Minimum-Delay Overlay Multicast

Kianoosh Mokhtarian
Department of Electrical and Computer Engineering
University of Toronto
Background: Overlay Multicast

Delay-sensitive multicast
- Event notification
- VoIP conferencing
- Online games

Dynamic receiver set

Multicast sessions in network? \(\times\)
Send directly to each receiver? \(\times\)
Source-based multicast trees? \(\checkmark\)
- Regular shortest path tree?
Node Degree Problem

Most shortest paths within few hops of source

=> Large node degrees
SPTDelay(x, w) = d(x, y) + d(y, z) + d(z, w)
**Node Degree Problem**

\[ \text{SPTDelay}(x, w) = d(x, y) + d(y, z) + d(z, w) \]

\[ \text{ActualDelay}(x, w) = \]
Node Degree Problem

\[
\text{SPTDelay}(x, w) = d(x, y) + d(y, z) + d(z, w)
\]

\[
\text{ActualDelay}(x, w) = \Delta(x) \times 3 + 
\]
Node Degree Problem

\[
\text{SPTDelay}(x, w) = d(x, y) + d(y, z) + d(z, w)
\]

\[
\text{ActualDelay}(x, w) = \Delta(x) \ast 3 + d(x, y) + \]

\[
\]
Node Degree Problem

\[ \text{SPTDelay}(x, w) = d(x, y) + d(y, z) + d(z, w) \]

\[ \Delta(x) \times 3 + d(x, y) + \Delta(y) \times 3 + \]
Node Degree Problem

\[ \text{SPTDelay}(x, w) = d(x, y) + d(y, z) + d(z, w) \]

\[ \text{ActualDelay}(x, w) = \Delta(x) \times 3 + d(x, y) + \Delta(y) \times 3 + d(y, z) + \]
SPTDelay(x, w) = \( d(x, y) + d(y, z) + d(z, w) \)

ActualDelay(x, w) = 
\[ \Delta(x) \times 3 + d(x, y) + \Delta(y) \times 3 + d(y, z) + \Delta(z) \times 2 + \]
SPTDelay(x, w) = d(x, y) + d(y, z) + d(z, w)

ActualDelay(x, w) =

Δ(x) * 3 +

\[ \Delta(x) \times 3 + d(x, y) + \Delta(y) \times 3 + d(y, z) + \Delta(z) \times 2 + d(z, w) \]
Previous Works

Ignore nodal delays

Tree with bounded node degrees
  Too coarse grained
  Thresholds?

Tree with minimum node+link delay?
  [Brosh et al. 2007]
  Only MinMax delay
  Outperformed (both tree efficiency and running time)
System Model

On-demand tree
Source calculates and attaches to data

1. Full tree; $O(N)$ overhead
2. Compact with Bloom Filters

Intermediary nodes only forward data

*No state maintained*

*Path vector* based *partial* view of topology known at node
Problem Statement

Given: Overlay view, source, receiver set

Goal: A multicast tree that minimizes:

Sum delay to all receivers

\[ \text{MSDOM} \ (\text{MinSum Delay Overlay Multicast}) \]

Max delay to the farthest receiver

\[ \text{MMDOM} \ (\text{MinMax Delay Overlay Multicast}) \]
Hardness

Both versions NP-hard

Inapproximable with $O(\log N)$

Best possible approximation guarantee =

$O(\log N)$ times worse than the optimal minimum delay
Our Solutions

(Recall: MS=MinSum, MM=MinMax)

**MSDOM-e** and **MMDOM-e** (delay-efficient)

Outperform prev works in both **efficiency** and **running time**

**MSDOM-f** and **MMDOM-f** (fast)

Applicable to new scales

Efficiencies proven empirically

Different real-world data sets, overlay models, various sensitivity tests, ...
1. For each edge \((u, v)\) s.t. \(u \in\) tree built so far, \(v \in\) tree:

   Cost\([v, u]\): delay to \(v\) (if attached to tree via \(u\)) + delay increase to other descendants of \(u\) (e.g. \(w\))

2. Add the minimum-cost node

3. Update delay[all \(u\)'s descen-

4. Repeat step 1

Complexity: \(O(N^3)\)
Farthest node first

1. Run "a modified Dijkstra algorithm" on overlay
   - From multiple sources (nodes currently in tree)
   - Distance: exact link+nodal delay (in the current tree)

2. Add the farthest node (longest path) to tree

3. Repeat step 1

Complexity: $O(N^3)$
MSDOM-\textit{f} and MMDOM-\textit{f}

1. Calculate the regular shortest-path tree
   i.e. merge shortest paths from the path vector table

2. \textit{Refine} the tree
   Move nodes around to minimize sum/max delay
Re-route to some nodes:

- MinSum: from top to bottom
- MinMax: from farthest to close

For each node $u$:

- what if detaching $u$ and attaching it somewhere else?
  - i.e. other paths from source to $u$ given in routing table

Complexity: $O(N^2 \log N)$
Two real-world datasets for node distances

DS2 [Zhang et al. 2010]; 4000 hosts
Meridian [Wong et al. 2005]; 2500 hosts
Evaluation: Setup

Two real-world datasets for node distances
Three different overlay models
  - Small-world
  - Scale-free (power-law)
  - Random
Two real-world datasets for node distances

Three different overlay models

Various sensitivity tests

Overlay size \((N)\)

Overlay connectivity \((D: \text{average degree in overlay})\)

Receiver group size \((|R|)\)

Average nodal delay per every copy of message \((\Delta)\)

Dynamics of nodal delays \((\Delta(\text{at forwarding time}) - \Delta(\text{announced}))\)
Two real-world datasets for node distances
Three different overlay models
Various sensitivity tests

Algorithms:

- Regular SPT (Shortest Path Tree)
- BLS [Brosh et al. 2007]: only for MinMax
- MLRS [Malouch et al. 2002]: only for MinSum
- Our MxDOM algorithm suite
Results: Different Overlay Scales

\( N = 200 \ldots 4000 \)

\(|R| = N-1; |D| = N / 5 \)

Average \( \Delta = 100 \text{ ms} \)

DS2 dataset

Small-world overlays
Different levels of overlay connectivity
Different receiver group sizes
Different nodal delays
Dynamics (uncertainty) of nodal delays

=> Same trend between the algorithms
Results: Different Datasets and Overlay Models (MinSum)

- **SPT**
- **MLRS**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SPT Delay (ms)</th>
<th>MLRS Delay (ms)</th>
<th>MSDOM-e Delay (ms)</th>
<th>MSDOM-f Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS2</td>
<td>666</td>
<td>388</td>
<td>303</td>
<td>217</td>
</tr>
<tr>
<td>Meridian</td>
<td>597</td>
<td>303</td>
<td>422</td>
<td>45</td>
</tr>
<tr>
<td>Power law</td>
<td>678</td>
<td>398</td>
<td>46</td>
<td>45</td>
</tr>
<tr>
<td>Random</td>
<td>697</td>
<td>398</td>
<td>46</td>
<td>519</td>
</tr>
<tr>
<td>Small world</td>
<td>519</td>
<td>45</td>
<td>45</td>
<td>45</td>
</tr>
</tbody>
</table>
Results: Different Datasets and Overlay Models (MinMax)
**Take-Away**

<table>
<thead>
<tr>
<th>Algorithm Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSDOM / MMDOM</td>
<td>Algorithm names standing for: MinSum and MinMax Delay Overlay Multicast</td>
</tr>
<tr>
<td>MSDOM-e / MMDOM-e</td>
<td>Highest delay efficiency: outperform previous approaches in both tree efficiency and running time. Best for overlays of up to a few hundred nodes.</td>
</tr>
<tr>
<td>MSDOM-f / MMDOM-f</td>
<td>Optimized to be fast. Best for overlays of up to a few thousand nodes, making trees with ~50% smaller delay than SPT, the only applicable alternative for this scale.</td>
</tr>
</tbody>
</table>

**Next step**

Adapt the existing tree upon updates in the receiver set


## Take-Away

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSDOM / MMDOM</td>
<td>Algorithm names standing for: MinSum and Min-Max Overlay Multicast.</td>
</tr>
<tr>
<td>MSDOM-e / MMDOM-e</td>
<td>Our algorithms with highest delay-efficiency. Outperform previous approaches in both tree efficiency and running time. Best for overlays of up to a few hundred nodes.</td>
</tr>
<tr>
<td>MSDOM-f / MMDOM-f</td>
<td>Our algorithms optimized to be fast. Best for overlays of up to a few thousand nodes, making trees with ~50% smaller delay than SPT, the only applicable alternative for this scale.</td>
</tr>
<tr>
<td>iMSDOM / iMSDOM</td>
<td>Our algorithms to incrementally build multicast trees w.r.t. recent changes in receiver group. Make delay-efficient trees in nearly 0 time. Best for multicasting a stream of messages to a group of receivers churning continuously.</td>
</tr>
</tbody>
</table>