On the “Treeness” of Internet Latency and Bandwidth

Venugopalan Rama Ramasubramanian
Microsoft Research Silicon Valley
Dahlia Malkhi, Fabian Kuhn, Mahesh Balakrishnan
Archit Gupta and Aditya Akella
Introduction

- Internet is inherently hierarchical
  - tiered organization of ISPs
  - densely connected core with sparsely connected edges
  - valley-free routing making Internet paths go up and down

source: technologyreview.com

source: caida.org
Goals/Contributions

1. study “treeness” of Internet path measures such as bandwidth and latency
   ▪ simple parameter to quantify treeness of metrics

2. construct tree-based models to represent Internet measures
   ▪ algorithms to embed bandwidth and latency into trees

3. explore practical implications of tree models
   ▪ bandwidth and latency estimation
   ▪ selection of closest and best-provisioned servers
   ▪ topological clustering of hosts
Is the Internet a Tree?

- The 4-Points Condition

\[ d(s,u) + d(v,t) = d(s,t) + d(u,v) \geq d(s,v) + d(t,u) \]

Distance metric is tree metric \( \iff \) 4PC is satisfied for every 4 points

- The \( \varepsilon \)-4-Points Condition [PODC 07]

\[ d(s,u) + d(v,t) = d(s,t) + d(u,v) + 2\varepsilon \min\{d(s,v),d(t,u)\} \]
Treeness of Latency and Bandwidth

- $4\text{PC}-\varepsilon \in [0,1]$, zero $\Rightarrow$ perfect tree, one $\Rightarrow$ any metric
  - analyzed on two latency and one bandwidth datasets

![Graph showing CDF for different metrics](image-url)
Metric Embedding into Trees

- end hosts (A, B, C, R) are leaf nodes
Metric Embedding into Trees

- end hosts (A, B, C, R) are leaf nodes
- inner nodes (s, t) are “virtual”
Metric Embedding into Trees

- end hosts (A, B, C, R) are leaf nodes
- inner nodes (s, t) are “virtual”
- edge weights model network measure
Tree Embeddings and “Coordinates”

- distance label = path to the root
  - example: A: (s, t, R) and C: (t, R)

- estimated metric = distance on path
  - latency: \(d(A, C) = d(A, s) + d(s, t) + d(t, C)\)

- convenience of traditional network coordinates
  - but, longer in size
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements
- node R acts as the root of the tree
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements
- node R acts as the root of the tree
- each node has an anchor
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements
- node R acts as the root of the tree
- each node has an anchor
  - distance to root R and anchor A is exactly preserved
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements
- node R acts as the root of the tree
- each node has an anchor
  - distance to root R and anchor A is exactly preserved
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements
- node R acts as the root of the tree
- each node has an anchor
  - distance to root R and anchor A is exactly preserved
- anchor maximizes Gromov product
  - max. $d(R,A) + d(R,B) - d(A,B)$
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements

- node R acts as the root of the tree

- each node has an anchor
  - distance to root R and anchor A is exactly preserved

- anchor maximizes Gromov product
  - \( \max. d(R,A) + d(R,B) - d(A,B) \)
Basic Tree-Embedding Algorithm

- pair-wise end-to-end measurements
- node R acts as the root of the tree
- each node has an anchor
  - distance to root R and anchor A is exactly preserved
- anchor maximizes Gromov product
  - max. $d(R,A) + d(R,B) - d(A,B)$
Properties of Basic Algorithm

- zero distortion for a tree metric
  - through Gromov product
Properties of Basic Algorithm

- zero distortion for a tree metric
  - through Gromov product

- bounded worst-case distortion [PODC 07]
  - two-pass algorithm that optimizes the join order
  - upper-bound \((1 + \varepsilon)^{c_1 \cdot \log n}\)
Practical Improvements

- **improve accuracy**
  - multiple trees at different roots
  - mitigate inaccuracies due to choice of a single root

- **balance tree**
  - heuristic to balance the distribution of anchors
  - short distance labels

- **reduce measurements**
  - approximate anchor selection
  - find anchor through a search on the tree
Embedding Bandwidth-Like Metrics

- bandwidth is an approximate tree metric
  - edge bottlenecks induce tree metrics

- natural embedding
  - edge weights represent bandwidth
  - tree bandwidth is the bottleneck bandwidth on tree path

- black-box embedding
  - treat bandwidth as a distance measure
  - invert values before and after embedding
Evaluation

- implemented as Sequoia
- aspects evaluated
  1. accuracy of latency/bandwidth estimation
  2. effectiveness of server selection
  3. correlations with Internet topology
- datasets used

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Hosts</th>
<th>Measurements</th>
<th>Δ&lt;&gt;</th>
<th>Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-PlanetLab Latency</td>
<td>125</td>
<td>15,625</td>
<td>15%</td>
<td></td>
</tr>
<tr>
<td>HP-PlanetLab Bandwidth</td>
<td>396</td>
<td>65,077</td>
<td>40%</td>
<td></td>
</tr>
<tr>
<td>Cornell-King Latency</td>
<td>2500</td>
<td>3,123,750</td>
<td>22%</td>
<td></td>
</tr>
</tbody>
</table>
Results:
Latency Estimation

- UC-PlanetLab Latency Sequoia 5 Trees
- Cornell-King Latency Sequoia 15 Trees
- UC-PlanetLab Latency Vivaldi 2+H
- Cornell-King Latency Vivaldi 2+H
Results: Bandwidth Estimation

C D F

Relative Error

HP-PlanetLab Bandwidth Sequoia 5 Trees
HP-PlanetLab Bandwidth Vivaldi 2+H
Results: Server Selection

Closest Host

Best-Provisioned Host

Error (ms)

Relative Error

UC-PlanetLab Latency Sequoia 5 Trees
Cornell-King Latency Sequoia 15 Trees
HP-PlanetLab Bandwidth Sequoia 5 Trees
Results:
Sequoia Tree for PlanetLab
Results:
Clustering of European Nodes

UK and Ireland

Spain and Portugal

Scandinavia
Conclusions

- Internet measures are approximate tree metrics
- Tree embeddings can model bandwidth and latency
- Surprising correlations with Internet topology

Email: rama@microsoft.com
http://research.microsoft.com/research/sv/sequoia