

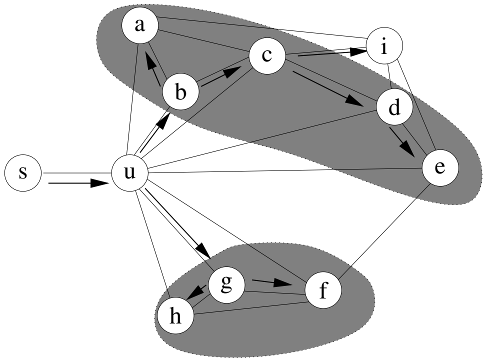


Fig. 3. Host u clusters its children into two groups to limit its maximum fanout and reduce network resource consumption.

IV. Forwarding Delegation

As mentioned before, the traditional DT protocol may result in high network resource usage. For example, if host a belong to domain A and hosts b and b' belong to domain B, then in general the delay of interdomain links ab and ab' are much longer than that of intradomain link bb'. Therefore, angle $\angle bab'$ is likely to be small. Using compass routing, if either b or b' is a child of a, the other one is likely to be a child of a also. In this case, two independent end-to-end connections across domain A and B are set up, which leads to a high usage of long delay (inter-domain) links and hence high network resource usage. Moreover, the traditional DT protocol may result in hosts with too many outgoing links, making the hosts bandwidth bottlenecks.

These problems can be solved if a host carefully selects its representative children and delegates to them the forwarding of the unselected children. The loading of hosts can hence be controlled by the number of selected children, and the usage of long-delay (inter-domain) links can be reduced by aggregating paths with small adjacent angles. Basically, when host u receives a multicast packet from host s, it first groups its children into several clusters and selects the closest child as the representative child in each cluster. Then host u, for each cluster, forwards the multicast packet to the representative child with a list of unselected children y_is, termed *delegation list*, which is ordered by the slopes $u \rightarrow y_i$. Moreover, host u also checks the *delegation list* embedded in the packet. If the ranking of host u is i in the list, it forwards the packet to the hosts with ranking $(i-1)$ and $(i+1)$ if they are not host s.

We show an example in Figure 3, where only a part of DT mesh is shown and the maximum fanout of host u is two. When u receives a multicast packet from host s, it groups its children $a, b, \ldots, h$ into two clusters $a-e$ and $f-h$, and forwards the packet to the closest child in each cluster (i.e., b and g). The *delegation list* embedded in the packet from u to b is $[a, b, c, d, e]$, while that of the packet from u to g is $[f, g, h]$. Therefore, when b receives the packet from u, it forwards to a and c. However, when c receives the packet from b, it forwards to d, but does not forwards the packet backward to b. Furthermore, c in turn, also forwards the packet to its child i.

In forward delegation, host u groups its children into several clusters with a hierarchical clustering algorithm (in ALGORITHM I). Initially, each child belongs to an independent group. Then, in each iteration, a pair of children from different clusters, c_i and c_{i+1}, is selected such that $\angle c_i u c_{i+1}$ is the minimum angle among all the possible pairs (i.e., $\angle c_i u c_{i+1} = \min_{c_j \in X, c_{j+1} \in Y, X \neq Y}(\angle c_j u c_{j+1})$).

ALGORITHM I

```
CHILDGROUPING($u$)
  Each child, $c_i$, belongs to an independent cluster.
  repeat
        $(c_i, c_{i+1}) \leftarrow$ a pair of children of different
                          clusters with the minimum
                          adjacent angle.
        if $\angle c_i u c_{i+1} < T$ OR
            (number of clusters $+$
             number of delegated forwardings) $> K$
           then Merge the groups of $c_i$ and $c_{i+1}$
           else return
    until
```

If the number of clusters plus the number of delegated forwarding (i.e., $b \rightarrow a$, $b \rightarrow c$, $c \rightarrow d$, etc. in Figure 3) exceeds its fanout limit given by K, or $\angle c_i u c_{i+1}$ is smaller than a certain threshold T, then host u merges the groups for c_i and c_{i+1}; otherwise the algorithm returns.

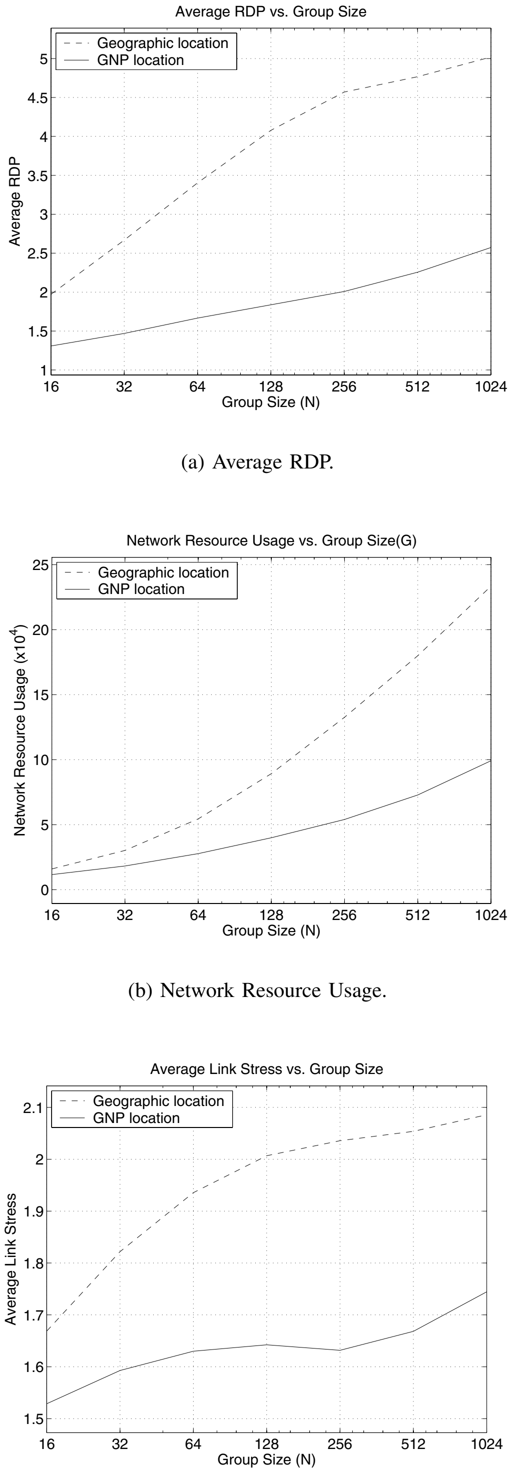
V. Illustrative Numerical Results

We have used simulations to evaluate the performance of our scheme. First, we generate a number (10) of *Transit Stub* topologies with Georgia Tech's random graph generator [12]. The parameters used for topology generation are according to the study of the traditional DT protocol in [3]. The generated topologies are a two-layer hierarchy of transit networks (with four transit domains, each with 16 randomly-distributed routers on a 1024×1024 grid) and stub networks (with 64 domains, each with 15 randomly-distributed routers on a 32×32 grid), where a host is connected to a stub router via a LAN (of 4×4 grid). The delays of LAN links are 1ms while the delays of core links are computed by the topology generator.

The baseline parameters of our scheme are $N = 128$ (i.e., a total of 128 hosts join the multicast group), $K = 6$ (i.e., the maximum fanout of every host is equal to or below 6.), and $T = 5^o$ (i.e, the adjacent angles among children should be larger than 5^o.) Based on this set of parameters, we first study the performance of GNP in location estimation. As a comparison, we take the geographic coordinates generated by the topology generator as host locations (the naive policy). For GNP we select a number (20) of landmarks based on N-cluster-median criterion as given in [6]. For each DT mesh, we randomly select a host as the source and send packets along the DT mesh to all hosts with compass routing. We compare the two location estimation schemes (naive and GNP) with the following performance metrics: 1) relative delay penalty (RDP), defined as the ratio between overlay delay to underlay delay of a host from the source, and 2) physical link stress, defined as the number of duplicated packets transmitted through a given physical link, and 3) network resource usage, defined as the total link delays covered by an overlay tree.

Figures 4(a) and 4(b) show the average RDP and network resource usage versus different group sizes, respectively. In general they increase with the group size. Since GNP is correlated well with the relative locations of hosts in the Internet, its DT mesh constructed consists of edges with short delay, and hence is better than the one based on geographic locations. As the group size increases, the savings are more remarkable. For a medium group ($\approx$ 128 hosts), RDP and network resource usage are already substantially reduced (both by more than 50%). GNP also reduces the average physical link stress (in Figure 4(c)). It is because GNP reduces the lengths of

[1] Equations (1) and (2) are the two-dimensional case of the simple error measurements originally mentioned in [6].

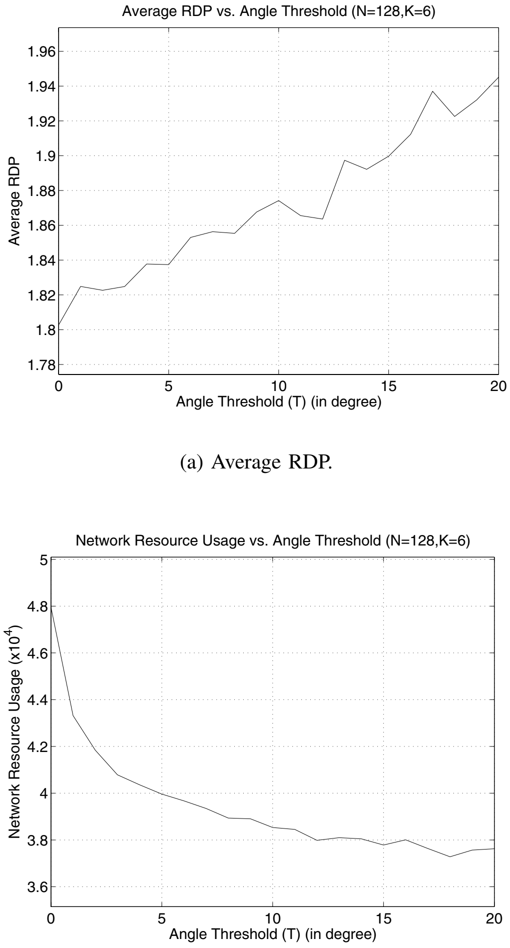


(a) Average RDP.

(b) Network Resource Usage.

(c) Average Physical Link Stress.

Fig. 4. Performance comparison of DT mesh formation based on GNP and geographic location.



(a) Average RDP.

(b) Network Resource Usage.

Fig. 5. Network resource usage and delay versus T.

overlay edges, less hop count is taken for a packet passing through an overlay edges, and hence fewer packets are generated in the network.

We then discuss the proper angle threshold T. Recall that T trades off end-to-end delay with network resource consumption. We take the coordinates obtained by GNP to construct a DT mesh and vary the threshold T (in Figure 5). Though the overall network resource usage is very high for $T = 0^o$ (i.e., if no delegation is done), it sharply decreases to a rather stable value for $T \approx 10^o$. From Figure 5(a), we see that such a decrease comes with only a mere cost of higher delay (due to more hops to destinations).

We next examine the effect of the maximum fanout of hosts (in Figure 6) by varying the value of K. The RDP first decreases rather sharply and settles to a low value for a certain K. Therefore, it is recommended a host to serve a few (about $5 - 7$) hosts in order to achieve a reasonably low RDP.

Finally, we show the cumulative distribution of RDP and physical link stresses in Figures 7 and 8, respectively, when our baseline parameters are used. As can be seen, our schemes effectively achieve load-balancing. The proportion of links experiencing high stresses is very low (less than 5% have stress higher than 4). Moreover, the RDP of a large portion of users is very low. Over 90% users have RDP less than 3, and over 70% hosts have RDP less than 2.

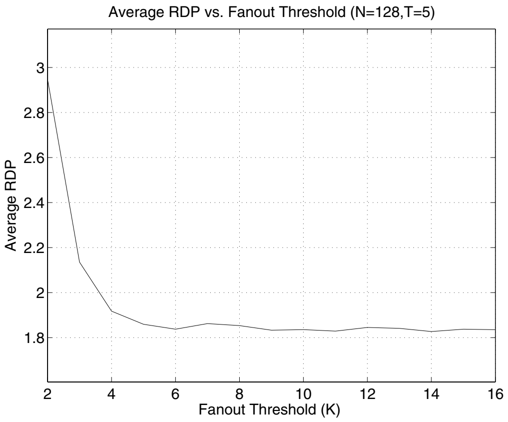


Fig. 6. RDP versus K for embedded tree and bypass tree.

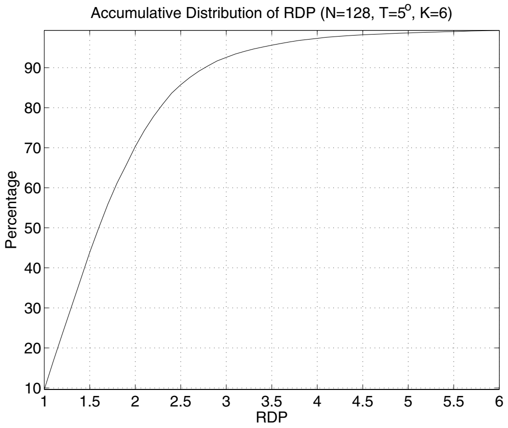


Fig. 7. Cumulative distribution of RDP.

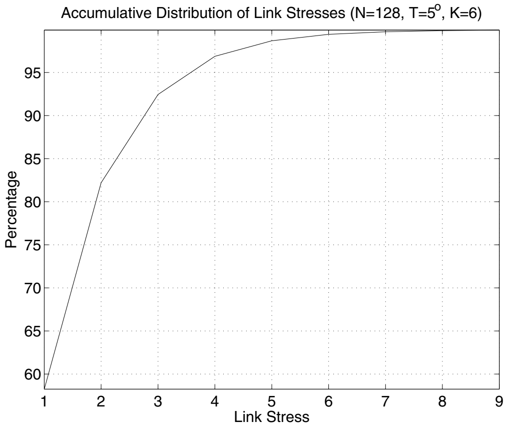


Fig. 8. Distribution of physical link stresses.

VI. Conclusion

In this paper, we address three weaknesses of the traditional Delaunay Triangulation Protocol, namely 1) inaccurate estimation of host location; 2) high network resource consumption; and 3) unlimited fanout of a host. To address the first weakness, we use Global Network Positioning to estimate the relative locations of hosts in the Internet based on measured network delays. Since the distances in GNP space is highly correlated with the network delay, this improves DT mesh substantially in terms of RDP, resource usage and physical link stress. To address the second and third weaknesses, we use *forwarding delegation* algorithm to limit the fanout of a host according to its capability. The algorithm also agrregates long-delay (inter-domain) paths to reduce network resource usage if the angle between these two paths is small.

Using Internet-like topology we show that our scheme, as compared to the original DT protocol, can substantially improve average relative delay penalty and network resource usage (by over 50%). Furthermore, we show that if the angle between two adjacent paths is low (10^o), we should aggregate them to further reduce the network resource usage. In DT, a host should be able to serve around $5-7$ neighbors in order to sufficiently reduce the RDP. Our scheme is able to achieve low RDP, network resource usage, physical link stress for a group even with more than a thousand hosts.

References

[1] SE. Deering and DR. Cheriton, "Multicast routing in datagram internetworks and extended lans.," *ACM Transactions on Computer Systems*, , no. 2, pp. 85–110, May 1990.

[2] Christophe Diot, Brian Neil Levine, Bryan Lyles, Hassan Kassem, and Doug Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, , no. 1, pp. 78–88, January/Februray 2000.

[3] Jörg Liebeherr, Michael Nahas, and Weisheng Si, "Application-layer multicasting with delaunay triangulation overlays," *IEEE Journal on Selected Areas in Communicastions*, vol. 20, no. 8, pp. 1472–1488, October 2002.

[4] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG9́9)*, Vancouver, August 1999, pp. 51–54.

[5] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Cheong, *Computational Geometry, Algorithms and Applications*, Sprnger-Verlag, 2 edition, 1997.

[6] T. S. Eugene Ng and Hui Zhang, "Predicting internet network distance with coordinates-based approaches," in *Proceeding of INFOCOM 2002*, New York, June 2002.

[7] Beichhuan Zhang, Sugih Jamin, and Lixia Zhang, "Host multicast: A framework for delivering multicast to end users," in *Proceedings of INFOCOM 2002*, 2002, pp. 1336–1375.

[8] Yang hua Chu and Sanjay G. Rao Srinivasan Seshanand Hui Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communicastions*, vol. 20, no. 8, pp. 1456–1471, October 2002.

[9] Miguel Castro, Peter Druschel, Anne-Marie Kermarec, and Antony I. T. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communicastions*, vol. 20, no. 8, pp. 1489–1499, October 2002.

[10] Sylvia Ratasamy, *A Scalable Content-Addressable Network*, Ph.D. thesis, University of California at Berkeley, Fall 2002.

[11] Sibson R., "Locally equiangular triangulations," *The Computer Journal*, vol. 3, no. 21, pp. 243–245, 1977.

[12] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of INFOCOM 1996*, San Francisco, CA, 1996.