Network-aware SuperPeers-Peers Geometric Overlay Network

Eng Keong Lua
NTT Service Integration Laboratories
NTT Corporation
Email: lua.engkeong@lab.ntt.co.jp

Xiaoming Zhou
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Email: x.zhou@ewi.tudelft.nl

***Abstract*— Peer-to-Peer (P2P) overlay networks can be utilized to deploy massive Internet overlay services such as multicast, content distribution, file sharing, etc. efficiently without any underlying network support. The crucial step to meet this objective is to design network-aware overlay network topologies connecting all nodes that offer promising properties in terms of excellent communication quality. We exploit the underlying network locality and proximity of the nodes for overlay routing and node placement strategy. In this paper, we describe in greater specific details our network-aware SuperPeers-Peers geometric overlay network hierarchy and study its communication quality in a massive scale network environment. We evaluate our proposal using ten massive scale networks each consisting of $100,000$ nodes. Our experimental results show high communication efficiency, quality and performance.**

I. Introduction

A Peer-to-Peer (P2P) overlay network [9] is an effective method to support new applications and protocols that require no modifications to the underlying network layer. A P2P overlay network is formed by connections between the overlay nodes; each overlay connection may comprise of one or more physical links, with an IP-layer path connecting each pair of the overlay nodes. Since P2P overlay networks are built at the application layer, it is very effective to provide higher level services to the users by taking the Internet as lower level infrastructures. It is important to have some levels of underlying network *awareness* to mitigate multiple overlay edges from traversing the same underlying network links and multiple communications across many end systems that will produce redundant traffic and increase latency. To design *network-aware* overlay network, we could initiate continuous network measurements to determine underlying network metrics such as latencies. However, such a method will result in a large measurement overhead when overlay usage and node churn are high. If we perform network measurements intermittently, the resulting measurement may not be related to the practical usage of the overlay and thus leads to stale information.

In our approach, we exploit accurate and scalable Internet subspace embedding (*Highways* [8]) of latencies such as Round-Trip-Times (RTTs) between nodes into a low-dimensional geometric space by measuring latencies between *some* nodes and assign geometric coordinates to *all* nodes in such a way that the geometric distance between node coordinates closely approximates their RTTs. The measurement overhead is reduced because non measurements are estimated. The geometric space can be maintained in a distributed manner with a small number of network latency measurements. The network embedding system adapts to dynamic network changes as the overlay nodes update their node coordinates iteratively. A network-aware SuperPeers-Peers geometric overlay hierarchy is then created to scale the overlay network communication and management — Lightweight SuperPeers Topologies (LST). The SuperPeers layer provides a backbone infrastructure for communications among all nodes in the network. The Peers in the Peers layer are connected to their closest SuperPeers in terms of their shortest geometric distances between them computed from their node coordinates. We use *Yao-graph* [14] to build the connectivity at SuperPeers layer — every SuperPeer is connected to *six* closest SuperPeers (neighbors).

In our previous work [3], [4], we gave a description our design principles and presented our evaluation results in the planetary-scale environment (PlanetLab) consisting of different groups of SuperPeers. In this paper we describe in specific details our Internet subspace geometry, construction algorithm of the SuperPeers-Peers hierarchy, geometric overlay routing mechanism, and overlay maintenance operations. This time, we evaluate our LST overlay network using *ten* massive scale networks each consisting of $100,000$ nodes. This massive scale network environment was used by Scribe [1] at the Microsoft Research Cambridge. Our performance evaluation results in the massive scale networks show that our LST overlay network has high communication efficiency, quality and performance. The rest of the paper is organized as follows. Section II describes the design principles of our network-aware geometric overlay network. We give details on the accurate and scalable Internet geometry that supports our overlay network construction, maintenance management and robustness during node churn, and geometric overlay routing. Section III explains the setup of the simulation experiments using ten massive scale networks, each consisting of $100,000$ nodes. Section IV discusses our performance evaluation results in the massive scale networks. Section V concludes.

II. Network-aware Geometric Overlay Network

The LST overlay network is divided into two layers: SuperPeers and Peers. The upper SuperPeers layer consists of

1-4244-1251-X/07/$25.00 2007 IEEE.

elected Superpeers based on the selection criteria of sufficient resources and reliability [3], [4], and acts as a reliable high-bandwidth backbone network infrastructure for communications among all nodes. Every Peer in the Peers layer is connected to their closest SuperPeer in terms of the shortest geometric distance between them for end-to-end overlay geometric routing. Figure 1 shows the topology structure of our network-aware SuperPeers-Peers geometric overlay hierarchy.

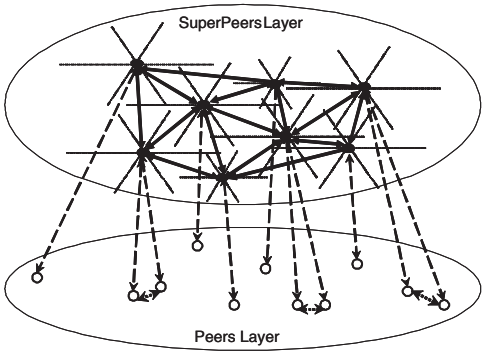


Fig. 1. Network-aware SuperPeers-Peers geometric overlay hierarchy.

A. Internet Subspace Geometry

Network embedding computes node coordinates and geometric distances between nodes to estimate their underlying network metrics such as latency in a scalable way. That is, nodes are mapped onto points in a geometric space and they are assigned geometric coordinates in such a way that the computed geometric distances between node coordinates closely approximates the latencies (RTTs) between nodes. These node coordinates also reflect their geometric position in the geometric space. RTT measurements from each node to some landmarks are performed for embedding into a geometric space. So, it does not require full mesh N^2 network measurements of N nodes that will cause extensive overheads to deduce the quality of the underlying network metric (i.e. RTTs) between nodes. Using these geometric distances, efficient and selective placement of nodes in the geometric overlay network can be done. In other words, we are able to determine node locality from the node coordinates and their geometric distances between nodes. Basically, the LST overlay network learns of the underlying RTTs between nodes through their coordinates and computed geometric distances. Our Internet geometry simplifies the construction of network-aware geometric overlays.

From the scalability meta-metric observations in [10], subspace embedding in Euclidean space of RTTs between nodes in smaller partitioned clusters achieves better embedding accuracy. The nodes in the clusters are closer to each other and they have closer landmarks. We exploit this idea in *Highways* [8] and develop it as an overlay network control plane service providing geometric location information for the LST overlay network. In *Highways*, superspace embedding [10] embeds the whole set of overlay nodes in the system as one large set into *global* geometric space while subspace embedding embeds small partitioned clusters of overlay nodes into *local* geometric space. In this manner, both the global and local geometric spaces are established to derive the global and local geometric positions of all overlay nodes respectively. The local geometric position information helps to provide an accurate geometric distance estimation between overlay nodes within each of the clusters while the global geometric position information estimates the inter-cluster geometric distances between overlay nodes in different clusters. We use 2-dimensional Euclidean space as the embedded geometric space in our Internet geometry.

In our system, we first measure the RTTs among the SuperPeers and use this measured RTT matrix to partition the SuperPeers into smaller clusters by adopting a simple approach of the K-means method [11]. The algorithm separates and combines nodes into clusters in the LST overlay network by assigning each node to the cluster having the nearest centroid (mean) based on the geometric distances. We use $K = 3$ as the number of partitions due to the geographical continents of the world, which comprises generally of North/South America, Africa/Europe and Asia Pacific. After all the SuperPeers have been partitioned into smaller clusters, all Peers measured their RTTs to all the SuperPeers and every Peer uses this information to find the closest SuperPeer. The Peer then joins the cluster whereby the closest SuperPeer belongs. These measured RTTs between SuperPeer-to-SuperPeer and Peer-to-SuperPeer as a result of clustering are usable and required for the network embedding. This is because the SuperPeers are being used as the landmarks for the network embedding technique. Thus there is *no* additional overhead and redundancy in measuring these raw RTT measurements and they are less than N^2 measurements taken.

Subspace embedding in Euclidean space is performed strictly in each of the partitioned clusters to compute node local geometric coordinates. To compute the inter-cluster geometric distances between all nodes residing in different clusters, we make use of the basis transition matrix. We would be able to transform the node local geometric coordinates from its local geometric space to the node global geometric coordinates in the global geometric space. Once the transformation is done, we are able to compute the inter-cluster geometric distances between these nodes residing in different clusters that spans the global geometric space, without the need to measure any property between itself and the landmarks that spans such a space. Here we describe the landmark-based embedding and singular value decomposition (SVD) technique that map the nodes into points in a low-dimensional geometric space. In our LST overlay network, the list of elected SuperPeers are the landmarks for the network embedding.

To calculate and assign coordinates of k-dimensional geometric space for all N nodes in X, *at least* $k + 1$ landmarks (SuperPeers) are selected. This is to solve the possible problem that coordinate vectors of the landmarks could be linearly dependent in the geometric space which may cause the nodes to unable to differentiate their distinct geometric

locations from these landmarks and hinder the computation of the node coordinates. That is, if the landmarks have their coordinate vectors as a multiple of the other i.e. the landmarks are in a straight vector line, then the nodes would not be able to compute their distinct geometric locations from these landmarks. As in [12], this framework relies on a set of landmarks from which the nodes may select any set consisting of at least $k+1$ landmarks out of a list of all landmarks for embedding into k-dimensional geometric space. This allows flexibility for a node to determine its geometric coordinates in choosing its set of landmarks without the need to use a fixed infrastructure of well-known landmarks. It solves the problems of communication bottlenecks and single points-of-failure caused by the use of well-known landmarks. However, note that the chosen set of landmarks must share at least one common landmark in their selection so that the vector basis constructed spans the embedded geometric space.

A symmetric measured network latency matrix D for the set of landmarks $L = \{l_1, l_2, \cdots, l_m\}$ is derived as $D = [d(i,j)]_{i,j=1,\cdots,m}$, where $m \geq k+1$ and $d(i,j)$ is the measured network latency (RTTs) between m landmarks. For $i \neq j$, $d(i,j) = d(j,i)$ and $d(i,i) = 0$. Dimensionality reduction to k is done using SVD:

$$D = U \cdot W \cdot V^T \tag{1}$$

where U and V^T are orthogonal matrices, and W is a diagonal matrix containing the singular values of D.

The RTT measurements of all overlay nodes $i \in X$ where $i = 1, \cdots, N$ to their sets of selected landmarks $L = \{l_1, l_2, \cdots, l_m\}$ are made. This can be written using column vector notation for a node i as below:

$$\phi(i) := \begin{pmatrix} d(i,l_1) \\ d(i,l_2) \\ \vdots \\ d(i,l_m) \end{pmatrix} \tag{2}$$

This is simply the Lipschitz embedding of X using set of landmarks L.

By using the first k columns of U denoted by Uk, we project the m-dimensional space into a new k-dimensional space: where $\phi^{'}(i) = Uk^T.\phi(i)$ is the coordinates of node i after dimensionality reduction.

To minimize the discrepancy between the distance represented in the coordinates system and the measured distance between m landmarks, we defined and used a scaling factor: $\alpha_k = \frac{\sum_i^m \sum_j^m d(i,j).\delta_(i,j)}{\sum_i^m \sum_j^m \delta(i,j)^2}$ where $\delta(i,j) = L_2(Uk^T.\phi(i), Uk^T.\phi(j))$, where L_2 is the Euclidean norm since we use Euclidean space.

In order for a node to know about the global geometric space G and derive its coordinates in G, without measuring any property between itself and the nodes that form such a vector basis, a basis changing technique is adopted and a basis transition matrix T_G is maintained. That is, a basis transition matrix T_G is computed for converting the node local geometric coordinates between its local basis of the local geometric space C to global basis of the global geometric space G to derive its global geometric coordinates. The basic insight is that a randomly selected set of landmarks defines an embedding geometric space that can be easily (linearly) mapped into another embedding space derived from a different set of landmarks. We maintain a basis transformation matrix for the ease of converting node local geometric coordinates from its local geometric space to the global geometric space to derive its global geometric space, without measuring any property between itself and the landmarks that spans such a space.

If we want to change the local basis of $\mathbb{R}^k$ from local geometric space C to the global geometric space G, the basis transition matrix T_G is calculated by a selected arbitrary set of nodes. This selected set of nodes measure coordinates to two landmark sets in the global geometric space G and local geometric space C. Then, the following equation is solved using least squares to obtain T_G:

$$\begin{aligned} T_G.P_C &= P_G \\ T_G &= P_G.P_{C^{-1}} \end{aligned} \tag{3}$$

where P_G is the selected set of node global geometric coordinates in global geometric space G and P_C is the selected set of node local geometric coordinates in original local geometric space C.

Once we have T_G, then we can calculate the global geometric coordinates of the node i in the global geometric space G from its local geometric coordinates in original local geometric space C:

$$\phi_G(i) = T_G.Uk_C^T.\phi_C(i) \tag{4}$$

Therefore, the node global geometric coordinates in the global geometric space G can be obtained relative to the basis transition matrix T_G and its node local geometric coordinates in the original local geometric space C, with nothing more than the information it already has.

We expect nodes to recompute their node coordinates iteratively due to node churn or topology changes. Such changes are captured by the RTTs between nodes. In this case, a node recomputes its coordinates following the above embedding steps. If for some reasons, a landmark becomes unavailable during this recomputation process, the node then chooses other alternative landmark to devise the basis transition matrix.

B. SuperPeers-Peers Geometric Overlay Hierarchy

The basic idea is that we construct *Yao-graph* [14] at the SuperPeers layer by cutting the 2-dimensional Euclidean space around each SuperPeer into *six* sectors, each with equal geometric angle of $\theta = \frac{\pi}{3}$. Every SuperPeer in the *Yao-graph* chooses the *closest* SuperPeer (neighbor) in terms of the shortest geometric distance to other SuperPeer in each of the *six* sectors. So, every SuperPeer is connected to *six* closest SuperPeer neighbors. The *Yao-graph* is proven to exhibit the Euclidean minimum spanning tree (EMST) in [14]. Previous

works in [2], [7] use *Yao-graph* for the design of mobile wireless networks. Their good communication performance results motivate us to use the *Yao-graph* in our overlay geometric structure. Such graph structure is able to minimize overhead during overlay maintenance management. *Yao-graph* can be maintained locally in a distributed manner because each node is connected to other *six* closest neighbors based on the shortest geometric distances between nodes. That is, the local maintenance algorithm is confined to the affected node and its immediate *six* closest neighbors. In this manner, this geometric graph structure allows efficient and lightweight local recovery from node churn. In addition, *Yao-graph* was the first technique to break the $O(N^2)$ time complexity barrier in the computation of the EMST in a connected graph with N nodes [14].

Here we describe our self-stabilizing and distributed *Yao-graph* construction protocol as shown in Figure 2. We consider a connected graph $G(V, E)$, where V corresponds to a set of points (nodes) in the Euclidean space $\mathbb{R}^2$, and E to the set of edges with weight corresponding to the Euclidean length of an edge. Suppose that every node $u \in V$ knows its neighborhood $N(u)$ and the current positions of the nodes in $N(u)$ in the Euclidean space. Every node aims at maintaining a connection to the closest node in every sector S (or cone). Let $E(u)$ be the current set of the connections of node u.

YAO-GRAPH(V, E)
1 **while** Node $u \in V$
2 **do for** Every Node $w \in E(u)$
3 **do if** Node $v \in N(u)$ in $w's$ sector with $\|uv\| < \|uw\|$
4 **then** Remove w from $E(u)$
5 **for** Every Sector S of u
6 **do if** S has at least one node in $N(u)$ but no node in $E(u)$
7 **then** Add the node w in S of shortest distance to u to $E(u)$

Fig. 2. A distributed and self-stabilizing *Yao-graph* topology construction protocol.

***Theorem** 1:* When the distributed *Yao-graph* topology construction protocol self-stabilizes in the stable state, the out-degree of every node is at most s where the 2-dimensional Euclidean space around every node $v \in V$ is cut into s sectors with angle $\theta = \frac{2\pi}{s}$.

Proof: Follows directly from the distributed and self-stabilizing protocol. In our 2-dimensional Euclidean space, the out-degree of a node is 6 since every node connects to 6 closest neighbors in the 6 sectors with angle $\theta = \frac{\pi}{3}$. ■

We use distributed *Yao-graph* topology construction protocol to build the overlay network connectivity among the SuperPeers based on their geometric coordinates and distances with other SuperPeers. These **SuperPeer-SuperPeer Yao-graph routes** serve as the reliable high-bandwidth backbone network connectivity for the overlay network. In the Peers layer, Peers are directly connected to the *closest* SuperPeers that are capable of serving an additional Peer and this connectivity is called the **Peer-SuperPeer 1-Hop route**. Among the Peers being served by their closest SuperPeer, direct connectivity between these Peers can be established if there exists a shortcut route between the Peers. That is, a **Peer-Peer Shortcut route** is established between two Peers belonging to a SuperPeer, if the direct connectivity between these two Peers is shorter than their Peer-SuperPeer 1-Hop routes. This architecture is illustrated in flat geometric view as shown in Figure 3. The various overlay routes in the 2-tier LST overlay network are illustrated in Figure 4.

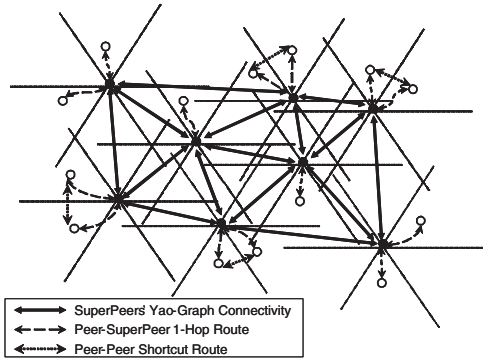


Fig. 3. Architecture of the 2-tier LST overlay network in flat geometric view.

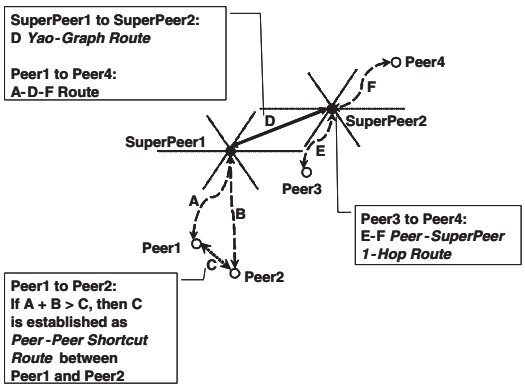


Fig. 4. Various overlay routes in the 2-tier LST overlay network.

C. Geometric Overlay Network Routing

We use the *localized* geometric routing algorithm — random compass routing [5] — to route data from one SuperPeer to destination SuperPeer in the *Yao-graph* at the SuperPeers layer. For end-to-end routing, the Peer-Peer Shortcut route and Peer-SuperPeer 1-Hop route are utilized to complete the routing process. That is, for end-to-end routing between Peers, if there exist a Peer-Peer Shortcut route, then packet is delivered using this route. Otherwise, the Peer-SuperPeer 1-Hop route is used to route the packet from the source Peer to the source SuperPeer at the SuperPeers layer serving the source Peer and localized random compass routing protocol is activated to deliver the packet to the destination SuperPeer that serve the destination Peer which the Peer-SuperPeer 1-Hop route is used to complete the routing process. The motivation for using localized random compass routing algorithm at SuperPeer layer is that the results in [6] show that the delivery rate of random compass routing in *Yao-graph* is 100%.

D. Geometric Overlay Network Maintenance

Our 2-tier LST overlay network uses stable overlay nodes as SuperPeers to handle most of the heavy system workloads and reduce the network maintenance overhead. When an overlay node leaves the system or a failure occurs, the information that is related to the leaving overlay node must be updated among the other affected overlay nodes. Similarly, if a new overlay node joins the system, information relating to the new overlay node will also have to be updated. For high node churn, network maintenance overhead can be heavy. The following cases describe the *LST* overlay network maintenance operations during node churn which will invoke the *local* topology repair algorithm.

New Overlay Nodes Joining.

A new overlay node will contact the *bootstrap service* operating at the SuperPeers layer for a standard overlay **JOIN** procedure. Once the new overlay node is elected as a SuperPeer or normal Peer, the following operations are executed.

A new normal Peer is joining the LST overlay network. During the **JOIN** procedure, the new normal Peer measured the RTTs to all existing SuperPeers and use this information to join the cluster whereby the closest SuperPeer belongs. Then, the new normal Peer's geometric coordinates within the cluster are computed by the *Highways* overlay control plane service. Using the estimated geometric distances, the Peer-SuperPeer 1-Hop route to the closest SuperPeer is established and all possible Peer-Peer Shortcut routes are setup and updated with the neighboring Peers within the cluster.

A new SuperPeer is joining the LST overlay network. A new overlay node is elected as the new SuperPeer L and measured the RTTs to all existing SuperPeers and use this information to join the cluster whereby the closest SuperPeers belongs. Then, the new SuperPeer's geometric coordinates within the cluster is computed by the *Highways* overlay control plane service.

This new SuperPeer L starts to cut the space surrounding itself into *six* sectors with equal angle of $\theta = \frac{\pi}{3}$. Then this new SuperPeer builds the *Yao-graph* overlay connectivity by connecting to other *six* closest SuperPeers in terms of the shortest geometric distance to other SuperPeers in its *six* sectors. It attempts to connect to the list of *six closest* neighboring SuperPeers in each of its *six* sectors. That is, the local topology repair algorithm will be invoked to reconstruct the *Yao-graph* topology at the SuperPeers layer to include this new SuperPeer. The existing Peers associated with the neighboring SuperPeers also reorganize and reestablish the Peer-SuperPeer 1-Hop routes to this new SuperPeer, if there exists shortest 1-Hop routes.

The new SuperPeer L initializes its state by routing **Join** messages to the list of closest neighboring SuperPeers found in each of its *six* sectors. Once **Join** messages are routed to this list of *six* closest neighboring SuperPeers, the new SuperPeer L will establish *Yao-graph* overlay connectivity in the *six* sectors. The new SuperPeer L learns of the IP addresses of these closest neighboring SuperPeers in the *six* sectors. The neighboring SuperPeers also require to update their neighbor tables to eliminate those SuperPeers that are no longer neighbors as a result of this new inclusion of SuperPeer L. In addition, the existing Peers associated with the *six* closest neighboring SuperPeers reorganize and reestablish the Peer-SuperPeer 1-Hop routes to this new SuperPeer if there exists shortest Peer-SuperPeer 1-Hop routes. The connection relationships of the affected peers that change their Peer-SuperPeer 1-Hop routes to this new SuperPeer L will be updated.

This update is done by broadcasting an **Update** message containing the new topological information to all affected SuperPeers and Peers, as illustrated in Figure 5. Once the *Yao-graph* overlay connectivity of this new SuperPeer L has been established, it will have a set of maximum *six* neighboring SuperPeers $N = \{N_1, N_2, \cdots, N_6\}$ in the SuperPeers layer. In this example, before SuperPeer L joins the SuperPeers layer, SuperPeers N_1, N_2 and N_3 are connected directly to each other with *Yao-graph* overlay routes, as shown in the dashed lines. After the SuperPeer L joins, the new SuperPeer L will connect to the neighboring SuperPeers N_1, N_2 and N_3 with their *Yao-graph* overlay routes, as shown in the dotted lines. The SuperPeers N_1, N_2 and N_3 have to adjust their *Yao-graph* overlay connectivity by updating their neighbor tables. As a result of this update, the direct *Yao-graph* overlay route between SuperPeers N_1 and N_2 does not exist and new *Yao-graph* overlay routes are established to the new SuperPeer L. The **Update** message is sent with a limited range of time-to-live (TTL). The expected number of hops in the TTLs, E[$Hops_N$] is $\log_6 N$ (where N is the number of SuperPeers in the SuperPeer layer, and a SuperPeer has *six* neighboring SuperPeers because of *six* sectors division in the *Yao-graph*). In our case where $N = 10,000$, this gives about 5 TTLs. This **Update** procedure ensures that the affected SuperPeers will quickly learn about the change and perform necessary update on their own neighbor tables accordingly.

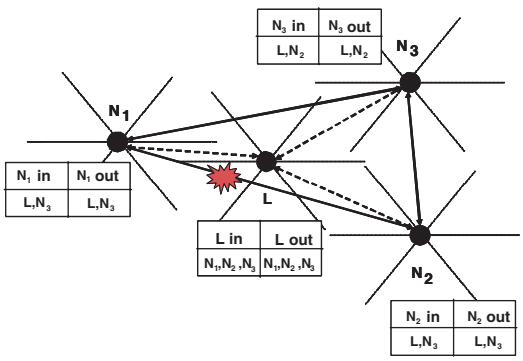


Fig. 5. Updates of neighbor tables in the *Yao-graph* topology at SuperPeers layer when a new SuperPeer joins.

Existing Overlay Nodes Leaving.

To be able to detect overlay nodes leaving the *LST* overlay network or overlay node failures, a *heartbeat* approach is used. Every overlay node sends small *alive* messages to each other

periodically and maintenance operations are invoked when heartbeats are lost. The following operations are executed:

A normal Peer is leaving the LST overlay network. The missing heartbeat will be detected from this normal Peer. The associated SuperPeer and peers who have their Peer-SuperPeer 1-Hop and Peer-Peer Shortcut routes with this normal Peer will attempt to free their connection resources. Only the affected SuperPeer and peers are reorganized locally and this minimizes global overhead management.

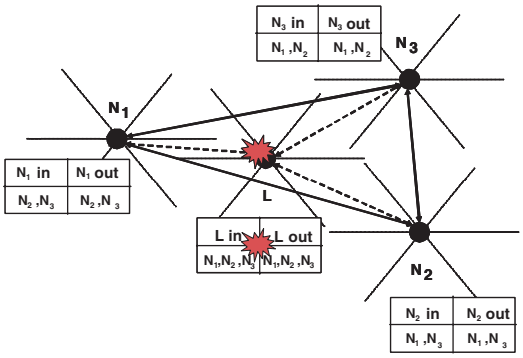


Fig. 6. Updates of neighbor tables in the *Yao-graph* topology at SuperPeers layer when a SuperPeer leaves.

A SuperPeer is leaving the LST overlay network. The *six* neighboring SuperPeers connected to this leaving SuperPeer will notice this failure through the missing heartbeats and trigger the local topology repair algorithm. It will reconstruct the *Yao-graph* relationships of the *six* neighboring SuperPeers and reorganize the Peer-SuperPeer 1-Hop and Peer-Peer Shortcut routes of its connecting Peers.

The neighboring SuperPeers of this leaving SuperPeer are notified of this change and update their neighbor tables. This notification occurred because SuperPeers periodically exchange *alive* heartbeat messages. When this leaving SuperPeer leaves the overlay network and heartbeats stop, every neighboring SuperPeers will send a **Discovery** broadcast message with a limited 5 TTLs to other neighboring SuperPeers. Each neighboring SuperPeer receiving the **Discovery** broadcast message will respond with its geometric position information and its IP addresses. The **Join** and **Update** procedures (described above) will help to adjust its current topology state for the affected SuperPeers and Peers as illustrated in Figure 6. This ensures that the SuperPeers' *Yao-graph* links and Peers' Peer-SuperPeer 1-Hop routes can be quickly reconstructed locally as a result of this change. Due to the lightweight properties of *Yao-graph*, node churn causes little problem to the hierarchical layers of the LST overlay network, as long as a SuperPeer does not become disconnected by the loss of all its neighboring SuperPeers. Even in the extreme case of losing all neighboring SuperPeers, the affected SuperPeer can contact the bootstrap service to rejoin the network.

III. Implementation and Experimental Setup

Our massive scale simulation experiments are implemented using the massive scale networks that were used by Scribe [1] at the Microsoft Research Cambridge. The massive scale networks are generated by Georgia Tech random graph generator [15]. The hierarchical transit-stub model contains 5050 routers. There are 10 transit domains at the top level with an average of 5 routers in each. Each transit router has an average of 10 stub domains attached, and each stub has an average of 10 routers. There are $100,000$ end-system nodes that were randomly assigned to routers in the core with uniform probability. Each end-system node is directly attached by a local area network (LAN) link to its assigned router.

There are ten different networks using the same parameters but different random seeds — we have 10 massive scale networks with network model named as 0 to 9. We use the policy routing link weights generated by Georgia Tech random graph generator to perform IP unicast routing. That is, all links of the same type such as intra-domain or inter-domain links, are assigned the same link weight. IP multicast routing uses a shortest path tree formed by merging the unicast routes from the source to each receivers. For such a massive scale network, it is more feasible to develop a simulator for our experiments. The well-known network simulator such as *ns*-2, would *not* be able to handle this large size of the networks involved and the dynamics of the overlay networks. The simulator models the propagation delay on the physical links as follows. The delay of each LAN link was set to 1 ms and the average delay of core links was 40.7 ms. Our simulator does not model queuing delay, packet losses, or any cross network traffic because modeling of such parameters would prevent the simulation of massive scale networks. To examine whether the LST overlay network is efficient in supporting multiple concurrent applications with varying requirements, we run experiments using a large number of groups with a wide range of group sizes. As in Scribe [1], since there are no sources of real-world trace data to drive the experiments, a Zipf-like distribution for the group sizes is adopted. The size of a group with rank r is given by $gsize(r) = \lfloor Nr^{-1.25} + 0.5 \rfloor$, where N is the total number of overlay nodes. In each network model, the total number of group ranks was fixed at 150 (i.e. the total number of groups is 150 with group rank 1 to 150) and the number of overlay nodes (N) was fixed at $100,000$, which were the numbers being simulated.

In each group, we choose 10% of the total number of overlay nodes to be the SuperPeers based on the election criteria. The reason for the choice was derived from the recent study [13] which states that there are approximately 10% of the overlay nodes that have high capacity, and they exhibit stability and reliable connectivity in the overlay network. In each network model, the maximum number of SuperPeers is $10,000$ in group rank 1 which consists of $100,000$ nodes. We run our simulation system on these 10 massive scale networks and the total number of groups is 1500. We generate 2-dimensional Euclidean coordinates for all the nodes in the system. Since our performance results in all 10 networks are similar, only the *average values* over the 10 massive scale networks are shown. Figure 7 visualizes an example of the *Yao-graph* geometric topology structure at the SuperPeers layer in 2-dimensional

Euclidean space for group rank 35 containing a total of 117 SuperPeers.

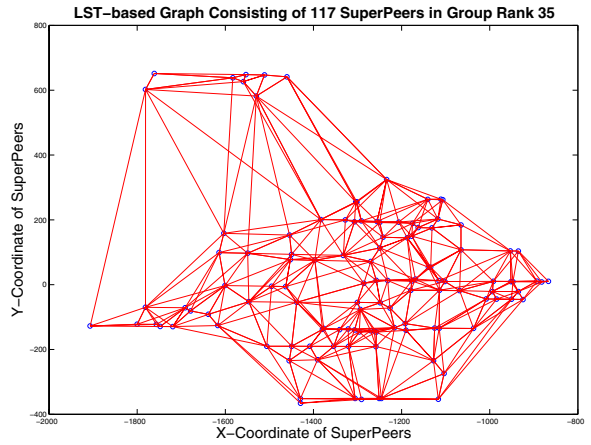


Fig. 7. *Yao-graph* geometric structure of 117 SuperPeers in group rank of 35.

IV. Evaluation of Performance Results

Overlay Cost vs Underlay Cost.

For each of the source-destination SuperPeers pair, we compare the LST overlay network cost and underlay network cost in terms of latencies on a path-by-path basis. That is, we measure and compare the network cost between two nodes for direct IP communication utilizing the underlying network and the cost of using the overlay network. Figure 8 shows the scatter-plot performance of geometric SuperPeers-to-Peers overlay hierarchy for group rank 72, 96 and 148 respectively. The X-axis is the LST overlay network cost and the Y-axis is the underlay network cost. The solid linear line gives the indication of network cost being equal to the overlay cost and its purpose is to show this boundary. The results show that for some cases, using the LST overlay network for communications outperforms the direct IP-based communications in the underlying network. It also shows that the LST overlay network communications' latencies are *reasonable* in delivering messages relative to their direct underlay communications. These results are expected because all overlay communications usually suffer a slightly higher communication cost than the direct Internet communications. This is due to the overlay network routing that is usually not as optimal as direct communications in the underlying network.

In-degree/Out-degree of a SuperPeer.

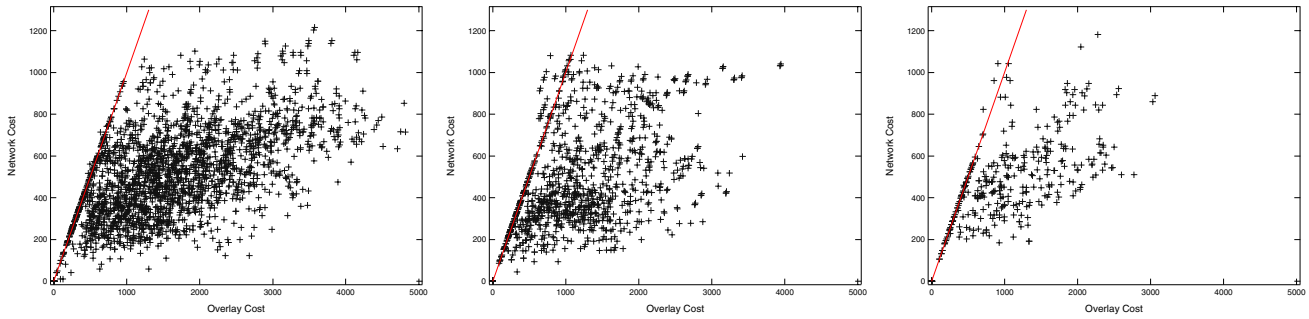
The in-degree/out-degree of a SuperPeer denotes the number of connected incoming and outgoing neighbors that are maintained by that SuperPeer. For a SuperPeer to limit its outbound network bandwidth, it must limit its out-degree in the overlay network, otherwise, its forwarding capacity can be exceeded. A *Yao-graph*'s node has its out-degree being bounded by s number of sectors and its in-degree can be as high as $N-1$ for a total of N nodes. In our LST overlay network, the out-degree of the SuperPeer is bounded at most 6, which is reasonably small. A SuperPeer with a high in-degree may easily become exhausted. It is interesting to find out the in-degree of the SuperPeer in the *Yao-graph* topology using the massive scale networks.

For each network model and group, we compute the in-degree/out-degree of each SuperPeer. As expected, the maximum out-degree of a SuperPeer in our *Yao-graph* is 6. This is due to the bounding characteristics of our *Yao-graph* geometric structure in 2-dimensional Euclidean space which has *six* sectors connecting to *six* outgoing neighbors. Our experimental results indicate that the average in-degree is the *same* as (equal to) that of average out-degree. The distributions of mean node degree and maximum in-degree for different group ranks are shown in Figure 9. The X-axis (log scale) is the group ranks in descending group size and the Y-axis (log scale) is the node degree. The figure shows that the average in-degree (node degree) of each group is relatively small, with a maximal average of 6. A small average in-degree suggests a low *link stress* for overlay communications in the massive scale networks. The maximal in-degree of group rank 1 (largest group size) is 150, which is still realistic.

Figure 10 illustrates the standard deviation of in-degree and out-degree for different group ranks. The axes are the same as the previous figure. The figure shows that the standard deviation of out-degree in SuperPeer decreases as group rank decreases for 150 to 1. This means that the standard deviation of out-degree decreases as the group size increases. The standard deviation of out-degree is small: the minimum standard deviation of out-degree is 0.34 in group rank 2 and maximum standard deviation is 1.22 in group rank 130, giving an average of 0.98 over all groups. Again, this result is expected for all groups — all SuperPeers in our *Yao-graph* have their out-degree bounded at most 6. However, the standard deviation of in-degree in SuperPeer increases with the group size. The minimum standard deviation of in-degree is 1.37 in group rank 138 and maximum standard deviation of in-degree is 5.3 in group rank 1, giving an average of 2.03 over all groups. This shows that the in-degree of a SuperPeer can be relatively high. This may be due to the possibility that there exists such a special SuperPeer that is the only nearest neighbor to many other SuperPeers. To overcome the possibility of exhausting the in-degree of a SuperPeer, undirected *sparsified Yao-graph* [14] can be considered. Basically, a *sparsified Yao-graph* is a *Yao-graph* whereby only the shortest incoming edge is selected for incoming link if the in-degree of a sector exceeds one.

V. Conclusion

Since we cannot ignore the underlying network metrics such as latencies (RTTs) between nodes to construct efficient P2P overlay network, we propose and design a P2P *network-aware geometric* overlay network. To evaluate our proposal in a massive scale network environment, we carry out simulation experiments on *ten* massive scale networks each consisting of $100,000$ nodes. Our experimental evaluation results show high communication efficiency, quality and performance. This is due to the awareness of underlying network locality and proximity that provides effective and selective placement strategy



(a) Group Rank 72 (b) Group Rank 96 (c) Group Rank 148

Fig. 8. LST overlay network cost versus underlay network cost for group rank 72, 96 and 148 of the network model 0.

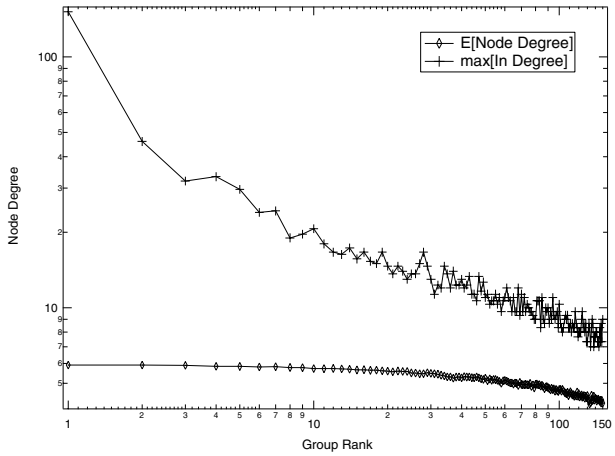


Fig. 9. Distribution of mean node degree and maximum in-degree in the LST SuperPeers overlay.

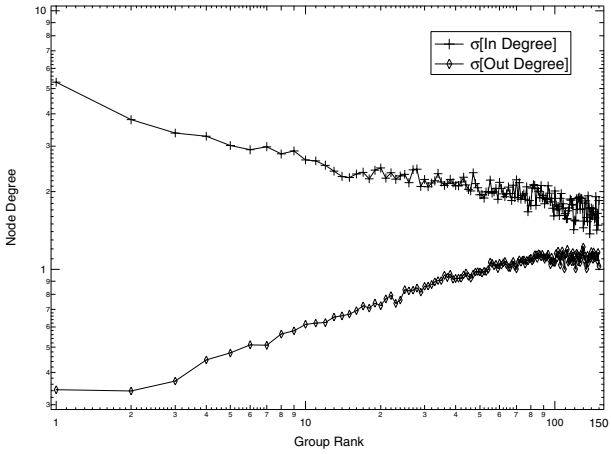


Fig. 10. Distribution of standard deviation of in-degree and out-degree in the LST SuperPeers overlay.

of the nodes in our geometric overlay hierarchy.

VI. Acknowledgements

The authors would like to thank Miguel Castro and Manuel Costa (Microsoft Research Cambridge, UK) for their discussions on the massive scale networks.

References

[1] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communication (JSAC)*, vol. 20, no. 8, October 2002.

[2] A. Czumaj, F. Ergun, L. Fortnow, A. Magen, I. Newman, R. Rubinfeld, and C. Sohler, "Sublinear-time approximation of euclidean minimum spanning tree," in *Proceedings of the* 14^{th} *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003, pp. 813–822.

[3] M. Kleis, E. K. Lua, and X. Zhou, "A case for lightweight superpeer topologies." in *Proceedings of the KiVS Kurzbeiträge und Workshop*, Germany, March 2005, pp. 185–188.

[4] ——, "Hierarchical peer-to-peer networks using lightweight superpeer topologies," in *Proceedings of the* 10^{th} *IEEE Symposium on Computers and Communications (IEEE ISCC 2005)*, La Manga del Mar Menor, Cartagena, Spain, June 27-30 2005.

[5] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of the* 11^{th} *Canadian Conference on Computational Geometry*, Vancouver, Canada, August 15-18 1999, pp. 51–54.

[6] X.-Y. Li, G. Calinescu, and P.-J. Wan, "Distributed construction of a planar spanner and routing for ad hoc wireless networks," in *Proceedings of the IEEE INFOCOM 2002*, vol. 3, New York, USA, June 23-27 2002, pp. 1268–1277.

[7] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder, "Sparse power efficient topology for wireless networks," in *Proceedings of the* 35^{th} *Hawaii International Conference on System Sciences (HICSS 2002)*, January 7-10 2002, pp. 3839–3848.

[8] E. K. Lua, J. Crowcroft, and M. Pias, "Highways: Proximity clustering for scalable peer-to-peer network," in *Proceedings of the* 4^{th} *IEEE International Conference on Peer-to-Peer Computing (IEEE P2P 2004)*, August 25-27 2004, pp. 266–267.

[9] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Tutorials and Surveys*, vol. 7, no. 2, pp. 72–93, July 2005.

[10] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft, "On the accuracy of embeddings for internet coordinate systems," in *Proceedings of the ACM SIGCOMM-Usenix Internet Measurement Conference 2005 (IMC 2005)*, October 19-21 2005.

[11] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," 5^{th} *Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

[12] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, "Lighthouses for scalable distributed location," in *Proceedings of the* 2^{nd} *International Workshop on Peer-to-Peer Systems*, February 2003.

[13] L. Plissonneau, J.-L. Costeux, and P. Brown, "Analysis of peer-to-peer traffic on adsl," in *Proceedings of the Passive and Active Measurement Workshop 2005 (PAM 2005)*, March 31 - April 1 2005.

[14] A. C.-C. Yao, "On constructing minimum spanning trees in k-dimensional space and related problems," *SIAM Journal on Computing*, vol. 11, pp. 721–736, 1982.

[15] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of the IEEE INFOCOM 1996*, vol. 2, San Francisco, CA, USA, March 1996, pp. 594–602.