Delaunay Triangulation Overlays

HyperCast Project

- **HyperCast** is a set of protocols for large-scale overlay multicasting and peer-to-peer networking

- **Motivating Research Problems:**
  - How to organize thousands of applications in a virtual overlay network?
  - How to do multicasting in very large overlay networks?
Overlay Multicasting

- **Logical overlay** resides on top of the Layer-3 network
- Data is transmitted between neighbors in the overlay
- No network support needed
- Overlay topology should match the Layer-3 infrastructure

HyperCast Approach

- **Build overlay network as a graph with known properties**
  - N-dimensional (incomplete) hypercube
  - Delaunay triangulation

- **Advantages:**
  - Achieve good load-balancing
  - Exploit symmetry
  - Next-hop routing in the overlay is free

- **Claim:** Can improve scalability of multicast and peer-to-peer networks by orders of magnitude over existing solutions
Hypercast Software

- Applications organize themselves to form a logical overlay network with a given topology
  - No central control
  - Dynamic membership

Data Transfer

- Data is distributed neighbor-to-neighbor in the overlay network
Delaunay Triangulation Overlays

Nodes are assigned x-y coordinates (e.g., based on geographic location)
The Voronoi region of a node is the region of the plane that is closer to this node than to any other node.

The Delaunay triangulation has edges between nodes in neighboring Voronoi regions.
**Delaunay Triangulation**

- An equivalent definition: A triangulation such that each circumscribing circle of a triangle formed by three vertices, no vertex of is in the interior of the circle.

**Locally Equiangular Property**

- Sibson 1977: **Maximize the minimum angle**

  For every convex quadrilateral formed by triangles ACB and ABD that share a common edge AB, the minimum internal angle of triangles ACB and ABD is at least as large as the minimum internal angle of triangles ACD and CBD.
Next-hop routing with Compass Routing

- A node’s parent in a spanning tree is its neighbor which forms the smallest angle with the root.
- A node need only know information on its neighbors – no routing protocol is needed for the overlay.

B is the Node’s Parent

Spanning tree when node (8,4) is root. The tree can be calculated by both parents and children.
Evaluation of Delaunay Triangulation overlays

- Delaunay triangulation can consider location of nodes in an (x,y) plane, but is not aware of the network topology

**Question:** How does Delaunay triangulation compare with other overlay topologies?

Hierarchical Delaunay Triangulation

- 2-level hierarchy of Delaunay triangulations
- The node with the lowest x-coordinate in a domain DT is a member in 2 triangulations
**Multipoint Delaunay Triangulation**

- Different (“implicit”) hierarchical organization
- “Virtual nodes” are positioned to form a “bounding box” around a cluster of nodes. All traffic to nodes in a cluster goes through one of the virtual nodes

**Overlay Topologies**

- Delaunay Triangulation and variants
  - DT
  - Hierarchical DT
  - Multipoint DT

Hypercube

- Degree-6 Graph
  - Similar to graphs generated in Narada

Degree-3 Tree
  - Similar to graphs generated in Yoid

Logical MST
  - Minimum Spanning Tree

 overlays used by HyperCast

 overlays that assume knowledge of network topology
Transit-Stub Network

Transit-Stub
- GeorgiaTech topology generator
- 4 transit domains
- 4×16 stub domains
- 1024 total routers
- 128 hosts on stub domain

Evaluation of Overlays

- Simulation:
  - Network with 1024 routers (“Transit-Stub” topology)
  - 2 - 512 hosts

- Performance measures for trees embedded in an overlay network:
  - Degree of a node in an embedded tree
  - “Relative Delay Penalty”: Ratio of delay in overlay to shortest path delay
  - “Stress”: Number of duplicate transmissions over a physical link
Illustration of “Stress” and “Relative Delay Penalty”

Unicast delay A→B : 4
Delay A→B in overlay: 6
Relative delay penalty for A→B: 1.5

Average Relative Delay Penalty
90th Percentile of Relative Delay Penalty

Average “Stress”
90th Percentile of “Stress”

![Graph showing 90th percentile of stress against number of nodes]

© Jörg Liebeherr 2003

The DT Protocol

Protocol which organizes members of a network in a Delaunay Triangulation

- Each member only knows its neighbors
- “soft-state” protocol

Topics:

- Nodes and Neighbors
- Example: A node joins
- State Diagram
- Rendezvous
- Measurement Experiments

© Jörg Liebeherr 2003
Each node sends Hello messages to its neighbors periodically.

- Each Hello contains the clockwise (CW) and counterclockwise (CCW) neighbors.
- Receiver of a Hello runs a “Neighbor test” (locally equiangular prop.)
- CW and CCW are used to detect new neighbors.

### Neighborhood Table of 10,8

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>CW</th>
<th>CCW</th>
</tr>
</thead>
<tbody>
<tr>
<td>5,2</td>
<td>12,0</td>
<td>4,9</td>
</tr>
<tr>
<td>4,9</td>
<td>5,2</td>
<td>-</td>
</tr>
<tr>
<td>12,0</td>
<td>-</td>
<td>10,8</td>
</tr>
</tbody>
</table>
A node that wants to join the triangulation contacts a node that is "close"

Node (5,2) updates its Voronoi region, and the triangulation
(5,2) sends a Hello which contains info for contacting its clockwise and counterclockwise neighbors.

(8,4) contacts these neighbors...
… which update their respective Voronoi regions.

(4,9) and (12,0) send Hellos and provide info for contacting their respective clockwise and counterclockwise neighbors.
(8,4) contacts the new neighbor (10,8) ...

Hello

…which updates its Voronoi region…
...and responds with a Hello

This completes the update of the Voronoi regions and the Delaunay Triangulation
Rendezvous Methods

- **Rendezvous Problems:**
  - How does a new node detect a member of the overlay?
  - How does the overlay repair a partition?

- **Three solutions:**
  1. Announcement via broadcast
  2. Use of a rendezvous server
  3. Use 'likely' members ("Buddy List")

**Rendezvous Method 1:** Announcement via broadcast (e.g., using IP Multicast)
**Rendezvous Method 1:**
A *Leader* is a node with a Y-coordinate higher than any of its neighbors.

**Rendezvous Method 2:**
New node and leader contact a rendezvous server. Server keeps a cache of some other nodes.
**Rendezvous Method 3:** Each node has a list of “likely” members of the overlay network.

New node with Buddy List: (12,0) (4,9)

**State Diagram of a Node**

[State diagram showing node states and transitions]
Sub-states of a Node

- A node is **stable** when all nodes that appear in the CW and CCW neighbor columns of the neighborhood table also appear in the neighbor column.

Measurement Experiments

- **Experimental Platform:**
  Centurion cluster at UVA (cluster of 300 Linux PCs)
  - 2 to 10,000 overlay members
  - 1–100 members per PC

- **Random coordinate assignments**
Experiment: Adding Members

How long does it take to add M members to an overlay network of N members?

Experiment: Throughput of Multicasting

100 MB bulk transfer for N=2-100 members (1 node per PC)
10 MB bulk transfer for N=20-1000 members (10 nodes per PC)
Experiment: Delay

100 MB bulk transfer for N=2-100 members (1 node per PC)
10 MB bulk transfer for N=20-1000 members (10 nodes per PC)