Approximation and Heuristic Algorithms for Minimum-Delay Application Layer Multicast Trees

Eli Brosh and Yuval Shavitt
Tel-Aviv University
Application-Layer Multicast

An alternative to IP multicast - provide multicast functionality at the application layer.

- End-Hosts perform packet duplication and routing.
- End-Hosts construct **Overlay Network** on unicast infrastructure
- Overcomes IP multicast limitations (scalability, deployment, etc)

**Challenge:** construct efficient overlay trees, minimizing the performance penalty involved with AL processing.
Efficiency of Multicast Trees

- Our goal is to construct delay optimized multicast trees.

- Many proposals attempt to minimize the multicasting delay: Narada [CRZ00], Yoid [F99], Almi [ST02], Omni [BKKBS03], and others.

- Common Properties
  - Build short diameter (height) trees with constrained degrees
  - Short diameter \(\Rightarrow\) Low delay
  - Low degree \(\Rightarrow\) Low stress and bandwidth utilization

- Implies a Dual Cost optimization objective
The Problem

- Arbitrary selection of degrees
  - Requires trial and error
- Neglect serialized message distribution
  - Scalability problem
Our Contributions

- New overlay network model
  - Mathematical generalization of Cidon’s model [CGK95].
  - Map the node load to delay penalty factor
    \[\Rightarrow\] Quantify multicast performance using a single delay measure.

- New approximation and heuristic algorithms
  - Generate trees that intrinsically balance short latency with small degree.

- Performance analysis of structured overlay topologies
The Overlay Network Model

- A complete directed graph $G=(V,E)$
- Communication cost function $c : E \rightarrow R^+$
- Processing cost function $p : V \rightarrow R^+$
- **Sequential communication**, $p(v)$ is the time the sender host is busy minimum time interval between message transmissions of host $v$
The Minimum Delay Multicast (MDM) Problem

Given:
- Directed complete graph $G=(V,E)$;
  - processing cost $p(v), v \in V$;
  - communication cost $c(e), e \in E$;
- Multicast group $M \subseteq V$;
- Source host $s \in M$;

Find:
- a scheme that \textbf{minimizes the delay} by which all the hosts in $M$ receive a message from $s$.

Assumption: only the hosts in $M$ are allowed to participate in the distribution.
The Ordered Tree Solution

- Optimal solution is represented by an ordered tree $T$ which spans $M$, rooted at $s$.
- The $i$-th outgoing edge of node $u$ corresponds to the $i$-th transmission from host $u$

Notations

Reception delay of $v$, $t_T(v)$

Tree cost, $\max_{v \in M} \{t_T(v)\}$

By Def. $t_T(s) = 0$
Optimal Multicast

- Given a multicast tree $T=(V,E)$ one can calculate the optimal ordering using a simple recursive computation, working bottom-up.
  - Idea: The i-th delivery goes to the i-th largest cost subtree
  - Time complexity $\Theta(n)$

- Neglect the ordering and focus on finding optimal trees.
- The optimal solution will be a ‘non-lazy’ multicast scheme

- The optimal multicast problem is NP-Complete
  - Reduction from the telephone broadcast problem
Related work
Parallel Computation Models

- **Homogenous models**
  - Cidon *et al.* [CGK95], high-speed network model
    - Optimal tree-based multicast algorithm.
    - The tree delay is logarithmic in the size of multicast group.
  - Postal [BK95], LogP [KPSS93], Active Networks [RS01]

- **The Heterogeneous Postal Model** [BGNSS01]
  - Incorporates communication latency cost function $\lambda$ and a sending time function $s$.
  - $\log(k)$ approximation algorithm for optimal multicast
    - $k$ is the size of multicast group.
    - Supports only undirected graphs
MDM Approximation

Postal approximation cannot be used directly to solve MDM due to distribution timing differences.

**Our approach**: devise an approximation algorithm, **Approx-MDM**, based on a modified version of the postal approx.

**Theorem 1**
The approximation ratio of Approx-MDM is

\[
(OPT + p_{\text{max}} - p_{\text{min}}) O(\log n)
\]

- Cost of optimal tree
- Maximal processing cost
- Minimal processing cost
- Size of the multicast group
Heuristic Algorithm

**Motivation**: Develop an algorithm with low computational overhead (Approx-MDM is $\Theta(n^7)$)

Greedy approach: Largest Ready Time First

**Algorithm Heuristic-MDM**

**Init**: Add $s$ to an empty tree

1. Compute the minimum reception delay of each non-notified host
2. Select the non-notified host with maximum reception delay
3. Add this host and the minimum latency path to the constructed tree

Repeat 1-3 till all hosts are notified
Heuristic Algorithm - Cont.

- Minimum latency path is computed using All-Pairs Shortest-Path (Floyd-Warshall) with weight matrix $W=(w_{vi,vj})$ defined as:

\[
w_{vi,vj} = \begin{cases} 
  p(v_i) + c(v_i,v_j) & \text{if } v_i \neq v_j \\
  0 & \text{otherwise}
\end{cases}
\]

- Time complexity $\Theta(n^3)$
- Supports arbitrary directed graphs
Lemma 2:
The approximation ratio of Heuristic-MDM is $\Omega(n^{0.5})$

Proof: Assume graph with $n+1$ hosts, where $p(v)=1$, $v \in V$,

- Cost of heuristic tree $n$
- Cost of optimal tree $(1+\delta) O(n^{0.5}) \Rightarrow The \ lemma \ follows.$
Simulation Results

- The multicast delay for a clique topology with random costs uniformly distributed on the interval $[1,10]$

- Lower bound: the weight of the longest path in the SPT
Simulations: Internet-Like costs

- The multicast delay for a clique topology with random communication costs from $[1,10]$ and unit processing costs

- SPT has almost linear growth rate for large sizes
Simulations Summary

- The heuristic algorithm has a **similar or better** performance than the approximation algorithm

- Heuristic trees are scalable for large group sizes
  - Near optimal result
  - Logarithmic like growth rate
Summary

- Solutions for delay-sensitive Applications

- New overlay communication model

- Cost effective heuristic algorithm
  - Scalable solution with near optimal results
  - Simple implementation, applicable for centralized server based and P2P overlay systems.

- New performance bounds for several degree constrained graphs
Future Work

- Examine graphs in which the triangle equality holds, and attempt to devise better bounds.

- Develop a distributed version of our algorithm, and explore its efficiency and applicability in real-world communication networks.