Application-layer Multicast with Delaunay Triangulations

Jörg Liebeherr, Michael Nahas, Department of Computer Science, University of Virginia, Charlottesville, VA 22904

Abstract— **Recently,** *application-layer multicast* **has emerged as an attempt to support group applications without the need for a network-layer multicast protocol, such as IP multicast. In application-layer multicast, applications arrange themselves as a logical overlay network and transfer data within the overlay network. In this paper, Delaunay triangulations are investigated as an overlay network topology for application-layer multicast. An advantage of Delaunay triangulations is that each application can locally derive next-hop routing information without the need for a routing protocol in the overlay. A disadvantage of a Delaunay triangulation as an overlay topology is that the mapping of the overlay to the network-layer infrastructure may be suboptimal. It is shown that this disadvantage can be partially addressed with a hierarchical organization of Delaunay triangulations. Using network topology generators, the Delaunay triangulation is compared to other proposed overlay topologies for application-layer multicast.**

Key Words: GLOBAL INTERNET 2001, Application-Layer Multicasting, Multicasting.

I. Introduction

Recently, the limited availability of IP Multicast in wide-area networks has motivated researchers to investigate solutions which implement multicast support completely in the application layer. The general approach is that applications self-organize into a logical overlay network, and transfer data along the edges of the overlay network using unicast transport services. Here, each application communicates only with its neighbors in the overlay network. By forwarding packets from neighbor to neighbor, multicast forwarding is performed at the application layer[1] (see Figure 1).

Application-layer multicast has a number of attractive features: (a) There is no requirement for multicast support at the network layer; (b) There is no need to allocate a global group identifier (such as the IP multicast address); (c) Since data is sent via unicast, flow control, congestion control, and reliable delivery services that are available for unicast can be exploited. Drawbacks of application-layer multicast are that, since data is forwarded between end-systems, end-to-end latencies can be large. In addition, if multiple edges of the overlay are mapped to the same network link, multiple copies of the same data may be transmitted over this link, resulting in an inefficient use of bandwidth. Thus, important performance measures for overlay network topologies for application-layer multicast are the relative increase of end-to-end latencies and the increase of bandwidth requirements as compared to network-layer multicast.

This work is supported in part by the National Science Foundation through grants NCR-9624106 (CAREER), ANI-9870336, and and ANI-0085955.

[1] Application-layer multicast can be implemented by proxy-servers at the edges of the network, e.g., [3].

Previous works on overlay topologies for application-layer multicast fall into three groups. The first group of works constructs a single tree [8], [9], [7], [14] and all applications use this tree to disseminate data. A drawback of using a single tree is that the failure of a single application causes a partition of the overlay topology. The second group of works constructs a graph and transmits data along spanning trees which are embedded in the graph [3], [4]. A drawback of this approach is that the calculation of spanning trees requires running a routing protocol within the overlay, which adds complexity to the overlay network. In both groups above, nodes send probe messages to each other to measure latencies in the network-layer topology. These measurements are used to build an overlay network that is a good fit for the network-layer topology. The third group of works completely ignores the network-layer infrastructure when the overlay network is constructed. Here, the overlay is built as a graph with properties so that spanning trees can be easily embedded without the need for a routing protocol, e.g., as a hypercube [12]. A disadvantage of ignoring delays at the network layer when building an overlay network is that the resulting overlay network may not be a good match for the network-layer topology.

The work presented in this paper falls into the third group of overlay topologies. Specifically, this paper examines Delaunay triangulations as overlay topologies for application-layer multicast. We show that Delaunay triangulations can be built in a distributed fashion, and that multicast trees can be embedded in a Delaunay triangulation overlay without requiring a routing protocol in the overlay. We also present two methods which impose a hierarchical structure on a Delaunay triangulation, thereby achieving a better match of the overlay to the network-layer topology. We evaluate the Delaunay triangulations using synthesized network topologies, and show that Delaunay triangulations are a viable solution for application-layer multicast.

II. Delaunay Triangulation as an Overlay Network Topology

A Delaunay triangulation, for a set of vertices A, is a triangulation such that for each circumscribing circle of a triangle formed by three vertices in A, no vertex of A is in the interior of the circle. In Figure 2 we depict a Delaunay triangulation and the circumscribing circles of some of its triangles. Delaunay triangulations have been studied extensively in computational geometry [6] and have found applications in many areas of

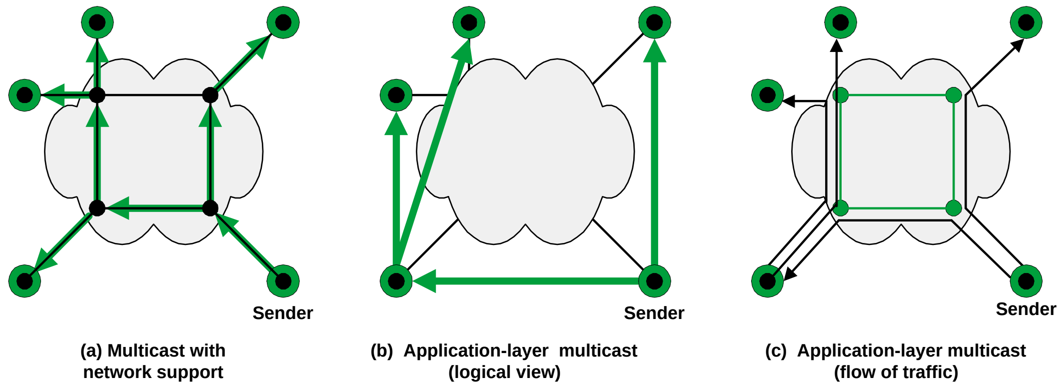


Fig. 1. Multicast with network support (a) performs packet replication in the network. Application-layer multicast (b) has all multicast functions in the end-systems. Here, data is transmitted in the network as unicast traffic between end-systems (c).

science and engineering, including communication networks, e.g., [1], [10].

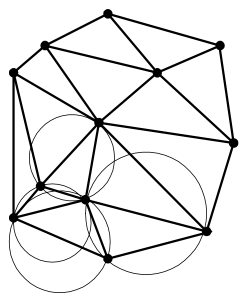


Fig. 2. A Delaunay Triangulation.

A. *Delaunay Triangulation Overlay Network*

For establishing a Delaunay triangulation overlay, each application, henceforth called 'node', is associated with a vertex in the plane with given (x,y) coordinates. The coordinates are assigned via some external mechanism (e.g., GPS, user input) and should reflect the geographical locations of nodes. Two nodes have a logical link in the overlay (i.e., are 'neighbors'), if their corresponding vertices are connected by an edge in the Delaunay triangulation constructed from the set of vertices from all nodes.

The properties of a Delaunay triangulation make it attractive as an overlay topology for application-layer multicast. The number of edges at a vertex in a Delaunay triangulation is generally small. Specifically, since each triangulation of n vertices has at most $3n - 3$ edges, the average number of edges at each vertex is less than 6. Even though, in the worst-case, the number of edges is $n - 1$, the maximum number of edges is usually small. [2]

Triangulations generally have a set of alternative non-overlapping routes between any pair of vertices. The existence of such alternative paths can be exploited by an application-layer overlay when nodes fail or are not responsive.

It is important to note that no network delay measurements are required to establish a Delaunay triangulation overlay network.

[2]The worst-case is created when $n - 1$ vertices form a circle, and the nth vertex is in the center of the circle.

In addition to the above, the Delaunay triangulation has two desirable, and in fact, with exception of [12], unique properties among proposed overlay topologies for multicast. First, once the topology is established, packet forwarding in the overlay can be performed without the need for a routing protocol. Second, the Delaunay triangulation can be established and maintained in a distributed fashion. Both properties will be discussed in the next subsections.

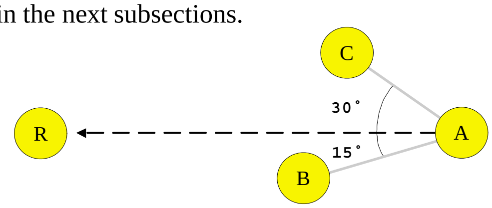


Fig. 3. Compass Routing. Node A has two neighbors, B and C, and computes B as the parent in the tree with root R, since the angle $\angle RAB$ is smaller than the angle $\angle RAC$.

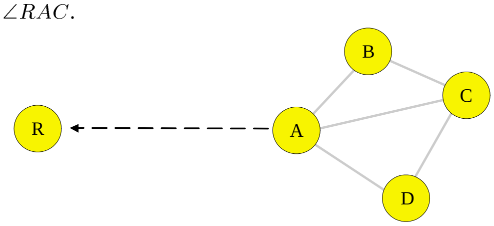


Fig. 4. Compass Routing. Node A determines that it is the parent node for node C, since the angle $\angle RCA$ is smaller than angles $\angle RCD$ and $\angle RCB$. Likewise, B and D determine that they are not the parent of node C, since $\angle RCA < \angle RCB$ and $\angle RCA < \angle RCD$, respectively.

B. *Compass Routing*

Multicast and unicast forwarding in the Delaunay triangulation is done using spanning trees that are embedded in the Delaunay triangulation overlay, and have the sender as the root of the tree. In the Delaunay triangulation, each node can locally determine its children nodes with respect to a given tree, using the coordinates of its neighbors and the coordinates of the sender.

Local forwarding decisions at nodes are done using the concept of *compass routing* [11]. The basic building block of compass routing is that a node A, for a root node R, computes a node B as the parent in the tree, if B is the neighbor with the smallest angle to R. This is illustrated in Figure 3. We can use the same concept for calculating children nodes. Specifically, a node A determines that a neighbor C is a child node with respect to a tree with root R, using the following considerations. Since the topology is a triangulation, the edge AC is a border of two triangles, say $\triangle ABC$ and $\triangle ACD$ (see Figure 4). A determines that C is a child node with respect to R, if selecting A leads to a smaller angle from C to R, than selecting B and D. If each node performs the above steps for determining children nodes, the nodes compute an embedded tree with root node R.

C. Building Delaunay Triangulations with Local Properties

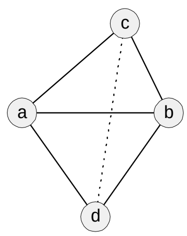


Fig. 5. Locally equiangular property. The property holds for triangles $\triangle abc$ and $\triangle abd$ if the minimum internal angle is at least as large as the minimum internal angle of triangles $\triangle acd$ and $\triangle cdb$.

A Delaunay triangulations can be defined in terms of a locally enforceable property. A triangulation is said to be *locally equiangular* [15] if, for every pair of triangles $\triangle acb$ and $\triangle abd$ that share a common edge (in this case, $\overline{a\,b}$), the minimum internal angle of triangles $\triangle acb$ and $\triangle abd$ is at least as large as the minimum internal angle of triangles $\triangle acd$ and $\triangle cbd$. This property is illustrated in Figure 5. In [15] it was shown that a locally equiangular triangulation is a Delaunay triangulation.

If a node knows the coordinates of its neighbors' neighbors it can determine (and enforce) that the local triangles satisfy the locally equiangular property. We have devised and implemented a network protocol where each node locally enforces the equiangular property, and, thereby, establishes and maintains a Delaunay triangulation. Space limitations prevent us from including a description of the protocol in this paper.

III. Improvements to Delaunay Triangulation Overlays

A Delaunay triangulation is a good overlay topology if coordinates of nodes reflect their relative geographical location and the network delays between nodes reflect the distances between their geographical locations. Since these conditions may not be met, that is, two nodes may be geographically close, but the shortest path through the network is long, we propose two methods that achieve a better mapping of the overlay topology to the network-layer infrastructure via hierarchically organized triangulations.

A. Delaunay Triangulations with Hierarchy

Many networks have a hierarchical topology, with backbone networks, regional networks, and local networks. We can account for this hierarchy by establishing a separate Delaunay triangulation for each local network, regional network, and so one, and representing each triangulation at a lower hierarchy level as a single node in the higher hierarchy level. In a Delaunay triangulation overlay this can be accomplished by enforcing that one node, e.g., the node with the highest coordinates, becomes the representative of the entire triangulation in the next hierarchy level. The representative for a triangulation acts as a gateway for all traffic that is not local to the triangulation. In Figure 6 we illustrate a network with a two-level hierarchy.

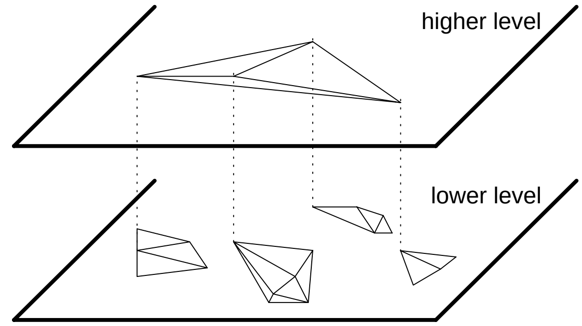


Fig. 6. Delaunay Triangulations with Hierarchy. One node in each lower-level triangulation represents the entire triangulation at the next hierarchy level.

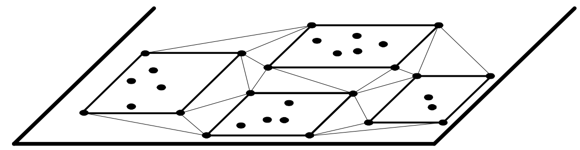


Fig. 7. Delaunay Triangulations with 'Bounding Boxes'. All traffic that crosses a bounding box must go through a node from the bounding box.

B. Delaunay Triangulations with 'Bounding Boxes'

Consider a subset of vertices in a triangulation. Suppose that four vertices are added to form a 'bounding box' around this subset, with coordinates (x_*, y_*), (x_*, y^*), (x^*, y_*), and (x^*, y^*). Suppose the coordinates of the added nodes are selected such that no vertex outside the bounding box is inside the circumscribing circles formed by two neighboring vertices from the bounding box and any vertex inside the bounding box. Then, when using compass routing (see Section II.2), the path between vertices inside and outside the bounding box must pass through one of the vertices of the bounding box (see Figure 7).

We exploit the notion of 'bounding boxes' to establish hierarchical organization of the Delaunay triangulation. The nodes in a local network are surrounded by a 'bounding box', all local networks in a regional network are surrounded by a larger bounding box, and so on. Bounding boxes can be established in several ways: Some nodes modify the coordinates of their corresponding vertices to those of the bounding box, or some dedicated node, e.g., the node whose vertex has the highest coordinates, is associated with all four vertices from the bounding box.

IV. EVALUATION

We present a comparative evaluation of the overlay topologies discussed in this paper and some overlay topologies from the literature. The basis for our evaluation is a software tool for network topology generation [2]. We present results for a so-called *Transit Stub* topology which generates a network with a 2-layer hierarchy.[3] The parameters for the topology are as follows. The network consists of 4 transit domains, each with 16 routers. The routers of each transit domain are randomly distributed over a 1024×1024 grid. There are 64 stub domains, each with 15 routers spread over a 32×32 grid. Each stub domain is connected to a transit domain router. Links between routers in transit and stub domains are set using the 'Waxman method' in [2]. The average number of links per router is approximately 3. Hosts are connected to a router of a stub domain and are distributed over a 4×4 grid with the stub domain router at the center of the grid. The total number of hosts in the entire network is varied from 2 to 512 hosts, in increments of powers of 2. The hosts are distributed uniformly over the stub domains.

We assume that all unicast traffic is carried on the shortest-delay path between two hosts, where the delay between two hosts is determined by the length of the shortest path in the generated topology. For each generated network topology, we construct a set of overlay networks. Each host participates in an overlay as a single node. We consider the following overlay topologies.

1. The *Delaunay triangulation* is as described in Section II. The coordinates of nodes are the grid coordinates of the hosts in the generated graph.
2. The *Delaunay triangulation (DT) with Hierarchy* was described in Section III.1. We use a two-layer hierarchy, which builds one triangulation for each stub domain, and one triangulation for all transit networks. In each stub domain, the node with the highest coordinates is the representative in the higher-level triangulation.
3. The *Delaunay Triangulation (DT) with Bounding Boxes* is from Section III.2. Here, one bounding box is generated for each stub domain. In each domain, one node represents all four nodes of the corresponding bounding box.
4. A *Minimum Spanning Tree (MST)* is an overlay network which builds a shared tree with minimum total delay, similar to the ALMI protocol [14].
5. A *Degree-3 minimum spanning tree* is used to represent a topology which will be very similar to topologies generated by the Yoid protocol [8]. In this overlay, each node has at most 3 links. We select a binary tree as an initial topology and use the update procedures described in [8] to improve the tree. We use the overlay network that is the result of 720 rounds of updates.
6. A *Degree-6 graph* is an overlay network similar to those created by the Narada protocol [4]. The algorithm establishes a mesh network, where each node has at most 6 logical links. For multicast delivery, the method uses a DVMRP routing algorithm for building per-source trees [5]. The protocol performs periodic unicast delay measurements and improves the mesh, based on these measurements. We show results of the overlay network which is obtained from 720 rounds of improvements.
7. The *Hypercube* topology builds an incomplete hypercube topology between hosts by completely ignoring the network topology [12]. Data is disseminated using trees which are embedded using an algorithm described in [13].

[3] We have conducted experiments with other topologies and will include the results in an upcoming technical report.

For a performance comparison of overlay networks we use the performance metrics *relative delay penalty (RDP)* and *stress*, which have been used in the related literature (e.g., [4]).

- The *relative delay penalty (RDP)* for two hosts is the ratio of the delay in the overlay to the delay of the shortest-delay unicast path.
- The *stress* of a network-layer link is the number of identical copies of a packet that traverse the link for any given tree embedded in an overlay network.

For network-layer multicasting, e.g., IP multicast, both RDP and link stress are equal to 1. We point out that there are many other measures which can be used to evaluate overlay networks, such as the robustness of the topology to link or node failures, the speed of convergence of the overlay topology, and the overhead of the routing protocol in terms of computation and bandwidth needs.

It is important to note that the results for stress and relative delay penalty are dependent on the randomly generated network topology. To account for some of the randomness, we present all numerical data as averages from 5 randomly generated network topologies, where for each network topology the same parameters are used.

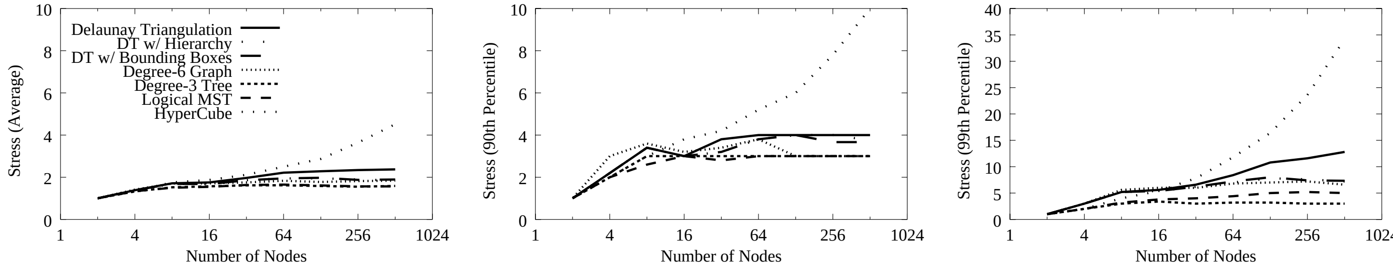
Results for Stress: In Figure 8 we show the values for stress for various overlays when the number of hosts is varied between 2 and 512. The results show the average values (Fig. 8(a)), the 90th percentile values (Fig. 8(b)), and the 99th percentile values (Fig. 8(c)) for the stress of links.

With exception of the hypercube topology, all overlay topologies show similar values for stress. The results in Fig. 8(c) show that the Delaunay triangulation yields higher stress values at the tail of the distribution. The figure also shows that our variants to the Delaunay triangulation, 'Bounding Boxes' and 'Hierarchy', result in noticeable improvements.

Results for Relative Delay Penalty (RDP): Figure 9 depicts the RDP values between all pairs of hosts. The results show that overlay networks which take into consideration the network-layer topology incur a lower RDP than our Delaunay triangulations. Here, the improvements presented in Section III do not result in considerable improvements.

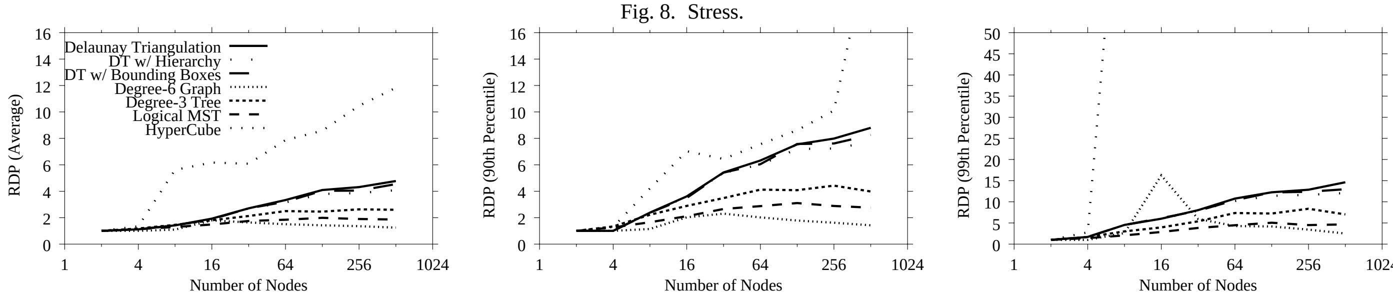
Note: The high value for the Degree-6 graph in Figure 9(c) for 16 nodes, is due to the fact that one of the five generated topologies contains a small number of paths with very high RDP values for the Degree-6 graph.

Considering that Delaunay triangulation are built without



(a) Average. (b) 90th percentile. (c) 99th percentile.

Fig. 8. Stress.



(a) Average. (b) 90th percentile. (c) 99th percentile.

Fig. 9. Relative Delay Penalty (RDP).

any knowledge of the network-layer topology, the numerical results are very encouraging. A comparison with the results of the hypercube overlay, the only other topology that is built without accounting for the network-layer infrastructure, indicates that the Delaunay triangulation is the preferred overlay network when measurements of the network-layer infrastructure are not available or not practical.

V. Conclusions

In this paper we have evaluated the use of Delaunay triangulations as overlay network topologies for application layer multicast. A key advantage of using a Delaunay triangulation is that multicast trees can be embedded in the topology without running a routing protocol in the overlay. Also, different from most existing approaches, no measurements between nodes are needed to establish the overlay networks. We showed approaches that can improve the matching of the Delaunay triangulations to a hierarchically organized network layer topology, and evaluated them using synthetically generated network topologies. In ongoing work we are experimenting with an implementation of a protocol which establishes a Delaunay triangulation. Preliminary results, run on a local cluster of 100 Linux PCs, show that the protocol can establish a Delaunay triangulation of several thousand applications in a few minutes (310 seconds for an overlay network with 10,000 nodes). In future work, we will report on these measurement experiments.

References

[1] F. Baccelli, D. Kofman, and J.-L. Rougier. Self organizing hierarchical multicast trees and their optimization. In *Proceedings of IEEE INFOCOM*, 1999.

[2] K. Calvert, M. Doar, and E. Zegura. Modeling internet topology. *IEEE Communications Magazine*, pages 160–163, 1997.

[3] Y. Chawathe, S. McCanne, and E. A. Brewer. RMX: Reliable multicast for heterogeneous networks. In *Proceedings of IEEE Infocom*, pages 795–804, 2000.

[4] Y. Chu, S. Rao, and H. Zhang. A case for end system multicast. In *Proceedings of ACM Sigmetrics*, pages 1–12, 2000.

[5] Y. K. Dalal and R. M. Metcalfe. Reverse path forwarding of broadcast packets. *Communications of the ACM*, 21(12):1040–1048, December 1978.

[6] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry, Algorithms and Applications*. Springer Verlag, 1997.

[7] J. Jannotti et. al. Overcast: Reliable multicasting with an overlay network. In *Proceedings of the Fourth Symposium on Operating Systems Design and Implementation*, San Diego, CA, October 2000. USENIX Association.

[8] P. Francis. Yoid: Extending the internet multicast architecture. Unrefereed Report, April 2000.

[9] D. A. Helder and S. Jamin. Banana tree protocol, an end-host multicast protocol. unpublished manuscript, July 2000.

[10] Limin Hu. *Distributed Algorithms for Packet Radio Networks*. PhD thesis, University of California at Berkeley, 1990.

[11] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG'99)*, 1999.

[12] J. Liebeherr and T. K. Beam. HyperCast: A protocol for maintaining multicast group members in a logical hypercube topology. In *Proc. First International Workshop on Networked Group Communication (NGC '99), Lecture Notes in Computer Science*, volume 1736, pages 72–89, 1999.

[13] J. Liebeherr and B. S. Sethi. A Scalable Control Topology for Multicast Communications. In *Proceedings IEEE Infocom '98*, 1998.

[14] D. Pendarakis, S. Sherlia, D. Verma, and M. Waldvogel. ALMI: An application level multicast infrastructure. Technical Report WUCS-00-17, Washington University in St. Louis, 2000.

[15] R. Sibson. Locally equiangular triangulations. *The Computer Journal*, 21(3):243–245, 1977.