
R.C. Chakinala, A. Kumarasubramanian, Kofi A. Laing
R. Manokaran, C. Pandu Rangan, R. Rajaraman
Push and Pull

Source

Sink
Push and Pull

Source

Push

Sink
Push and Pull

Push

Source

push

Sink
Push and Pull

**Push**

Source → push → Sink

**Pull**
Push and Pull

**Push**
- Source
  - **push**
  - Sink

**Pull**
- Source
  - **query**
  - response
Push and Pull

Push

Source \( \rightarrow \) Sink

Push

Pull

Mixed

Store
Push and Pull

**Push**
- Source → Sink
- Push

**Pull**
- Source ← Sink
- Response
- Query

**Mixed**
- Source → Store
Push and Pull

Push

Source ➞ Sink
push

Pull

Source ➞ Store
query
response

Mixed

Source ➞ Store
A Simple Example

Using average source and sink frequencies.

Sources

\[ x \]

\[ x \]

Sink

\[ y \]

<table>
<thead>
<tr>
<th></th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
<td>7y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
<td>2x + 3y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
<td>3y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Push**

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>$4x$</td>
<td>$4x$</td>
</tr>
<tr>
<td>Pull</td>
<td>$6y$</td>
<td>$7y$</td>
</tr>
<tr>
<td>Mixed</td>
<td>$2x + 2y$</td>
<td>$2x + 3y$</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>$2y &lt; 2x &lt; 4y$</td>
<td>$3y &lt; 2x &lt; 4y$</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Push**

<table>
<thead>
<tr>
<th>Source</th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
<td>7y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
<td>2x + 3y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
<td>3y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example
Using average source and sink frequencies.

**Push**

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

Pull

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push 4x</td>
<td>Push 4x</td>
</tr>
<tr>
<td>Pull 6y</td>
<td>Pull 7y</td>
</tr>
<tr>
<td>Mixed 2x + 2y</td>
<td>Mixed 2x + 3y</td>
</tr>
<tr>
<td><strong>Mixed is Best</strong></td>
<td>2y &lt; 2x &lt; 4y</td>
</tr>
<tr>
<td></td>
<td>3y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Pull**

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$y$</td>
</tr>
<tr>
<td>$x$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>$4x$</td>
</tr>
<tr>
<td>Pull</td>
<td>$6y$</td>
</tr>
<tr>
<td>Mixed</td>
<td>$2x + 2y$</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>$2y &lt; 2x &lt; 4y$</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

Pull

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sources</th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
<td>7y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
<td>2x + 3y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
<td>3y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Pull**

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4$x$</td>
</tr>
<tr>
<td>Pull</td>
<td>6$y$</td>
</tr>
<tr>
<td>Mixed</td>
<td>$2x + 2y$</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>$2y &lt; 2x &lt; 4y$</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Pull**

<table>
<thead>
<tr>
<th></th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
<td>7y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
<td>2x + 3y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
<td>3y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Pull**

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>$4x$</td>
<td>$4x$</td>
</tr>
<tr>
<td>Pull</td>
<td>$6y$</td>
<td>$7y$</td>
</tr>
<tr>
<td>Mixed</td>
<td>$2x + 2y$</td>
<td>$2x + 3y$</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>$2y &lt; 2x &lt; 4y$</td>
<td>$3y &lt; 2x &lt; 4y$</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

Mixed

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

Mixed is Best

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Mixed**

<table>
<thead>
<tr>
<th></th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
<td>7y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
<td>2x + 3y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
<td>3y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Mixed**

<table>
<thead>
<tr>
<th>Sources</th>
<th>Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td></td>
</tr>
</tbody>
</table>

Mixed is Best

<table>
<thead>
<tr>
<th></th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>$4x$</td>
<td>$4x$</td>
</tr>
<tr>
<td>Pull</td>
<td>$6y$</td>
<td>$7y$</td>
</tr>
<tr>
<td>Mixed</td>
<td>$2x + 2y$</td>
<td>$2x + 3y$</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>$2y &lt; 2x &lt; 4y$</td>
<td>$3y &lt; 2x &lt; 4y$</td>
</tr>
</tbody>
</table>
A Simple Example

Using average source and sink frequencies.

**Mixed**

<table>
<thead>
<tr>
<th></th>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
<td>7y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
<td>2x + 3y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
<td>3y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
**A Simple Example**

Using average source and sink frequencies.

### Mixed

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>No Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push</td>
<td>4x</td>
</tr>
<tr>
<td>Pull</td>
<td>6y</td>
</tr>
<tr>
<td>Mixed</td>
<td>2x + 2y</td>
</tr>
<tr>
<td>Mixed is Best</td>
<td>2y &lt; 2x &lt; 4y</td>
</tr>
</tbody>
</table>
General Problem — High Level

- **INPUTS:** Graph $G = (V, E)$ with:
  - cost of updating set of stores: $\text{SetC} : V \times \text{Powerset}(V) \rightarrow \mathbb{R}^+$
  - **Source Set** $\mathcal{P} \subseteq V$, **Sink Set** $\mathcal{Q} \subseteq V$
  - For every source $i \in \mathcal{P}$, a **source frequency** $p_i$
  - For every sink $j \in \mathcal{Q}$, a **sink frequency** $q_j$
  - For every sink $j \in \mathcal{Q}$, an **interest set** $I_j$
General Problem — High Level

- **INPUTS:** Graph $G = (V, E)$ with:
  - cost of updating set of stores: $\text{SetC} : V \times \text{Powerset}(V) \rightarrow \mathbb{R}^+$
  - **Source Set** $\mathcal{P} \subseteq V$, **Sink Set** $\mathcal{Q} \subseteq V$
  - For every source $i \in \mathcal{P}$, a **source frequency** $p_i$
  - For every sink $j \in \mathcal{Q}$, a **sink frequency** $q_j$
  - For every sink $j \in \mathcal{Q}$, an **interest set** $I_j$

- **OUTPUTS:**
  - For every source $i \in \mathcal{P}$, a **Push set** $P_i$
  - For every sink $j \in \mathcal{Q}$, a **Pull Set** $Q_j$
  - Intersection requirement: $i \in I_j \Rightarrow P_i \cap Q_j \neq \emptyset$.
  - **MINIMIZE:** total cost of push-updates, queries and responses:
    \[
    \sum_{i \in \mathcal{P}} p_i \cdot \text{SetC}(i, P_i) + \sum_{j \in \mathcal{Q}} q_j \cdot \text{SetC}(j, Q_j) + \sum_{j \in \mathcal{Q}} q_j \cdot \text{RespC}(j)
    \]
Routing Cost Models

**Multicast**

<table>
<thead>
<tr>
<th>Cost</th>
<th>Example</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast</td>
<td>3</td>
<td>Steiner tree cost</td>
</tr>
<tr>
<td>Unicast</td>
<td>4</td>
<td>Sum of path costs (non-)metric Distance function</td>
</tr>
<tr>
<td>Broadcast Model</td>
<td>5</td>
<td>Breadth first tree cost to depth $r$</td>
</tr>
</tbody>
</table>
Routing Cost Models

Unicast

<table>
<thead>
<tr>
<th>Cost</th>
<th>Example</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast</td>
<td>3</td>
<td>Steiner tree cost</td>
</tr>
<tr>
<td>Unicast</td>
<td>4</td>
<td>Sum of path costs (non-)metric Distance function</td>
</tr>
<tr>
<td>Broadcast Model</td>
<td>5</td>
<td>Breadth first tree cost to depth $r$</td>
</tr>
</tbody>
</table>
Routing Cost Models

Controlled Broadcast

<table>
<thead>
<tr>
<th>Cost</th>
<th>Example</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast</td>
<td>3</td>
<td>Steiner tree cost</td>
</tr>
<tr>
<td>Unicast</td>
<td>4</td>
<td>Sum of path costs (non-)metric Distance function</td>
</tr>
<tr>
<td>Broadcast Model</td>
<td>5</td>
<td>Breadth first tree cost to depth $r$</td>
</tr>
</tbody>
</table>
Related Work

- FeedTree: RSS via P2P Multicast, [Sandler et al., IPTPS’05]
- Web Caching applications
- Combs, Needles and Haystacks Paper, [Liu et al. SENSYS’04]
- Data Gerrymandering, [Bagchi et al. T.A. TKDE]
- Minimum Cost 2-spanners: [Dodis & Khanna STOC’99] and [Kortsarz & Peleg SICOMP’98]
- Multicommodity facility location, [Ravi & Sinha SODA’04]
- Classical Theory Problems
  - Facility Location
  - Steiner Tree (including Group Steiner Tree)
Our Results

- Multicast Model
  - Exact Tree Algorithm (Distributed)
  - General Graphs
    - $O(\log n)$-Approximation
    - NP-Completeness
Our Results

• Multicast Model
  * Exact Tree Algorithm (Distributed)
  * General Graphs
    * \(O(\log n)\)-Approximation
    * NP-Completeness

• Unicast Model
  * Nonmetric Case — \(O(\log n)\)-Approximation
  * Identical Interest Sets / Metric Case — \(O(1)\)-Approximation
  * NP-Completeness
Our Results

- Multicast Model
  - Exact Tree Algorithm (Distributed)
  - General Graphs
    - $O(\log n)$-Approximation
    - NP-Completeness

- Unicast Model
  - Nonmetric Case — $O(\log n)$-Approximation
  - Identical Interest Sets / Metric Case — $O(1)$-Approximation
  - NP-Completeness

- Controlled Broadcast Model
  - A Polynomial LP solution
  - A Combinatorial solution
The Multicast Model – With Aggregation

- want the following
  - A push subtree $T_i$ for each source $i$
  - A pull subtree $T'_j$ for each sink $j$
  - Whenever $j$ is interested in $i$ ($i \in I_j$), $T_i \cap T'_j \neq \emptyset$.
  - Total cost of all trees (summing edge weights in each tree) is minimized.

- For Trees:
  - Basic idea: for each edge, compute minimum possible cost for connectedness of trees.
  - **Claim**: Global optimum consists of this solution at every edge.
• Interest sets: \( \{x, z\} \) want \( \{a, b, c\} \); \( y \) wants only \( a \).
Interest sets: \( \{x, z\} \) want \( \{a, b, c\} \); \( y \) wants only \( a \).
• Interest sets: \{x, z\} want \{a, b, c\}; y wants only a.

• Question: What is the **minimum** we can pay on edge vw?
• Interest sets: \(\{x, z\}\) want \(\{a, b, c\}\); \(y\) wants only \(a\).

• Question: What is the \textbf{minimum} we can pay on edge \(vw\)?
- Long chain, no sources or sinks.
• Long chain, no sources or sinks.

• Identical Bipartite graph problem
Long chain, no sources or sinks.

Identical Bipartite graph problem

Suppose many possible MWVCs (eg $a + b + c = a + z = y + z$).

How to break MWVC ties?
• Long chain, no sources or sinks.
• Identical Bipartite graph problem
• Suppose many possible MWVCs (eg $a + b + c = a + z = y + z$).
• How to break MWVC ties?

**Defn:** In bipartite $G = (A \cup B, E)$, an MWVC is $A$-maximum if it has maximum weight in $A$. 
• Interest sets — recall: \( \{x, z\} \) want \( \{a, b, c\} \); \( y \) wants only \( a \).
• Interest sets — recall: \( \{x, z\} \) want \( \{a, b, c\} \); \( y \) wants only \( a \).
• What about \( G_{uv} \)? Clearly different.
• Interest sets — recall: \{x, z\} want \{a, b, c\}; y wants only a.

• What about \(G_{uv}\)? Clearly different.
• Interest sets — recall: \( \{x, z\} \) want \( \{a, b, c\} \); \( y \) wants only \( a \).
• What about \( G_{uv} \)? Clearly different.
• Interest sets — recall: \{x, z\} want \{a, b, c\}; y wants only \(a\).

• What about \(G_{uv}\)? Clearly different.

• Are push trees, pull trees and response paths connected?

**Lemma 1.** If we compute push-maximum MWVC for every edge, then Push and Pull subtrees are connected.
Tree Algorithm

for each directed edge \( uv \)

\[
\text{construct the graph } G_{uv} \\
\text{find its canonical minimum cut } C_{uv}
\]

for all \( i \in P_{uv} \)

\[
\text{if } i \in C_{uv} \text{ then include } uv \text{ in } T_i
\]

for all \( j \in Q_{vu} \)

\[
\text{if } j \in C_{uv} \text{ then include } uv \text{ in } T'_j
\]

for all \( (i, j) \in X_{uv} \)

\[
\text{if } x_{ij} \in C_{uv} \text{ then include } uv \text{ in } P(T_i, j)
\]
Distributed Implementation

- Global **All-to-all** exchange of
  - sets of push nodes’ frequencies,
  - pull nodes’ frequencies and interest sets.

- Locally, each edge solves both its directions *independently*.

- Use the solution to push and pull information

**Notes:**

- Cost of first phase small compared to third.

- For small sets of distinct values, communication improved.
Multicast Model – General Graph Approximation algorithm

- Reduction from Min Steiner Tree; NP-hard to approximate within 96/95. Chlebík & Chlebíková SWAT’02

Theorem 1. There is an expected $O(\log n)$-approximation for the Multicast problem in general graphs.
Multicast Model – General Graph Approximation

- Reduction from Min Steiner Tree; NP-hard to approximate within 96/95. Chlebík & Chlebíková SWAT’02

**Theorem 1.** There is an expected $O(\log n)$-approximation for the Multicast problem in general graphs.

We use the following:

**Theorem 2 (Fakcharoenphol et al. STOC’03).** The distribution over tree metrics resulting from (their) algorithm $O(\log n)$-probabilistically approximates the metric $d$. 
The Unicast Model
The Unicast Model

- Given (non-)metric distances $d_{uv}$ for every pair $(u, v) \in V \times V$.
- $\text{SetC}(u, S) = \sum_{k \in S} d_{uk}$
- find push-sets $P_i$ and pull-sets $Q_j$ that minimize total communication cost:
  $$\sum_{i \in P} p_i \sum_{k \in P_i} d_{ik} + \sum_{j \in Q} q_j \sum_{k \in Q_j} d_{kj} + \sum_{j \in Q} q_j \cdot \text{RespC}(j),$$
- and satisfies: for all $i \in I_j$, $P_i \cap Q_j \neq \emptyset$
- where
  $$\text{RespC}(j) = \begin{cases} 
  \text{SetC}(j, Q_j) & \text{(aggregation model)} \\
  \sum_{i \in I_j} \text{MinC}(P_i \cap Q_j, j) & \text{otherwise.}
\end{cases}$$
Unicast Model with Aggregation
An Integer Program

- Replace response cost by doubling sink frequencies
- \( x_{ik} = 1 \) means \( i \) pushes to \( k \)
- \( y_{kj} = 1 \) means \( j \) pulls from \( k \)
- \( r_{ijk} = 1 \) means \( i \) talks to \( j \) through \( k \).

Minimize:

\[
\sum_{i \in P} p_i \sum_{k \in V} d_{ik} x_{ik} + \sum_{j \in Q} q_j \sum_{k \in V} d_{kj} y_{kj}
\]

subject to

\[
\begin{align*}
    r_{ijk} & \leq x_{ik} \\
    r_{ijk} & \leq y_{kj} \\
    \sum_k r_{ijk} & \geq 1
\end{align*}
\]

where \( x_{ik}, y_{kj}, r_{ijk} \in \{0, 1\} \).
Unicast Model with Aggregation
Uniform Interests, Metric Case — $O(1)$-Approximation

**Overview**

* Applies for Identical/Disjoint Interest Sets
* Uses same Integer Program.
* Deterministic Rounding with Filtering Technique Lin & Vitter IPL’92, Shmoys et al STOC’97, Ravi & Sinha SODA’04
**Basic definitions**

* Optimal solution to the LP is \((x^*, y^*, r^*)\).
* LP gives cost lower bounds

\[
C_i = \sum_k d_{ik} x_{ik}^* \quad \text{and} \quad C_j' = \sum_k d_{kj} y_{kj}^*
\]

\[
C_j' = 6.5
\]
Basic definitions

* Optimal solution to the LP is \((x^*, y^*, r^*)\).
* LP gives cost lower bounds
  \[ C_i = \sum_k d_{ik} x_{ik}^* \quad \text{and} \quad C_j' = \sum_k d_{kj} y_{kj}^* \]
* For node \(u\), \(r > 0\), define \(B_u(r) = \{v : d_{uv} \leq r\}\).
* Let \(1 < \alpha < \beta\). Clearly
  \[ B_j(C_j') \subseteq B_j(\alpha C_j') \subseteq B_j(\beta C_j') \]
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

Choose leaders: nodes with disjoint \( \beta \)-balls, by nondecreasing cost.
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
- Define push sets: $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S' \text{ and } C'_j \leq C_i\}$
  and pull sets: $Q_j = \{j\} \cup \{\ell'_j\} \cup \{i : i \in S \text{ and } C_i < C'_j\}$. 
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint $\beta$-balls, by nondecreasing cost.
- Define push sets: $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S' \text{ and } C'_j \leq C_i\}$ and pull sets: $Q_j = \{j\} \cup \{\ell'_j\} \cup \{i : i \in S \text{ and } C_i < C'_j\}$.
- Intersection guarantee: For each $i \in \mathcal{P}$ and $j \in \mathcal{Q}$, $P_i \cap Q_j \neq \emptyset$. 
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm Proof

- Relative distance limits total push extent:
  For $i \in \mathcal{P}$, $\alpha > 1$, \[ \sum_{k \notin B_i(\alpha C_i)} x^*_ik \leq 1/\alpha \]
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm Proof

- Relative distance limits total push extent:
  For \( i \in \mathcal{P}, \alpha > 1 \),
  \[ \sum_{k \notin B_i(\alpha C_i)} x_{ik}^* \leq \frac{1}{\alpha} \]

- Derive Approximation Ratio.
  * Recall: \( P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S' \text{ and } C'_j \leq C_i \} \)
  * Cost to \( i \)'s leader \( \ell_i \): \( 2\beta C_i \)
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm Proof

- Relative distance limits total push extent:
  For $i \in \mathcal{P}$, $\alpha > 1$, \[ \sum_{k \notin B_i(\alpha C_i)} x_{ik}^* \leq 1/\alpha \]

- Derive Approximation Ratio.
  * Recall: $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S' \text{ and } C'_j \leq C_i\}$
  * Cost to $i$'s leader $\ell_i$: $2\beta C_i$
  * Cost to (other) leaders $S_i$:
    \[ C_i \geq \sum_{j \in S_i} (d_{ij} - \alpha C'_j) \sum_{k \in B_j(\alpha C'_j)} r_{ijk}^* \]
Unicast Model with Aggregation

Uniform Interest Set / Metric — Algorithm Proof

- Relative distance limits total push extent:
  For $i \in \mathcal{P}$, $\alpha > 1$, $\sum_{k \notin B_i(\alpha C_i)} x_{ik}^* \leq 1/\alpha$

- Derive Approximation Ratio.
  * Recall: $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S' \text{ and } C_j' \leq C_i\}$
  * Cost to $i$’s leader $\ell_i$: $2\beta C_i$
  * Cost to (other) leaders $S_i$:
    \[
    C_i \geq \sum_{j \in S_i} (d_{ij} - \alpha C_j') \sum_{k \in B_j(\alpha C_j')} r_{ijk}^*
    \]
    \[
    \geq \sum_{j \in S_i} d_{ij} \left[1 - \frac{\alpha}{\beta}\right]\left[1 - \frac{1}{\alpha}\right]
    \]
    \[
    = \frac{(\beta - \alpha)(\alpha - 1)}{\alpha \beta} \sum_{j \in S_i} d_{ij}.
    \]
  * $\alpha = 1.69$ and $\beta = 2.86$ obtains 14.57-approximation.
Conclusions and Open Problems

- **Multicast:**
  - General Graphs; Can $O(\log n)$ UB be improved to $O(1)$?

- **Nonmetric Unicast:**
  - Derandomizing $O(\log n)$ algorithm.
  - Close gap $O(1)$ LB vs $O(\log n)$ UB gap

- **Metric Unicast Case**
  - Improving the 14.57 bound for Uniform Interest sets.
  - Non-uniform interest sets (UB and/or Hardness)

- **Dynamic Graphs — Frequency, Position and Topology changes**
Thank You!