Playing Push vs Pull: Models and Algorithms for Disseminating Dynamic Data in Networks.

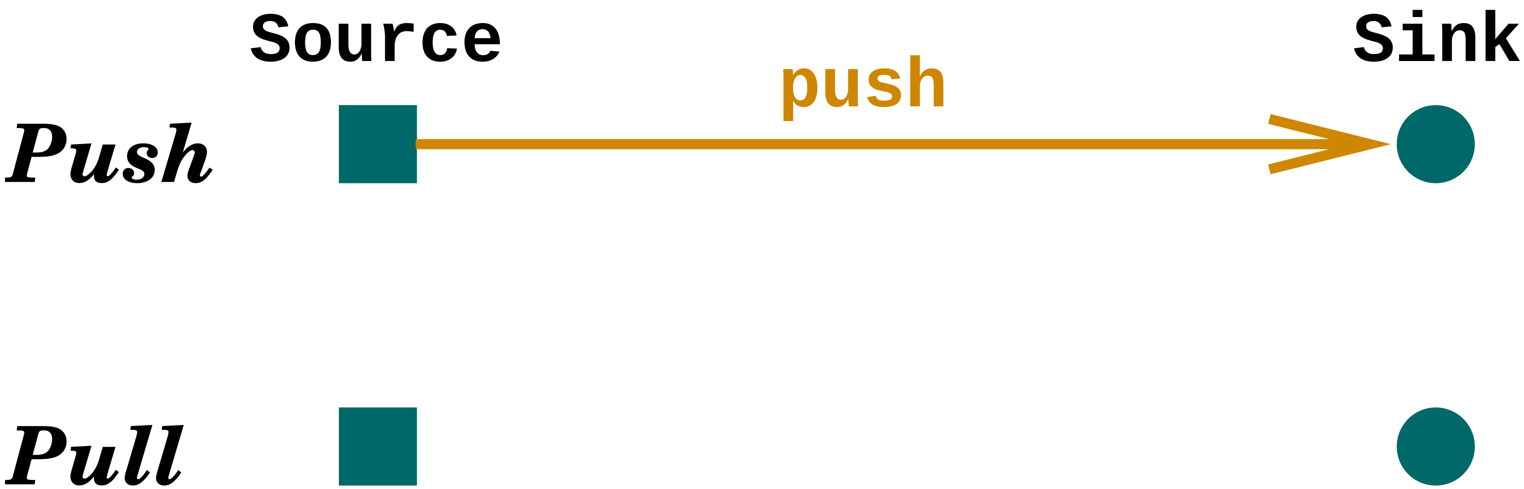
R.C. Chakinala, A. Kumarasubramanian, Kofi A. Laing
R. Manokaran, C. Pandu Rangan, **R. Rajaraman**

Push and Pull

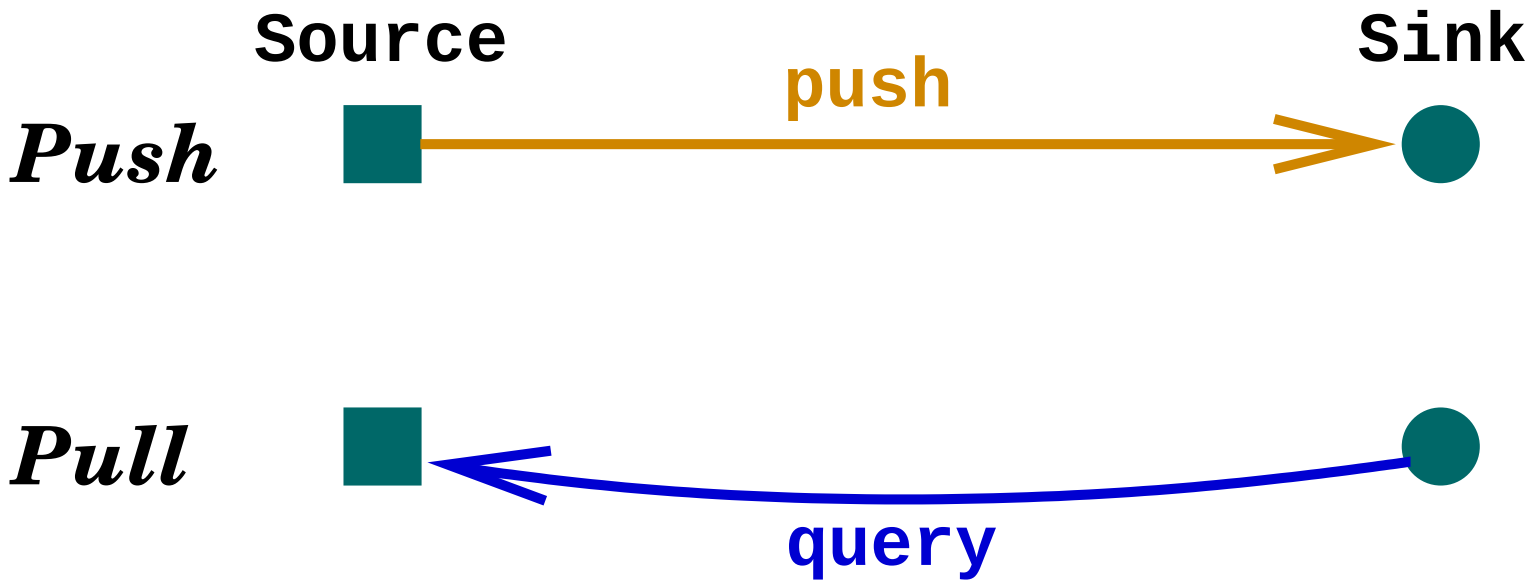
Push and Pull

Push and Pull

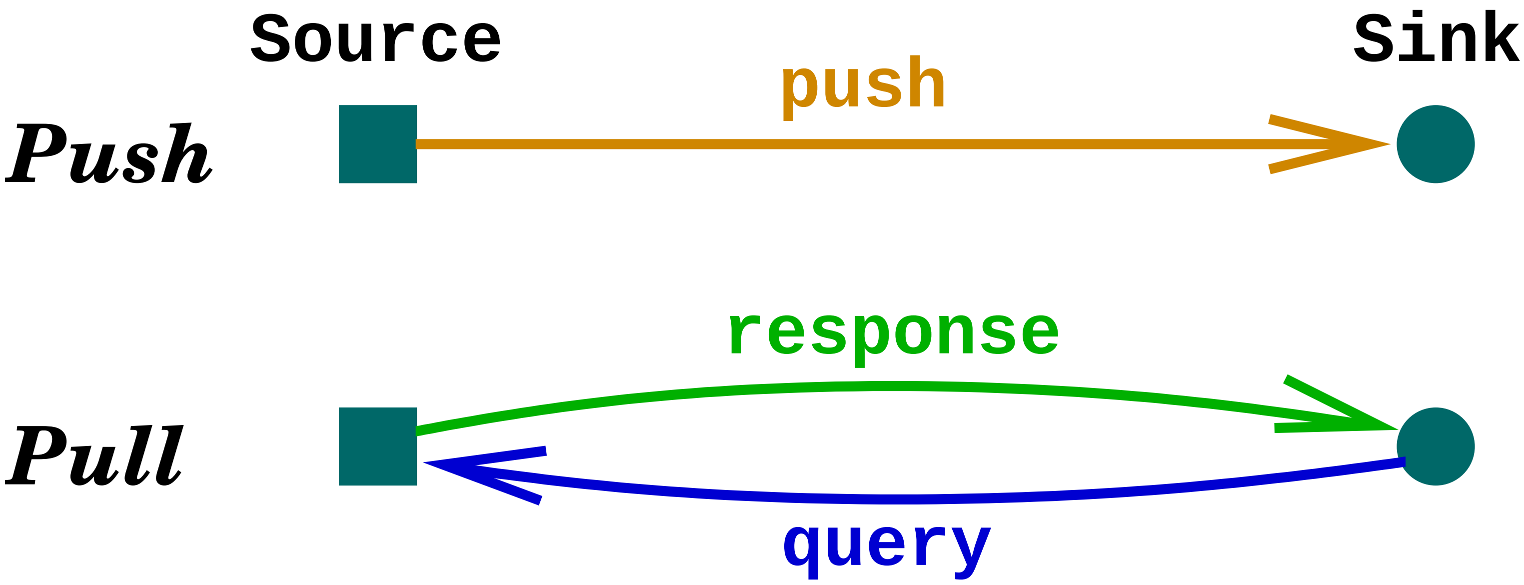
Push and Pull



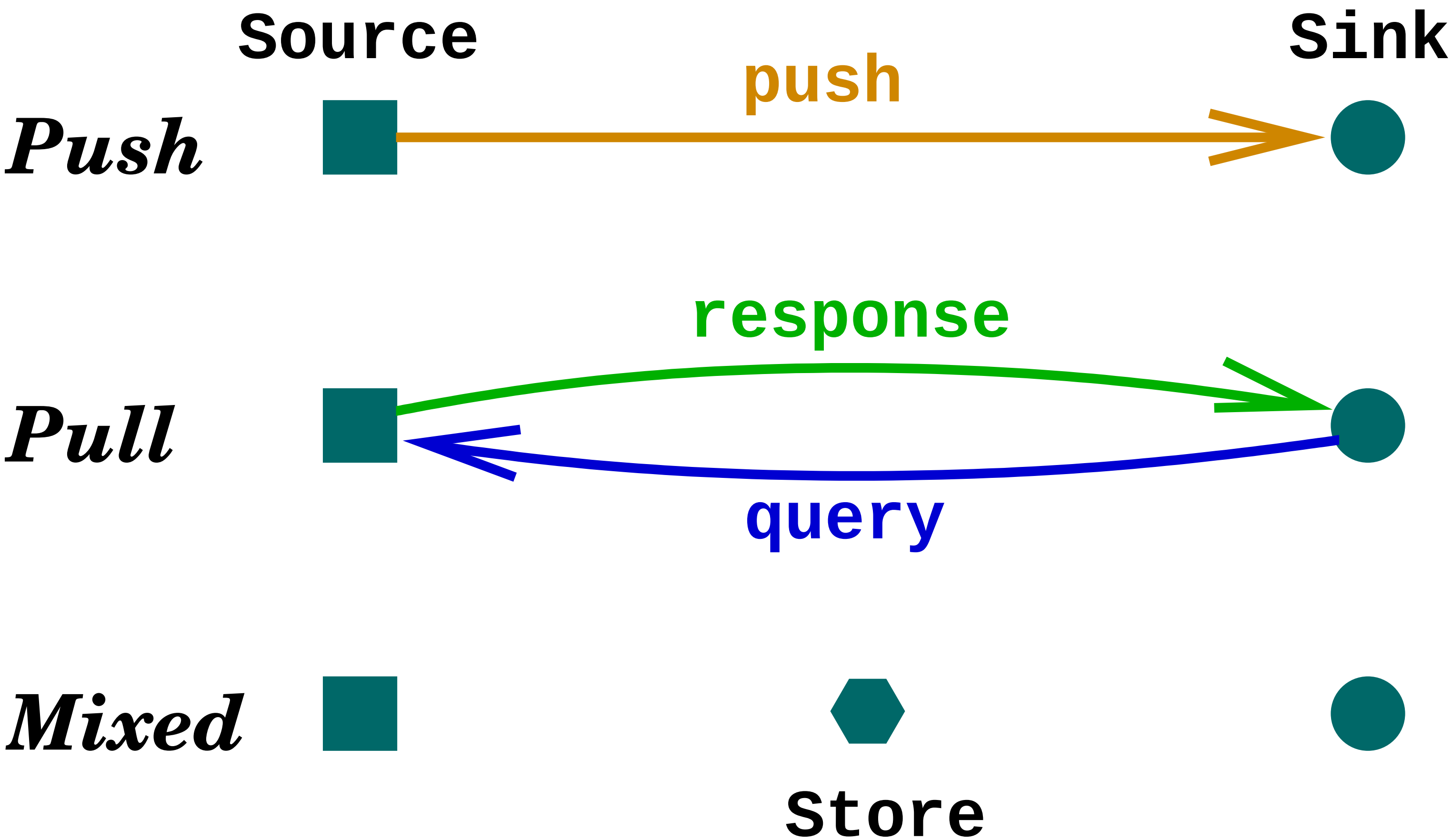
Push and Pull



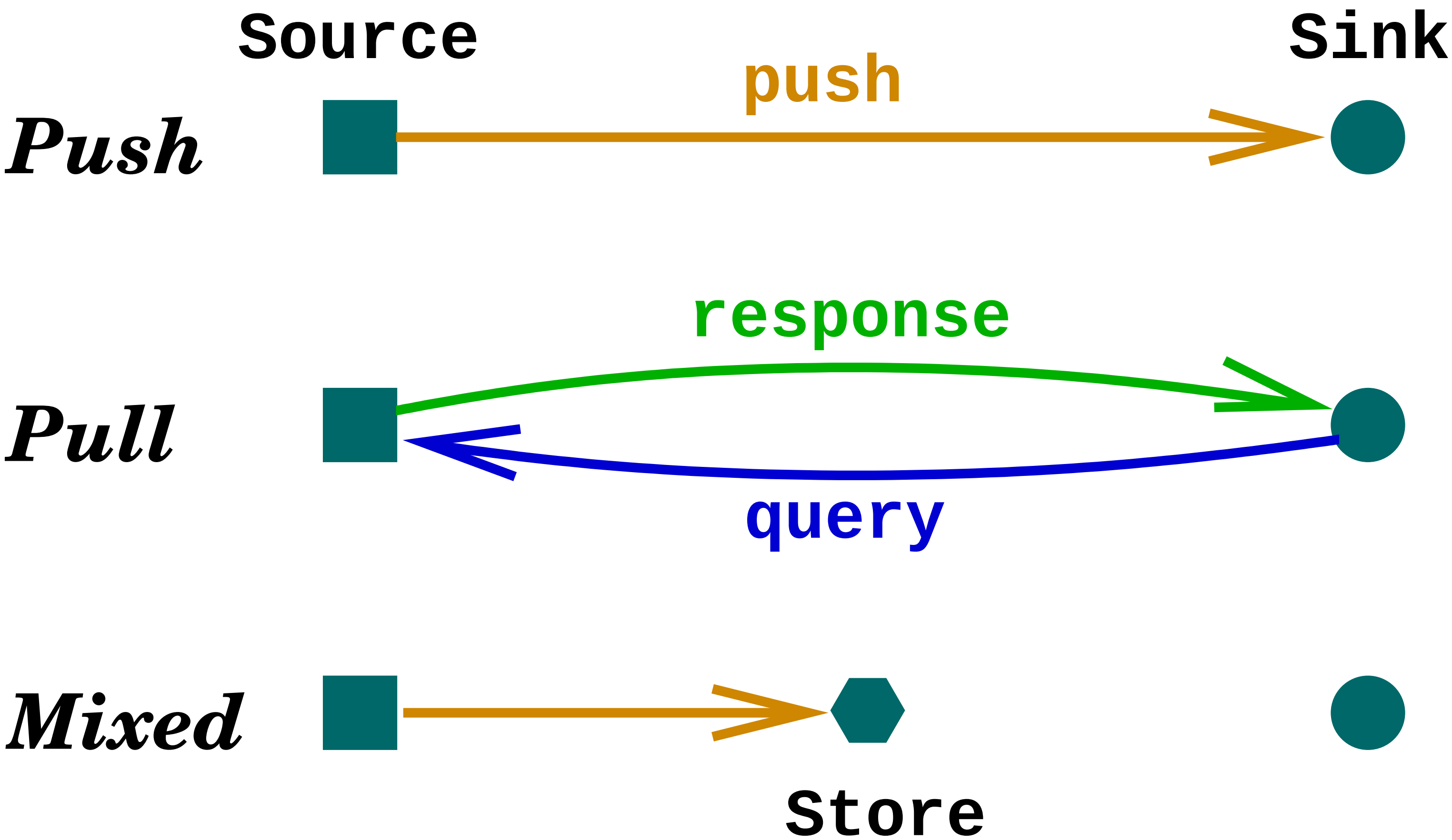
Push and Pull



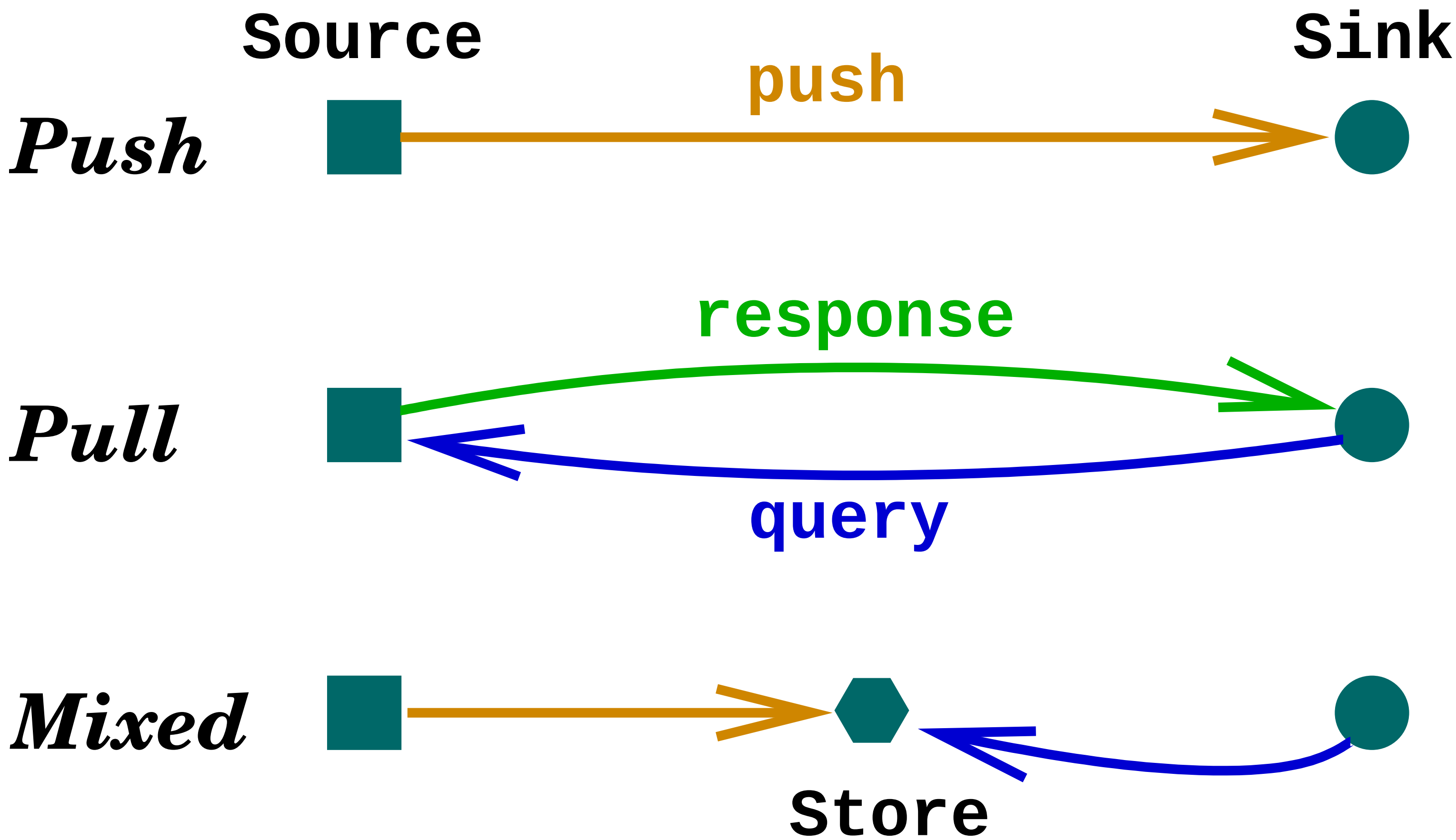
Push and Pull



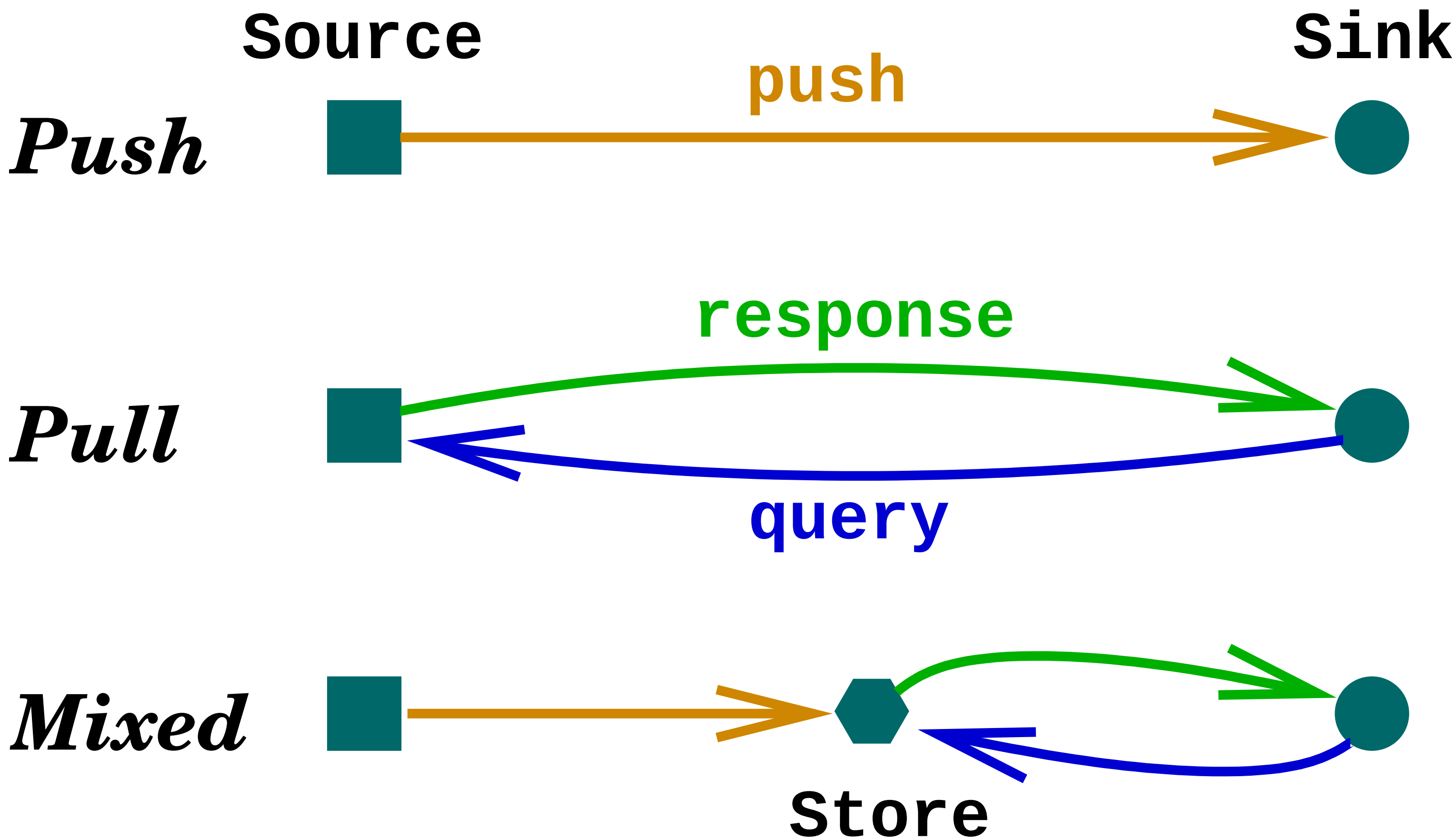
Push and Pull



Push and Pull

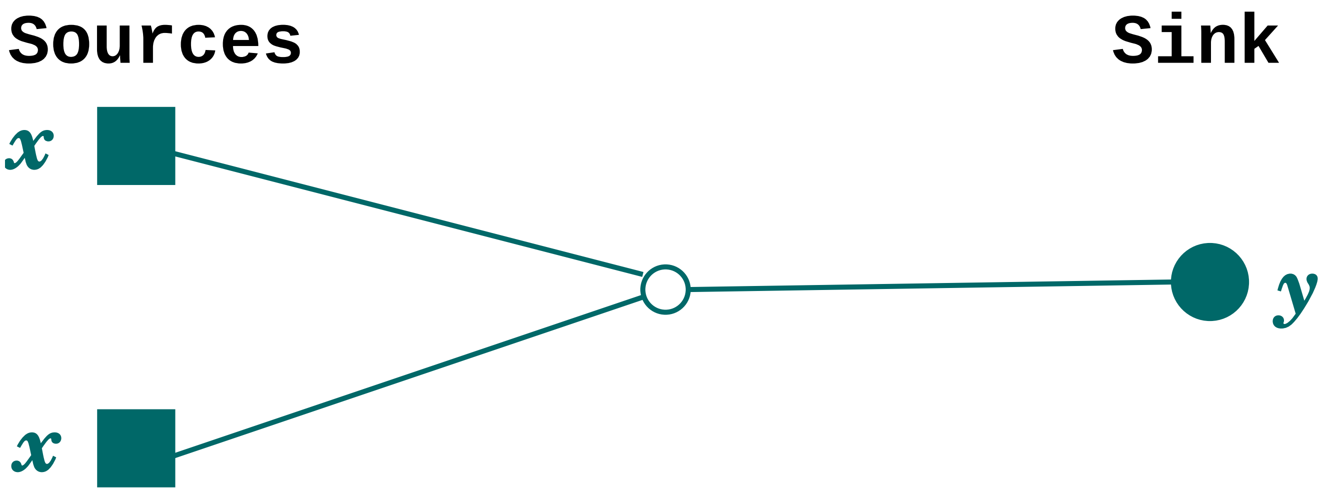


Push and Pull



A Simple Example

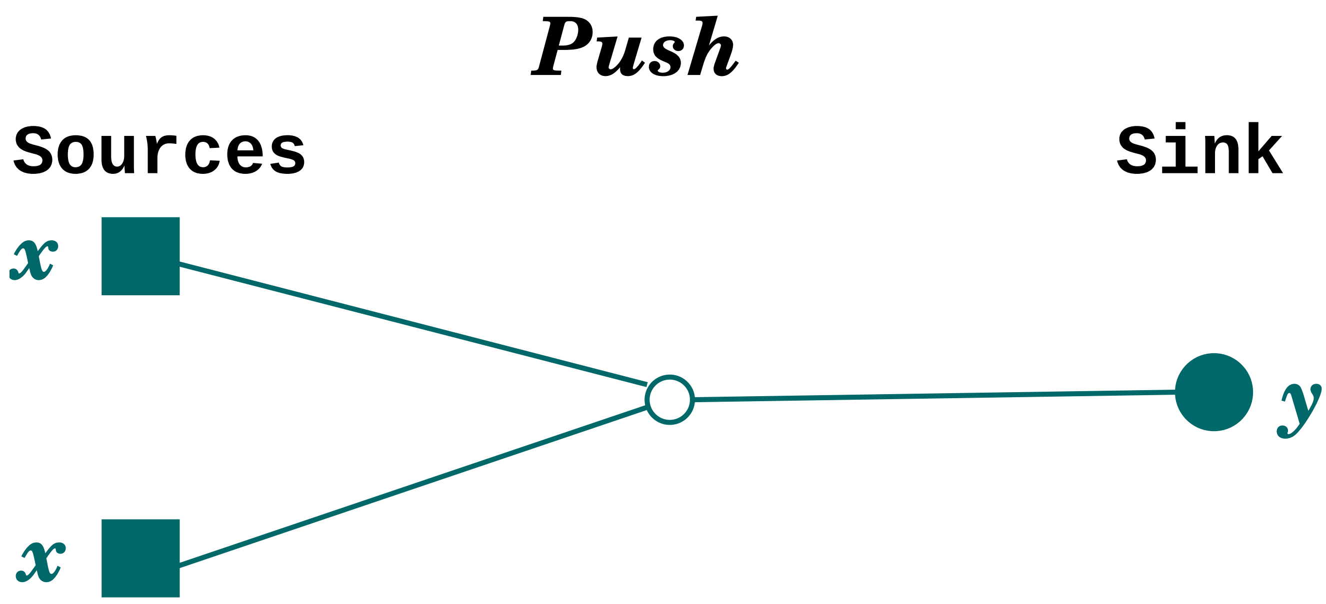
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

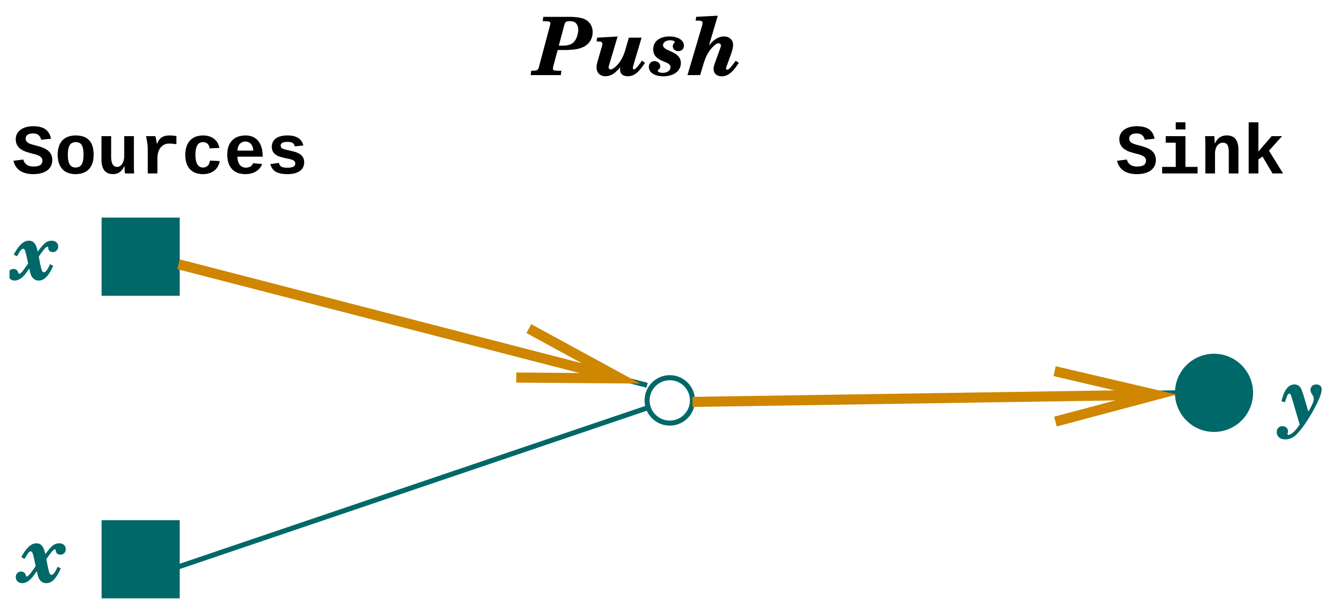
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

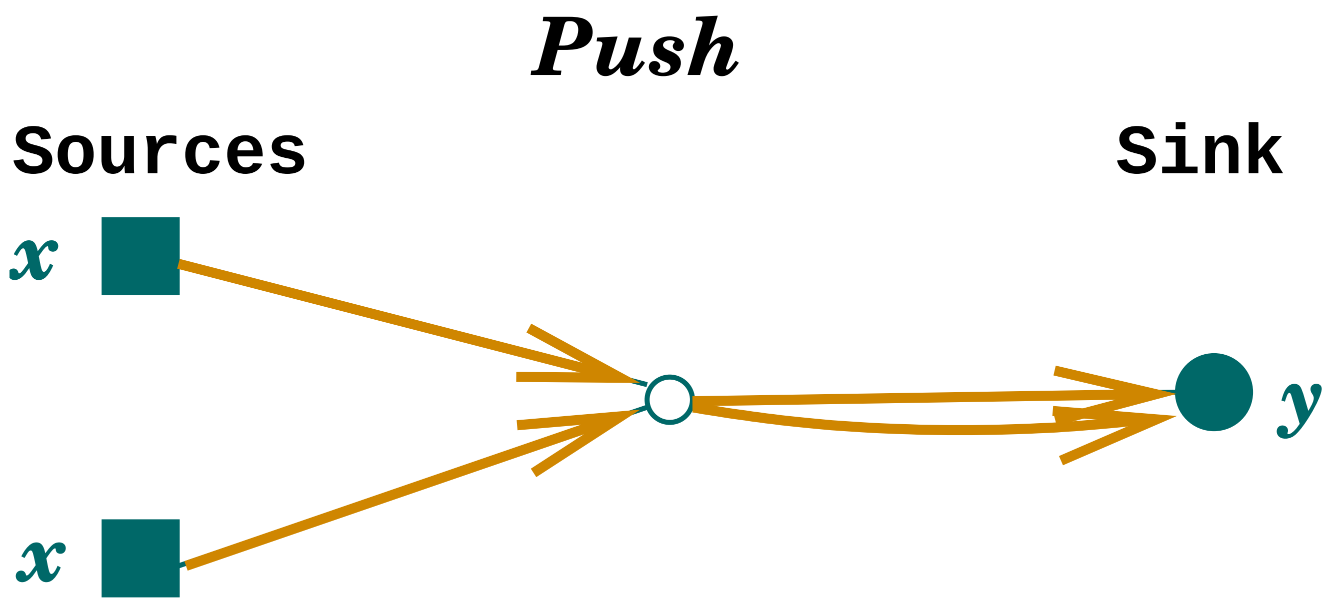
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

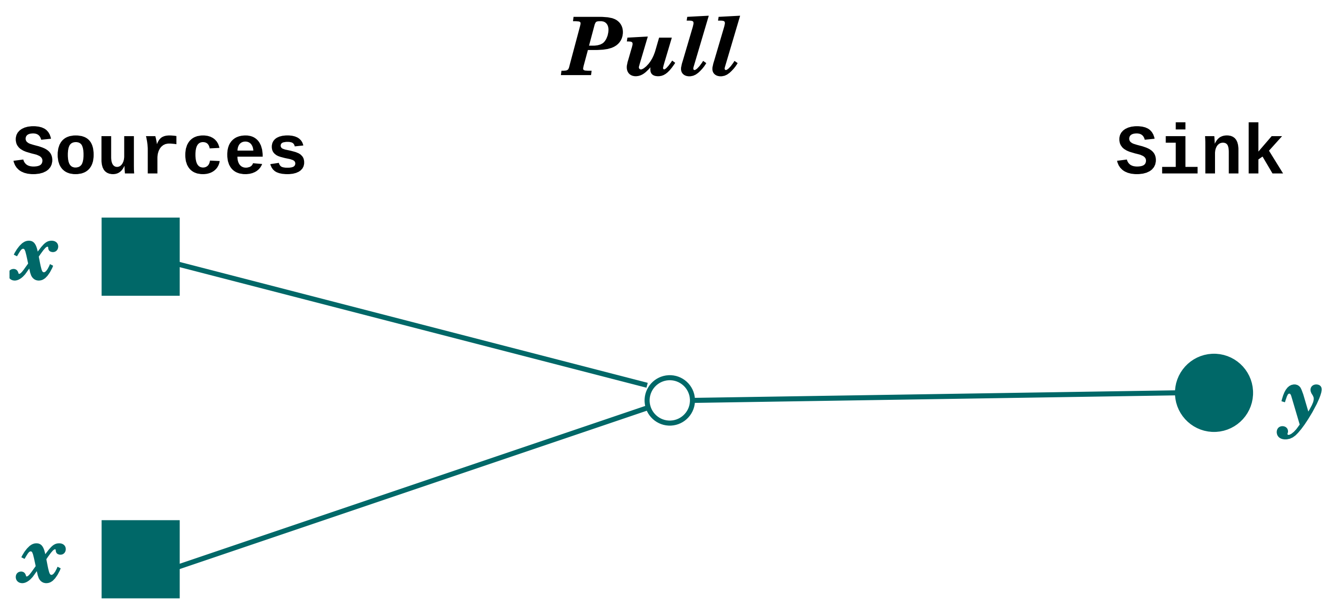
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

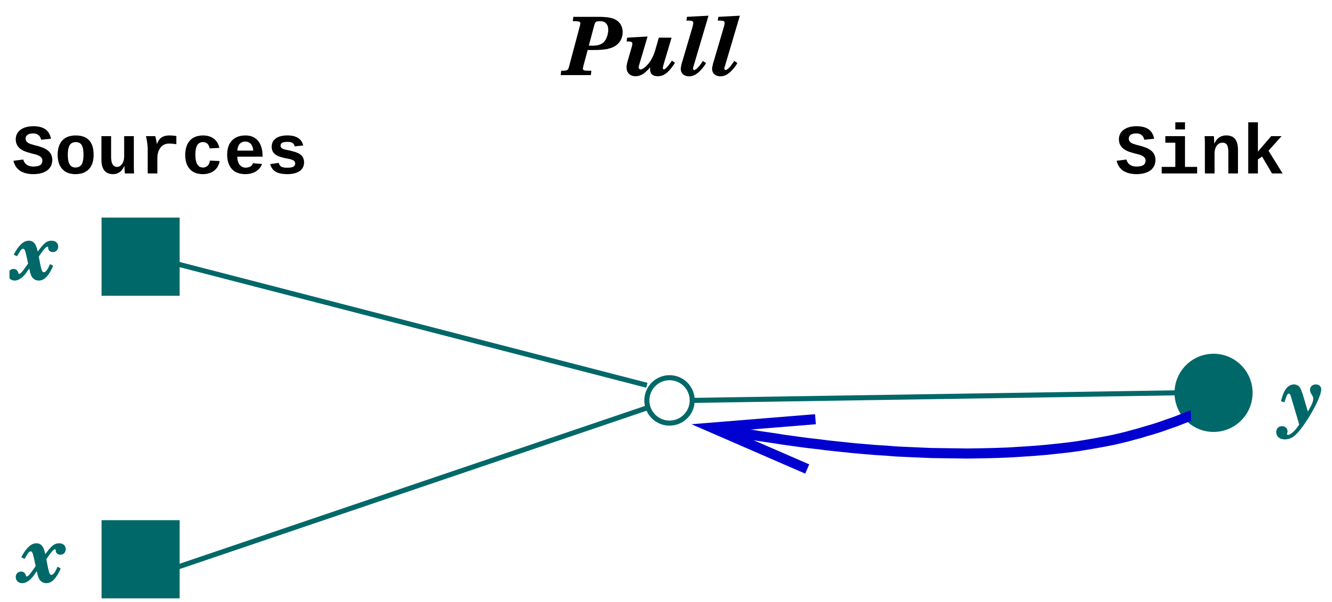
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

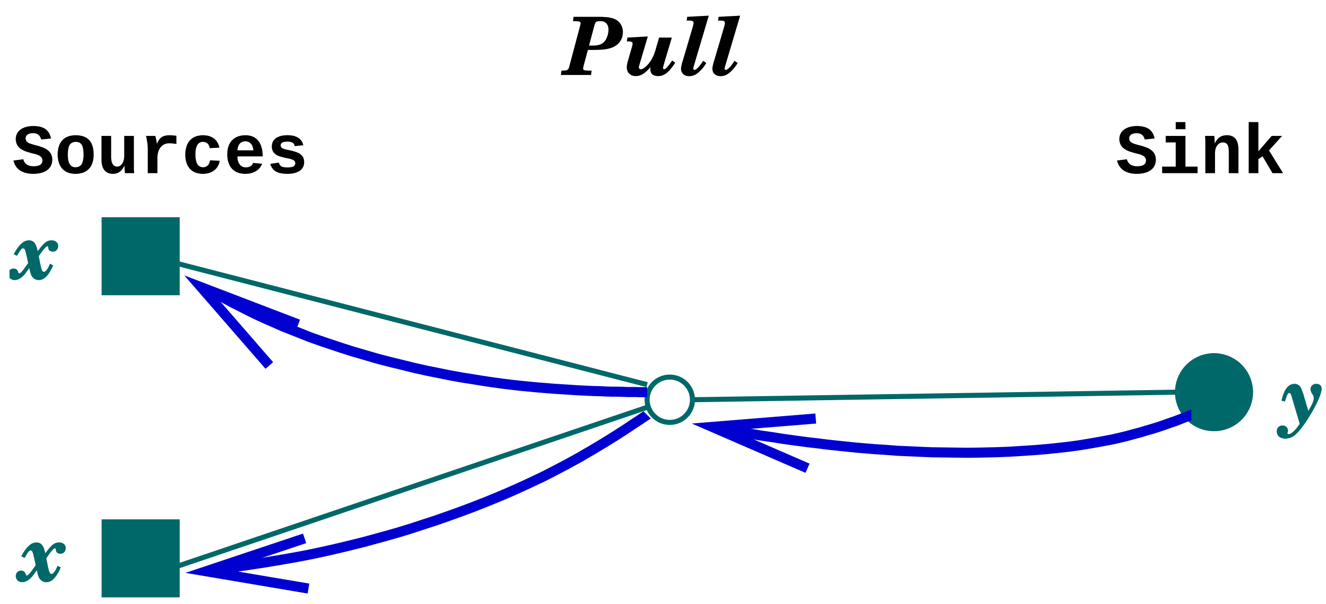
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

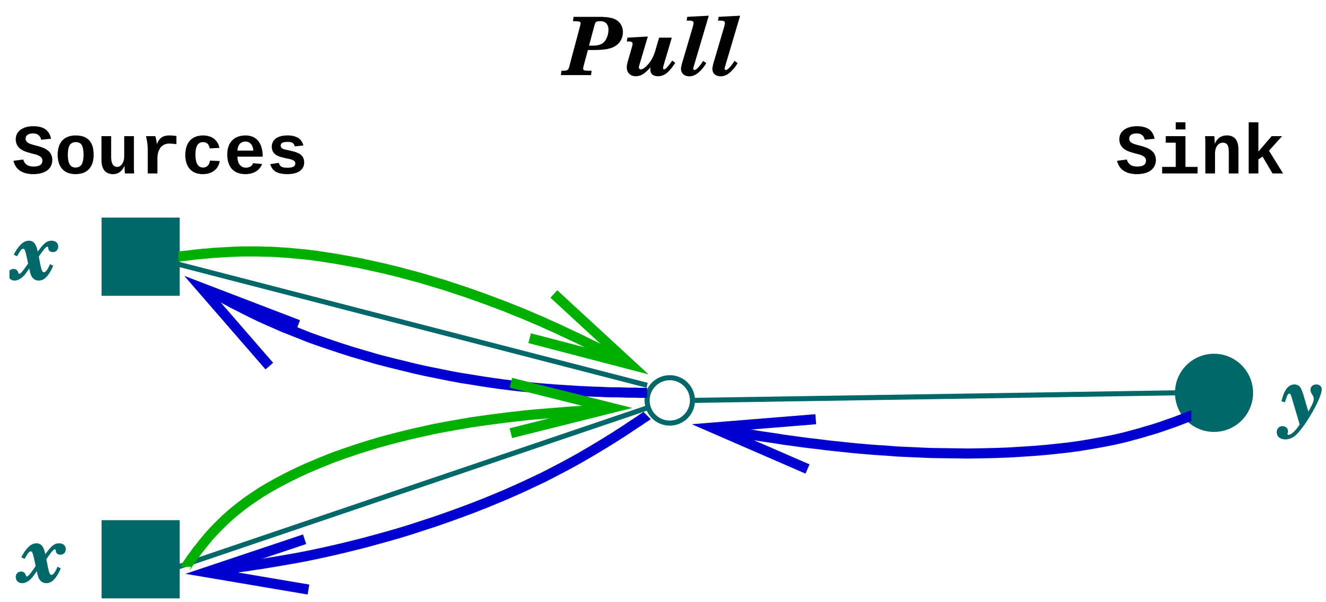
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

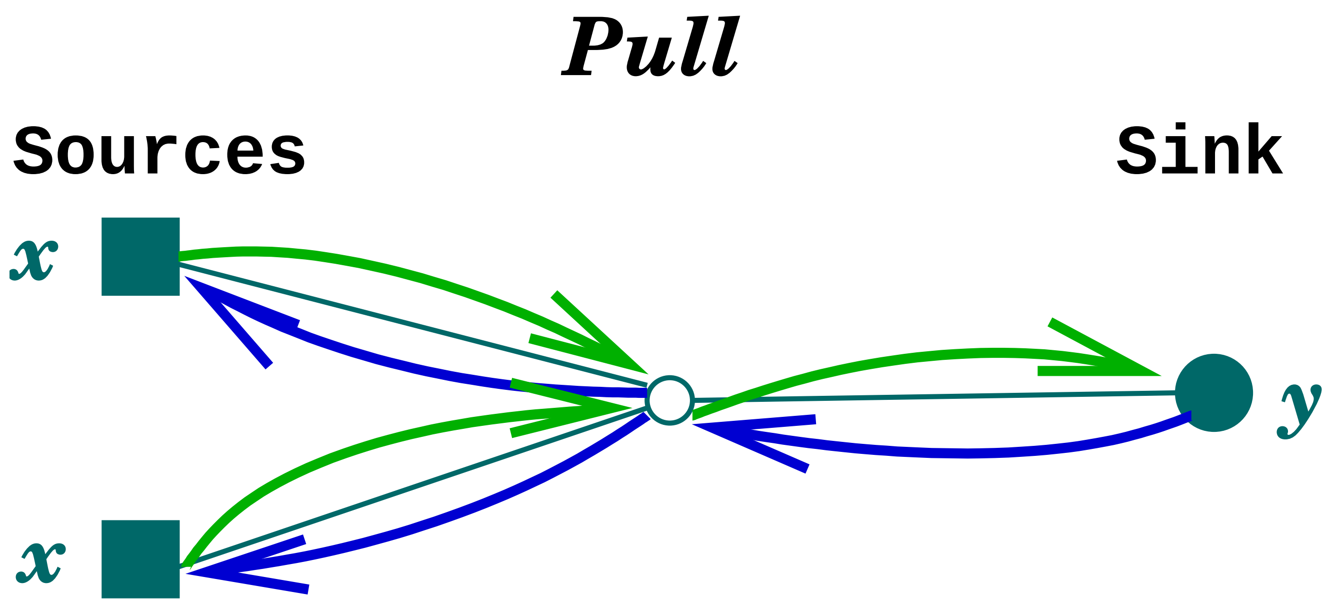
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

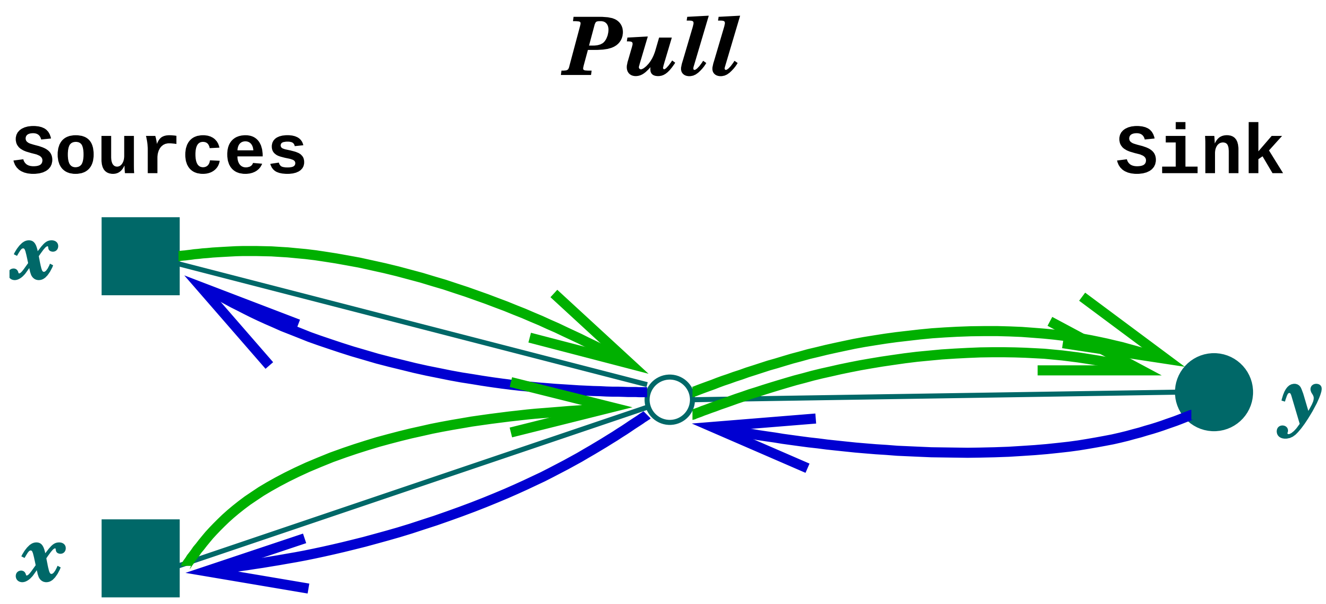
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

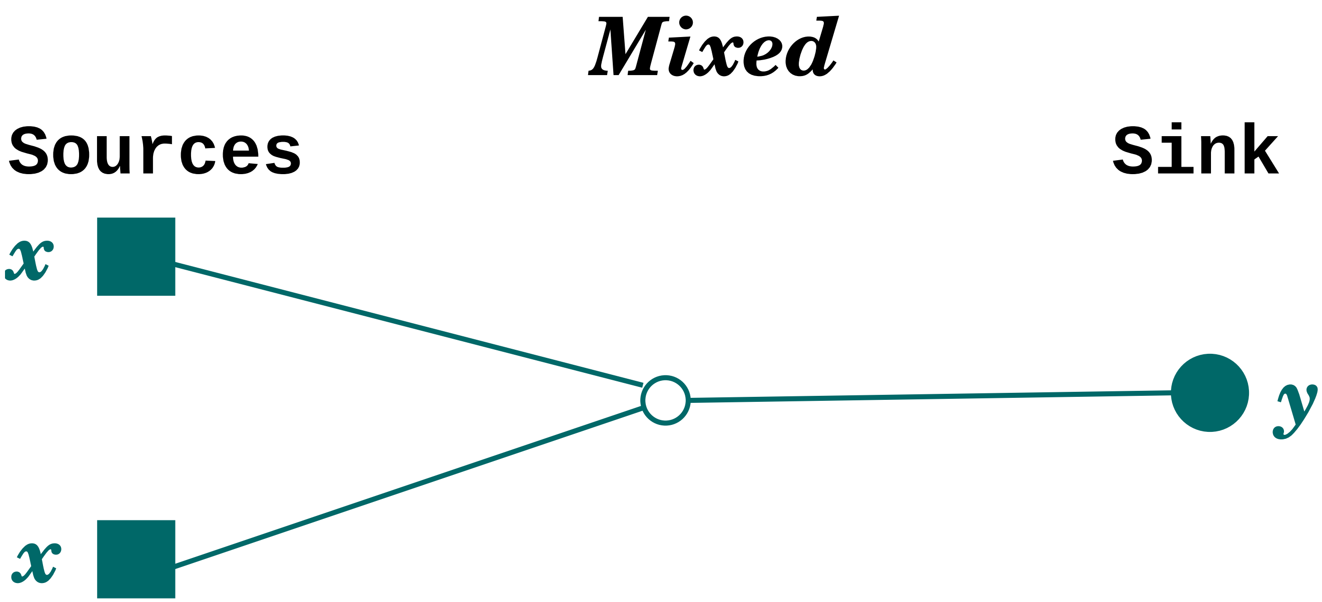
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

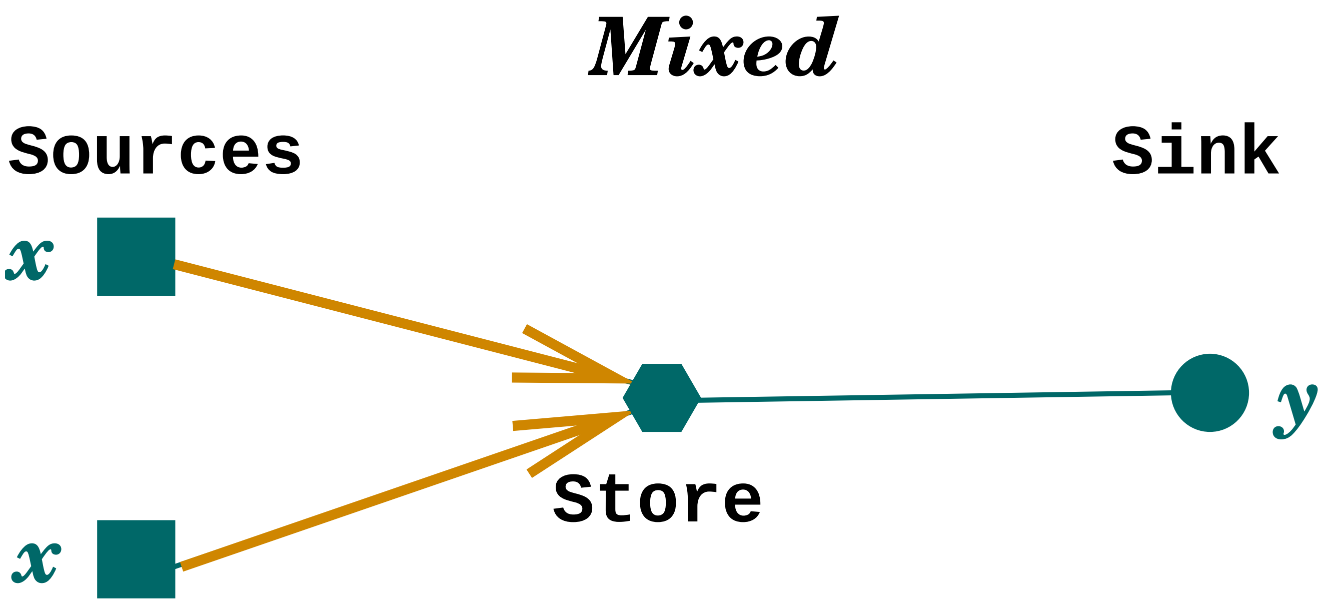
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

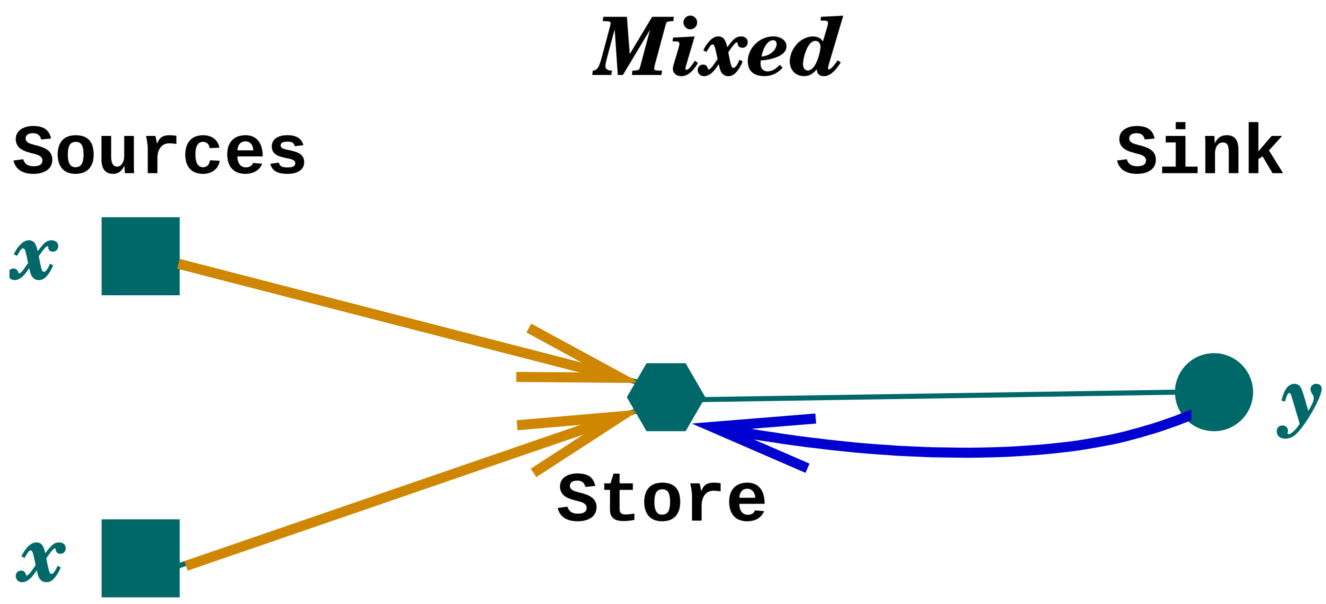
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

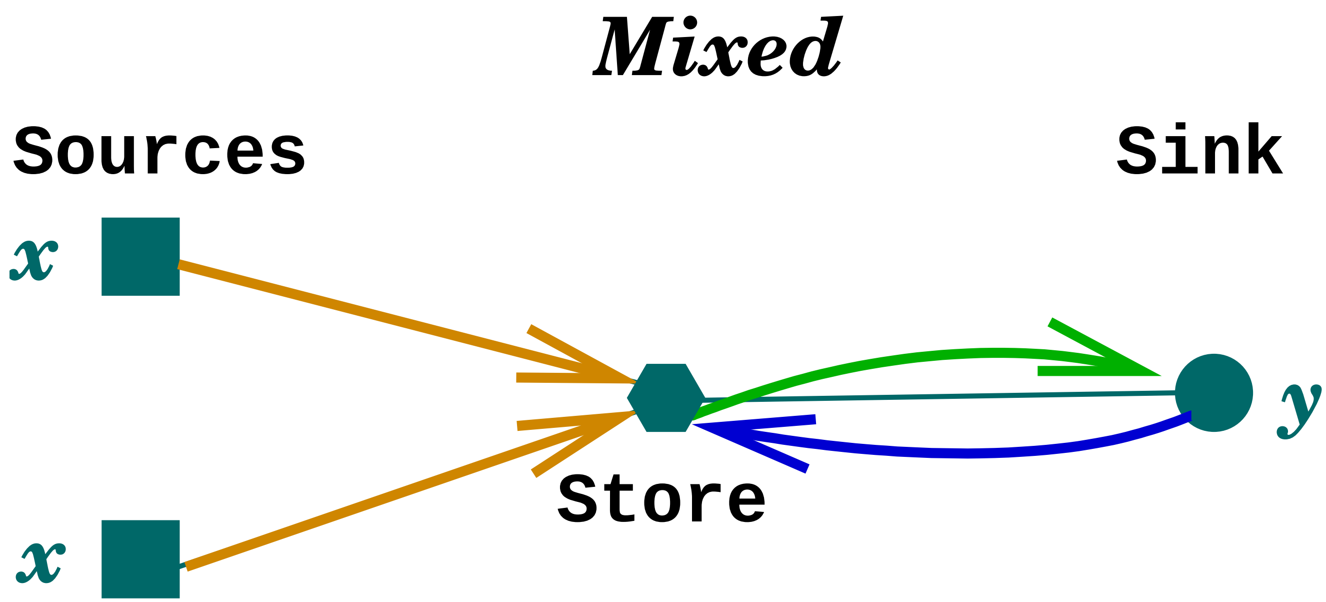
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x+2y$	$2x+3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

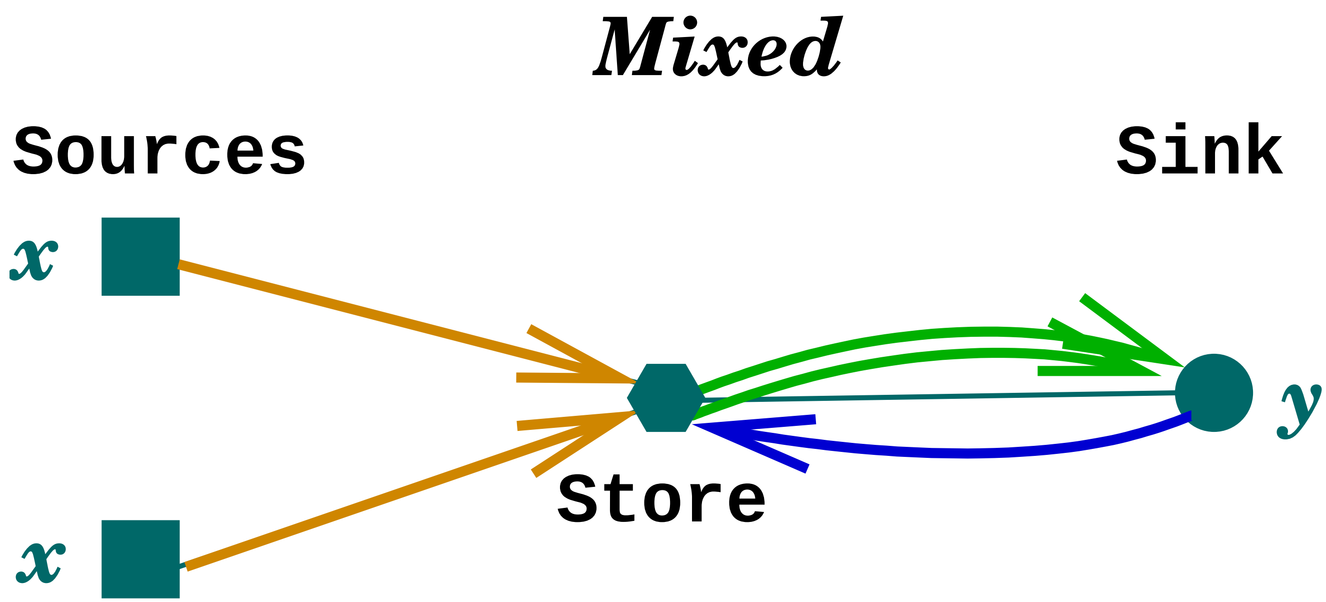
Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

A Simple Example

Using average source and sink frequencies.



	Aggregation	No Aggregation
Push	$4x$	$4x$
Pull	$6y$	$7y$
Mixed	$2x + 2y$	$2x + 3y$
Mixed is Best	$2y < 2x < 4y$	$3y < 2x < 4y$

General Problem — High Level

- **INPUTS:** Graph $G = (V, E)$ with:
  - ∗ cost of updating set of stores: $\mathsf{SetC} : V \times \mathsf{Powerset}(V) \longrightarrow \mathbb{R}^+$
  - ∗ **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

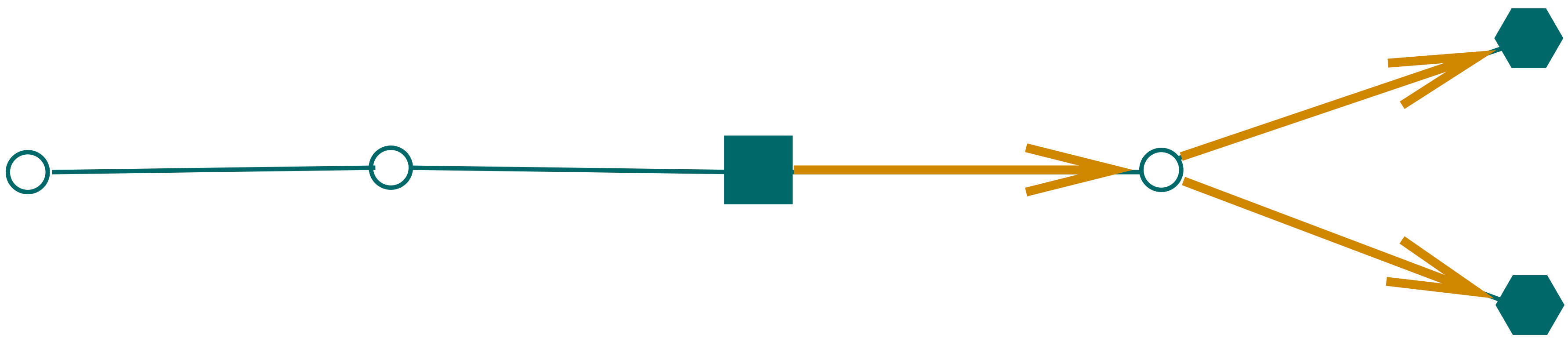
- **INPUTS:** Graph $G = (V, E)$ with:
  - $*$ cost of updating set of stores: $\text{SetC} : V \times \text{Powerset}(V) \longrightarrow \mathbb{R}^+$
  - $*$ **Source Set** $\mathcal{P} \subseteq V$, **Sink Set** $\mathcal{Q} \subseteq V$
  - $*$ For every source $i \in \mathcal{P}$, a **source frequency** 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: $i \in I_j \Rightarrow P_i \bigcap Q_j \neq \emptyset$.
  - $*$ 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

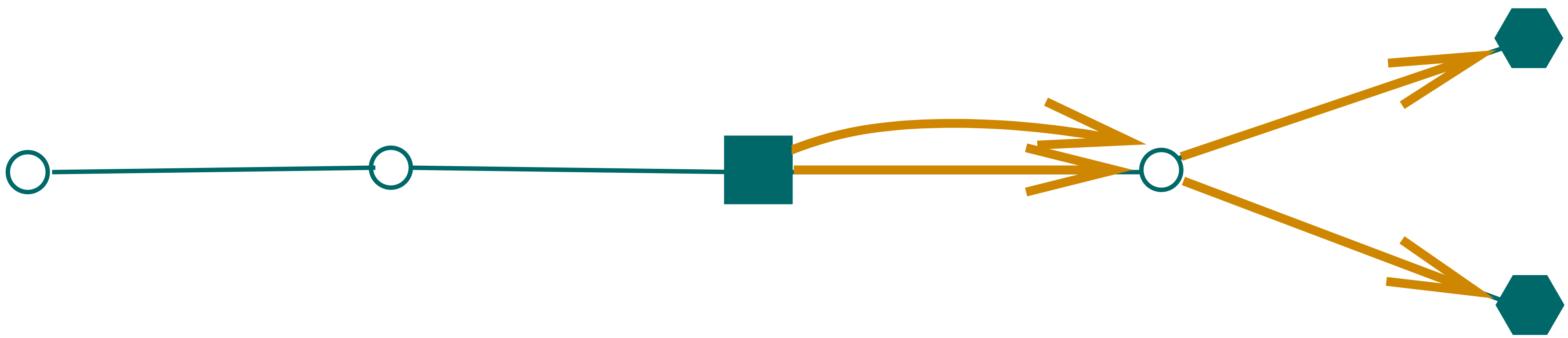
Multicast



Cost	Example	Definition
Multicast	3	Steiner tree cost
Unicast	4	Sum of path costs (non-)metric Distance function
Broadcast Model	5	Breadth first tree cost to depth r

Routing Cost Models

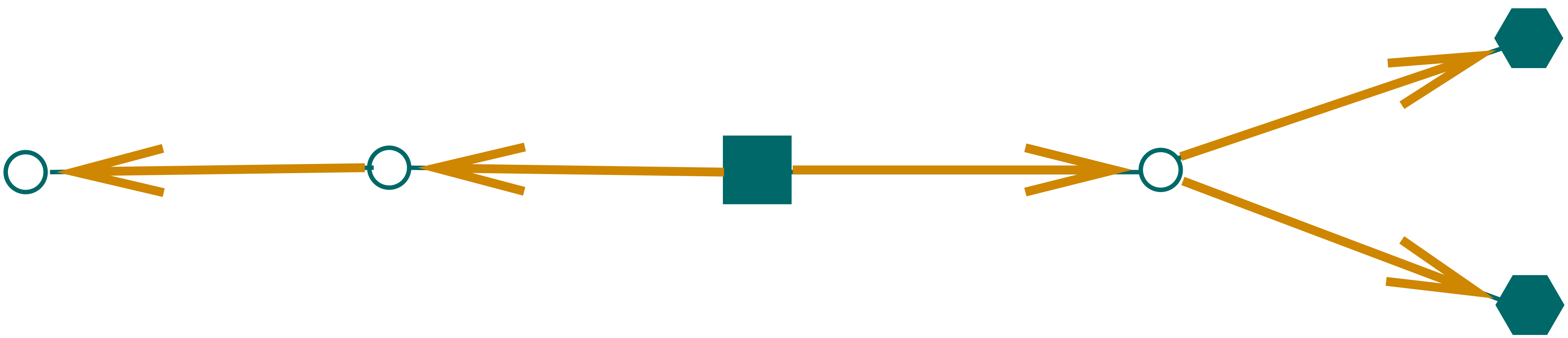
Unicast



Cost	Example	Definition
Multicast	3	Steiner tree cost
Unicast	4	Sum of path costs (non-)metric Distance function
Broadcast Model	5	Breadth first tree cost to depth r

Routing Cost Models

Controlled Broadcast



Cost	Example	Definition
Multicast	3	Steiner tree cost
Unicast	4	Sum of path costs (non-)metric Distance function
Broadcast Model	5	Breadth first tree cost to depth r

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.**

The Multicast Model

- Indicator variable x_{uvi} says whether $uv \in T_i$ (push tree i)
- y_{uvj} indicates $uv \in T'_j$ (pull tree j)
- z_{uvij} indicates $i \in I_j$ and $uv \in P(T_i \cap T'_j, j)$
- arbitrary m_{ij} is average response frequency
- Minimize Objective function

$$\sum_{i \in \mathcal{P}} p_i \sum_{uv \in E} c_{uv} x_{uvi} + \sum_{j \in \mathcal{Q}} q_j \sum_{uv \in E} c_{uv} y_{uvj} + \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{Q}} m_{ij} \sum_{uv \in E} c_{uv} z_{uvij}$$

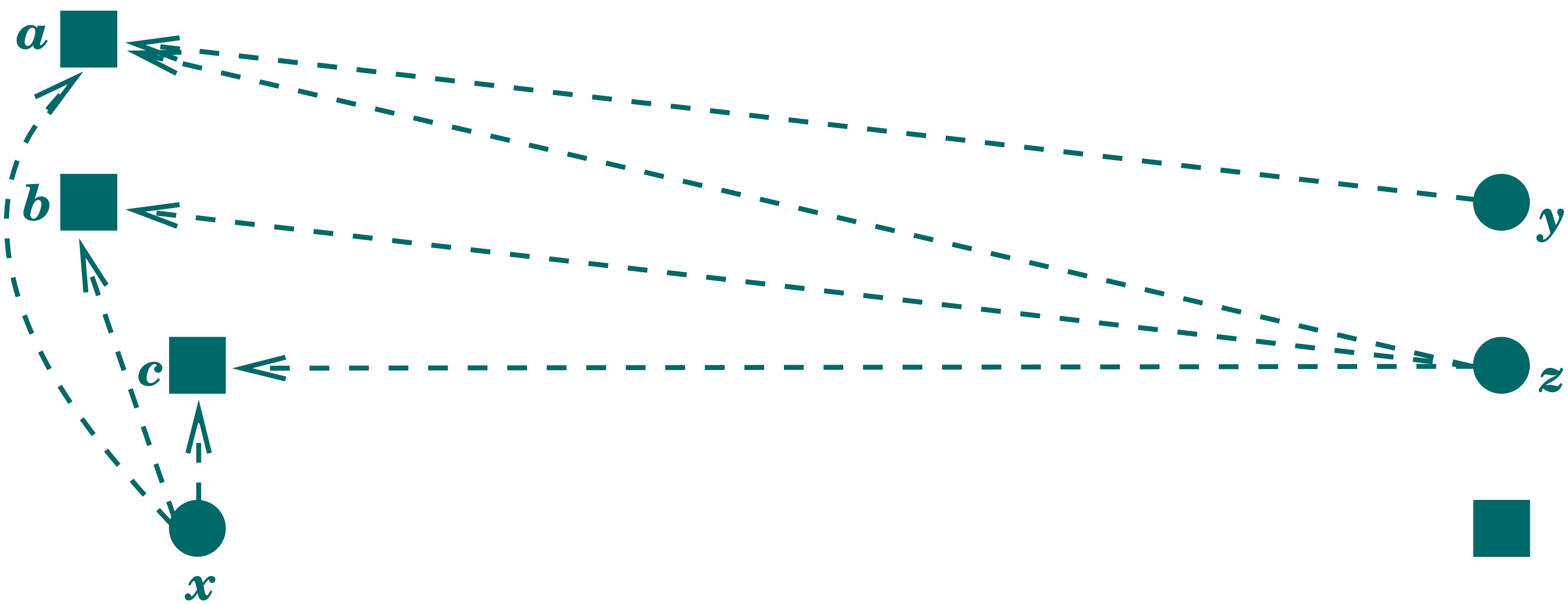
Multicast Model
An Exact (Distributed) Tree Algorithm

- G is a tree $T = (V, E)$
- MinC$(T_i \cap T'_j, j)$ is sum of edge weights on shortest path $P(T_i, j)$
- For edge uv, let S_{uv} be largest subtree containing u but not v
- Note $S_{vu} = V \setminus S_{uv}$
- Substituting $V = S_{uv} \cup S_{vu}$, we obtain two symmetric terms (eg.):

$$\sum_{uv \in E} c_{uv} \left[\sum_{i \in S_{uv}} p_i x_{uvi} + \sum_{j \in S_{vu}} q_j y_{uvj} + \sum_{i \in S_{uv}} \sum_{j \in S_{vu}} m_{ij} z_{uvij} \right]$$

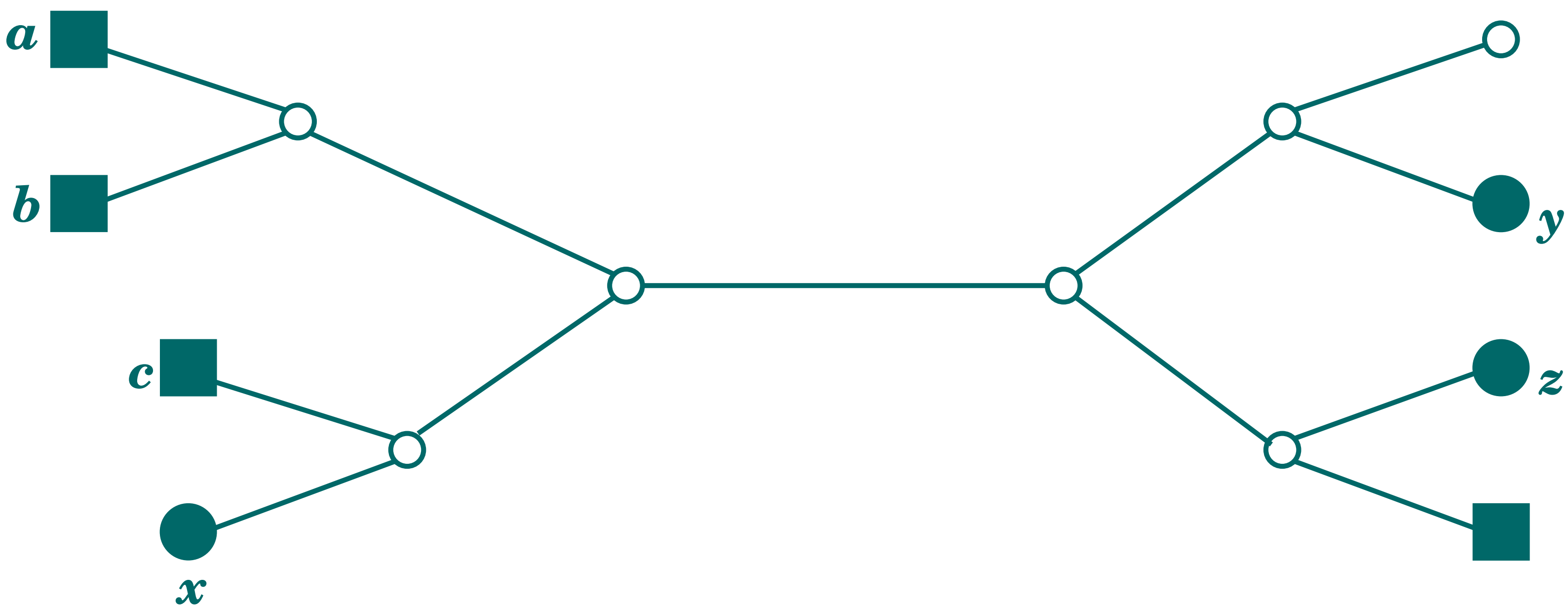
- **Claim: Global optimum minimizes $[\ldots]$ independently!**

Tree Algorithm Diagram



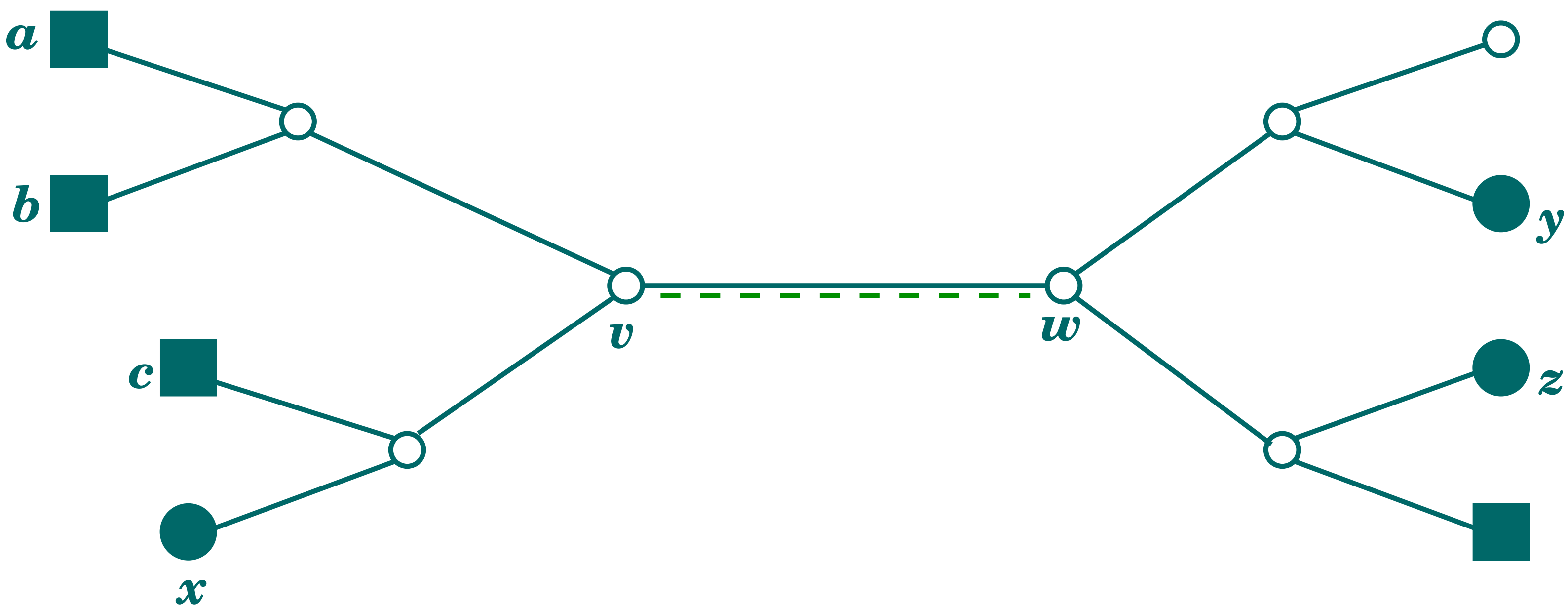
- Interest sets: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.

Tree Algorithm Diagram



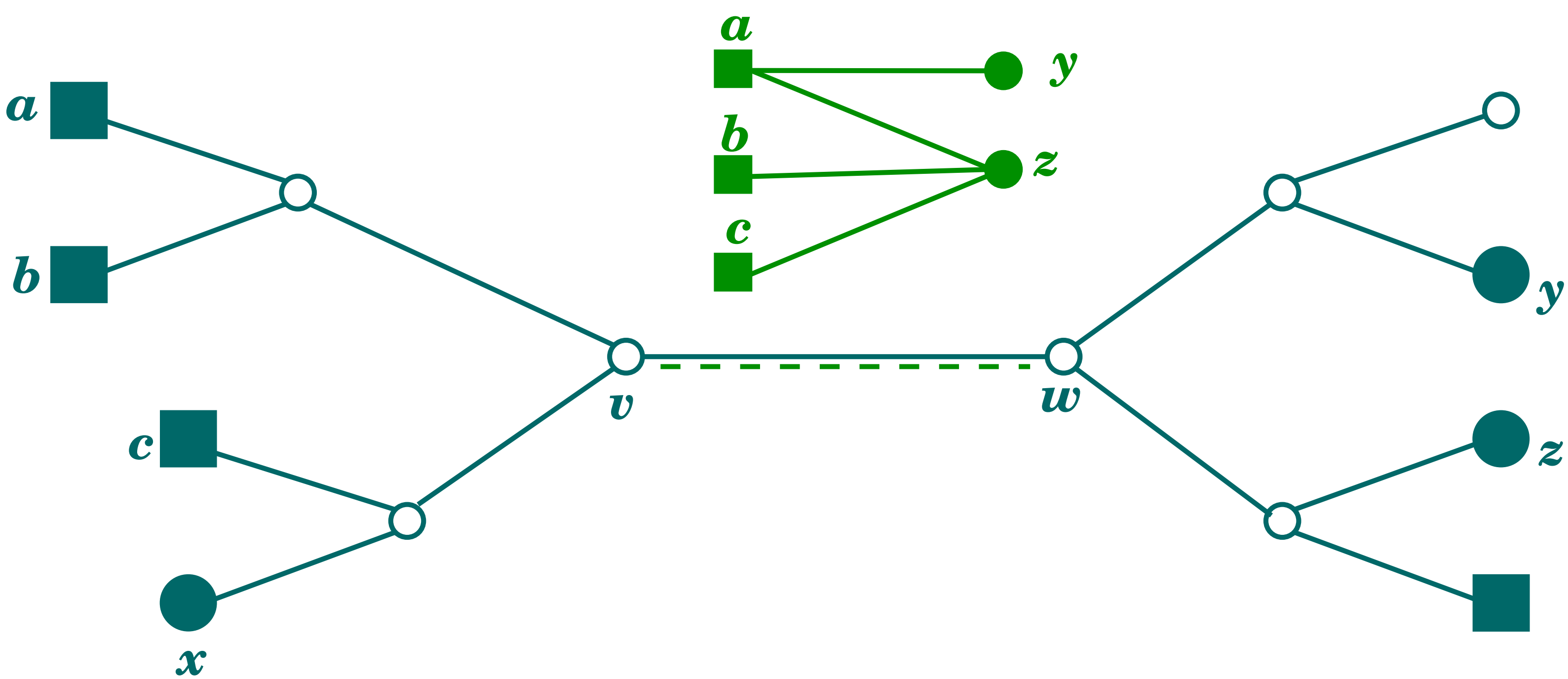
- Interest sets: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.

Tree Algorithm Diagram



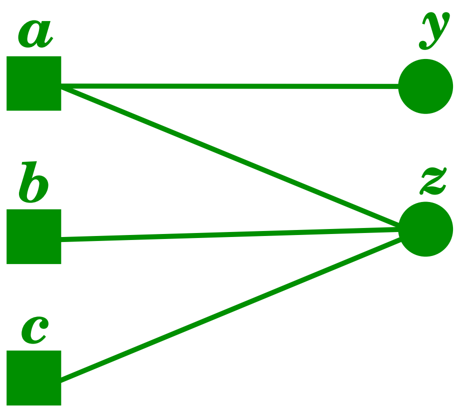
- Interest sets: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.
- Question: What is the **minimum** we can pay on edge vw?

Tree Algorithm Diagram

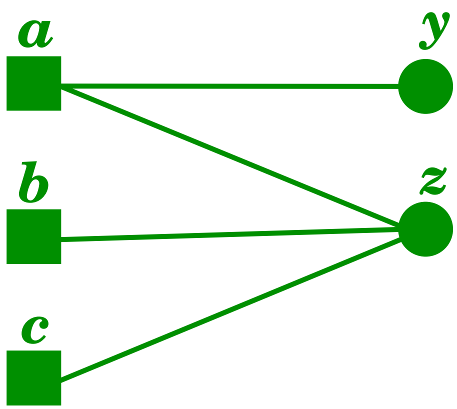


- Interest sets: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.
- Question: What is the **minimum** we can pay on edge vw?

Bipartite Minimum Weight Vertex Cover

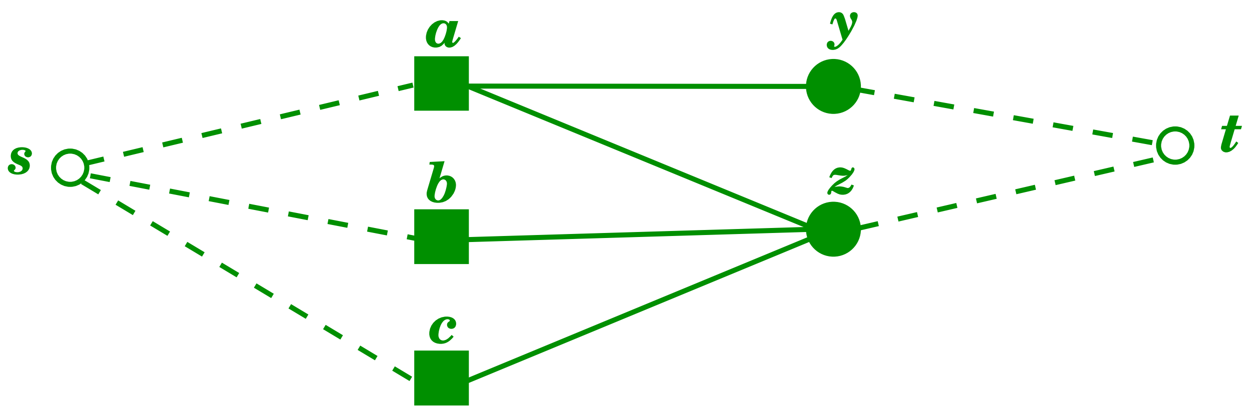


Bipartite Minimum Weight Vertex Cover



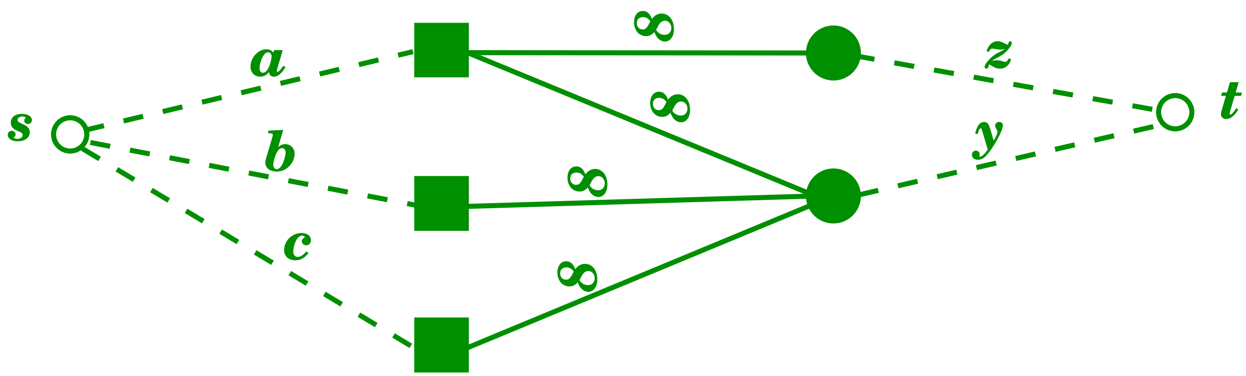
- **Well Known:** For bipartite $G_{vw} = (A \cup B, E)$, MWVC $\in P$ (Max flow). Find min cut R, to get MWVC $C_{vw} = (A \setminus R) \cup (B \cap R)$

Bipartite Minimum Weight Vertex Cover



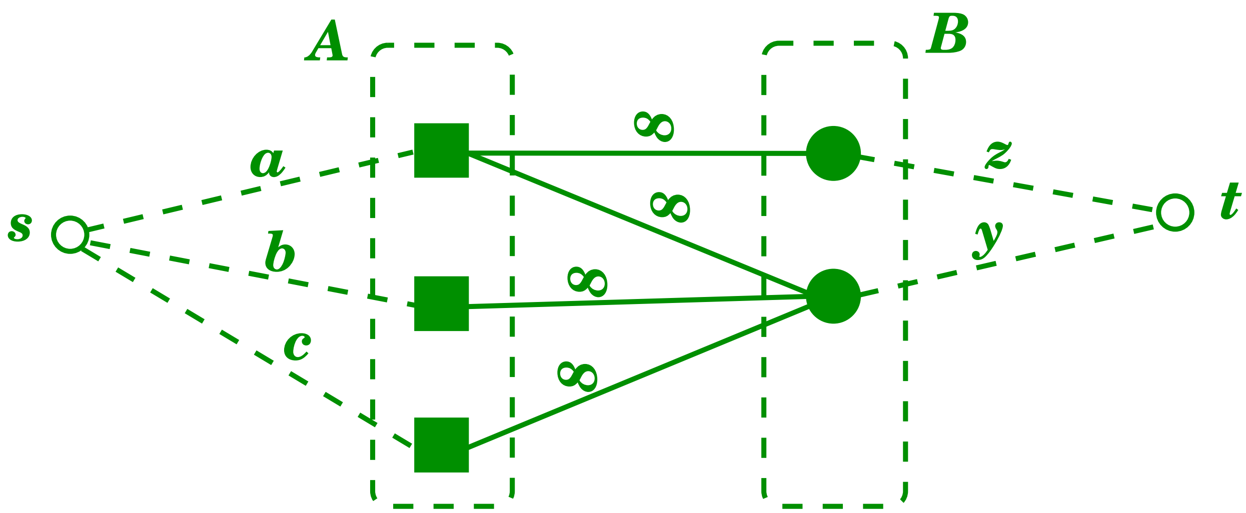
- **Well Known:** For bipartite $G_{vw} = (A \cup B, E)$, MWVC $\in P$ (Max flow). Find min cut R, to get MWVC $C_{vw} = (A \setminus R) \cup (B \cap R)$

Bipartite Minimum Weight Vertex Cover



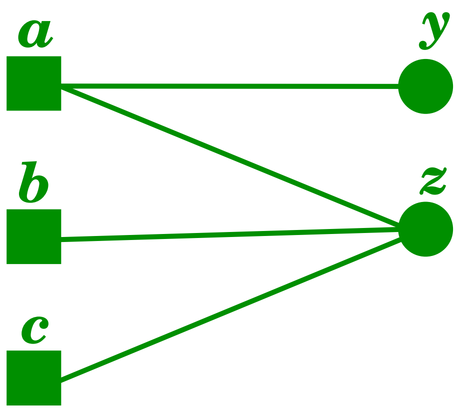
- **Well Known:** For bipartite $G_{vw} = (A \cup B, E)$, MWVC $\in P$ (Max flow). Find min cut R, to get MWVC $C_{vw} = (A \setminus R) \cup (B \cap R)$

Bipartite Minimum Weight Vertex Cover



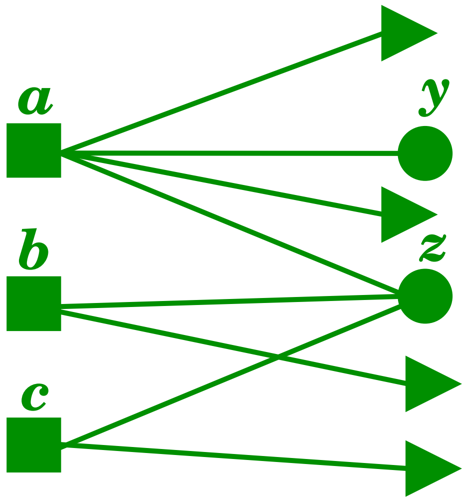
- **Well Known:** For bipartite $G_{vw} = (A \cup B, E)$, MWVC $\in P$ (Max flow). Find min cut R, to get MWVC $C_{vw} = (A \setminus R) \cup (B \cap R)$

Bipartite Minimum Weight Vertex Cover



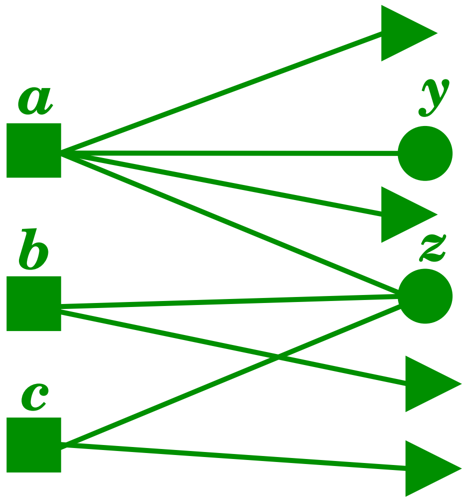
- **Well Known:** For bipartite $G_{vw} = (A \cup B, E)$, MWVC $\in P$ (Max flow). Find min cut R, to get MWVC $C_{vw} = (A \setminus R) \cup (B \cap R)$
- **Application:** Set $A = P_{vw}$ and $B = Q_{vw}$

Bipartite Minimum Weight Vertex Cover



- **Well Known:** For bipartite $G_{vw} = (A \cup B, E)$, MWVC $\in P$ (Max flow). Find min cut R, to get MWVC $C_{vw} = (A \setminus R) \cup (B \cap R)$
- **Application:** Set $A = P_{vw}$ and $B = Q_{vw} \cup \{x_{ij} \mid (i,j) \in X_{vw}\}$ for **response costs**.

Bipartite Minimum Weight Vertex Cover

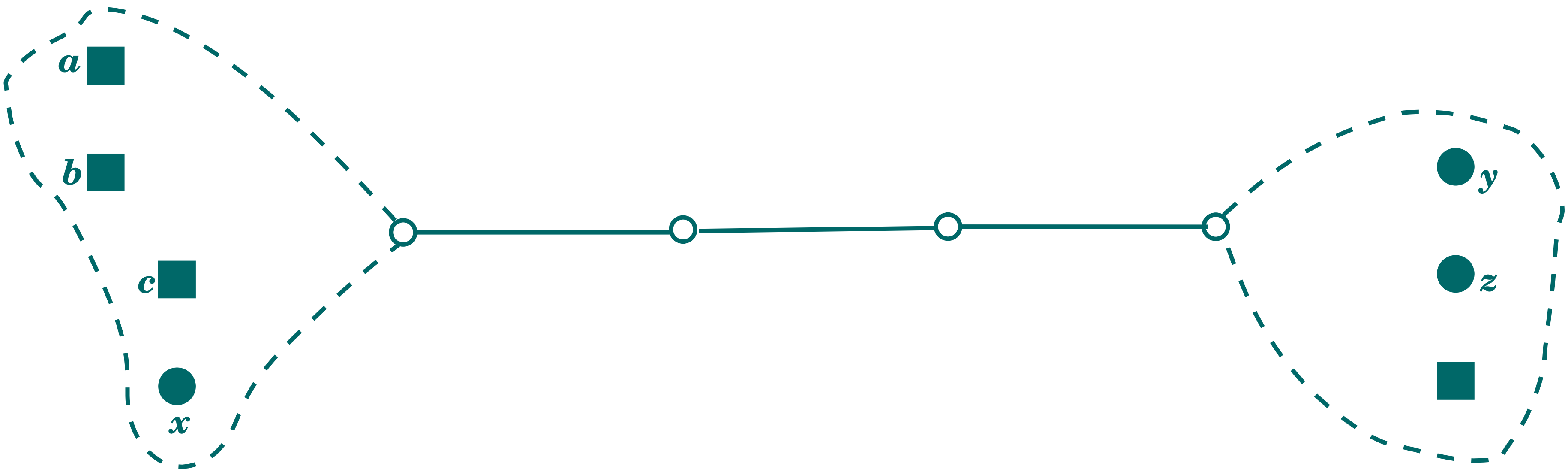


- **Well Known:** For bipartite $G_{vw} = (A \cup B, E)$, MWVC $\in P$ (Max flow). Find min cut R, to get MWVC $C_{vw} = (A \setminus R) \cup (B \cap R)$

- **Application:** Set $A = P_{vw}$ and $B = Q_{vw} \cup \{x_{ij} \mid (i,j) \in X_{vw}\}$ for **response costs**.

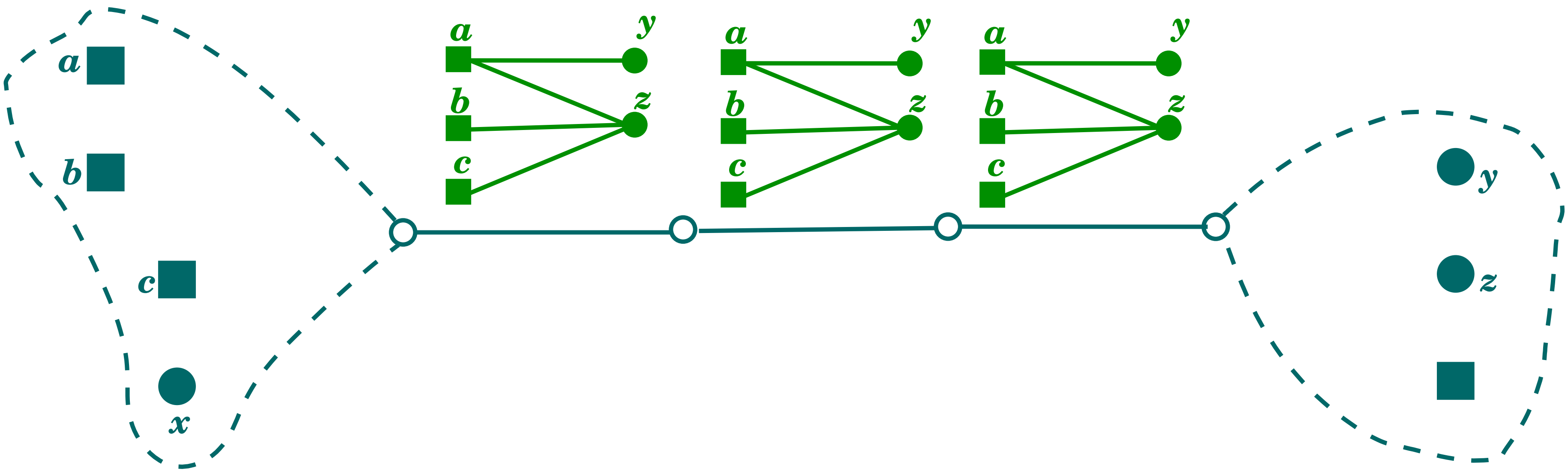
Lemma 1. *For each arc $e = vw$, the MWVC weight of G_{vw} is the* **minimum** *value paid for vw in* **any** *optimal solution.*

Tree Algorithm — Tiebreaking



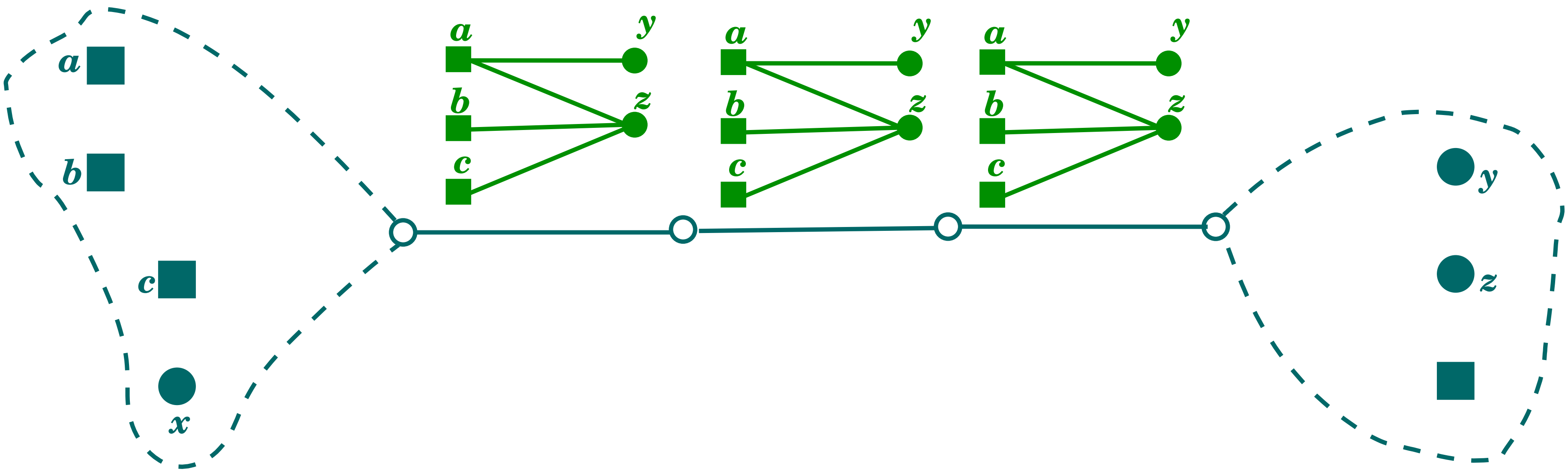
- Long chain, no sources or sinks.

Tree Algorithm — Tiebreaking



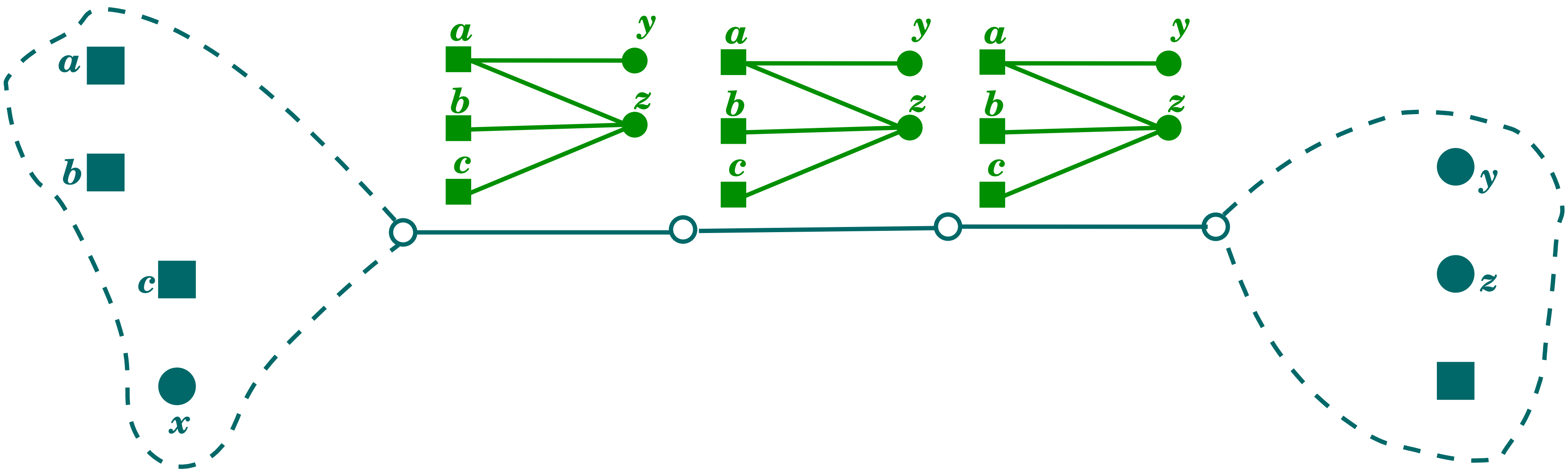
- Long chain, no sources or sinks.
- Identical Bipartite graph problem

Tree Algorithm — Tiebreaking



- 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?**

Tree Algorithm — Tiebreaking



- 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?**

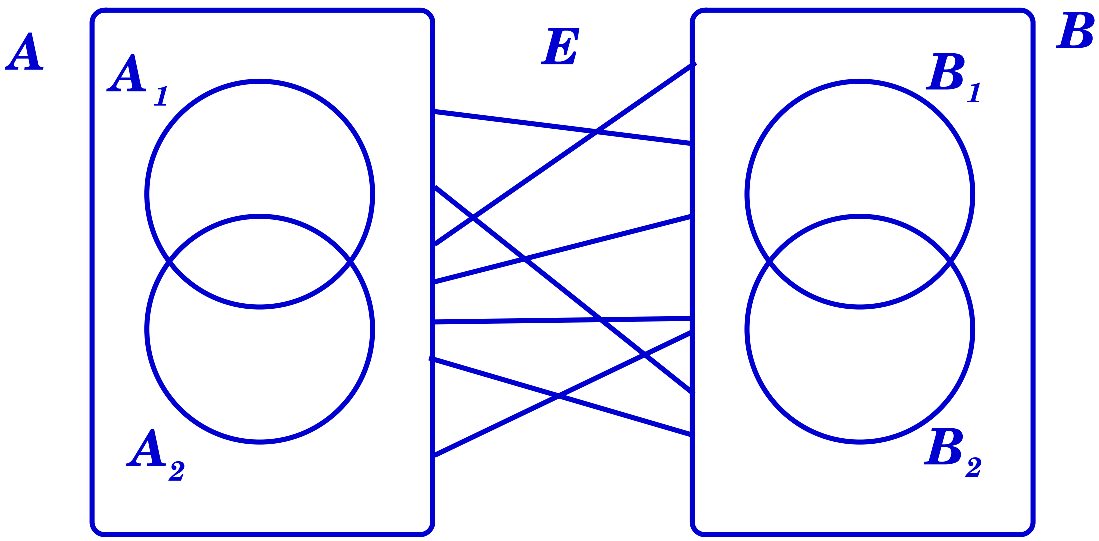
Defn: In bipartite $G=(A\cup B,E)$, an MWVC is A-**maximum** if it has maximum weight in A.

A Consistent Tiebreaking Solution

Defn: In bipartite $G = (A \cup B, E)$, an MWVC is A-**maximum** if it has maximum weight in A.

A Consistent Tiebreaking Solution

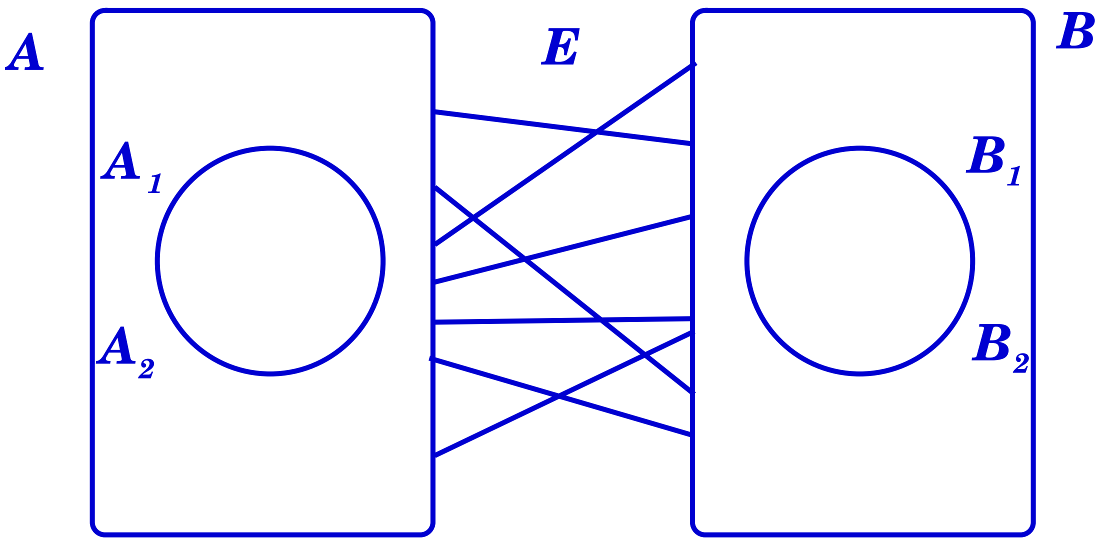
Defn: In bipartite $G = (A \cup B, E)$, an MWVC is A-**maximum** if it has maximum weight in A.



Lemma 2. *Let* $G = (A \cup B, E)$*, let* $A_1, A_2 \subseteq A$ *and let* $B_1, B_2 \subseteq B$*. If* $A_1 \cup B_1$ *and* $A_2 \cup B_2$ *are both* A*-maximum minimum weight vertex covers, ...*

A Consistent Tiebreaking Solution

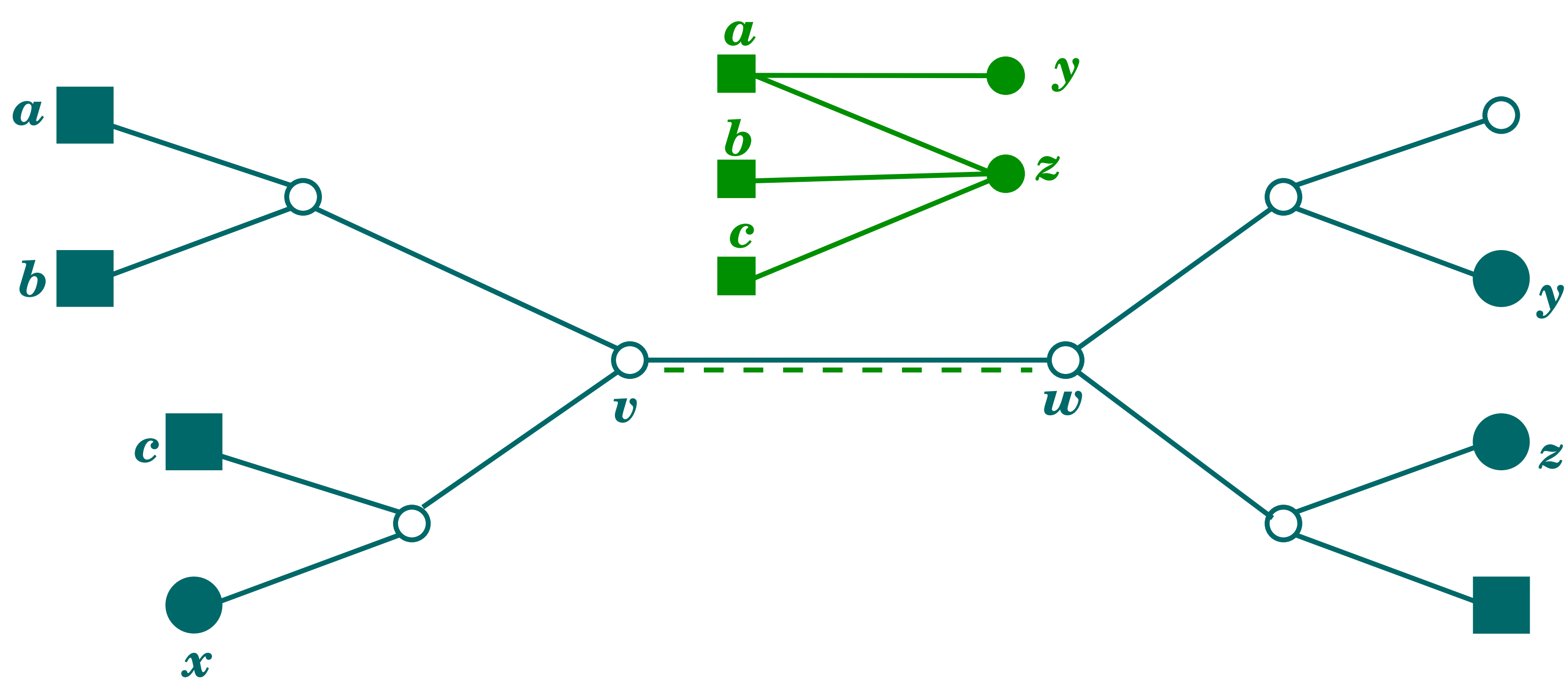
Defn: In bipartite $G = (A \cup B, E)$, an MWVC is A-**maximum** if it has maximum weight in A.



Lemma 2. *Let* $G = (A \cup B, E)$*, let* $A_1, A_2 \subseteq A$ *and let* $B_1, B_2 \subseteq B$*. If* $A_1 \cup B_1$ *and* $A_2 \cup B_2$ *are both* A*-maximum minimum weight vertex covers, ... then* $A_1 = A_2$ *and* $B_1 = B_2$*.*

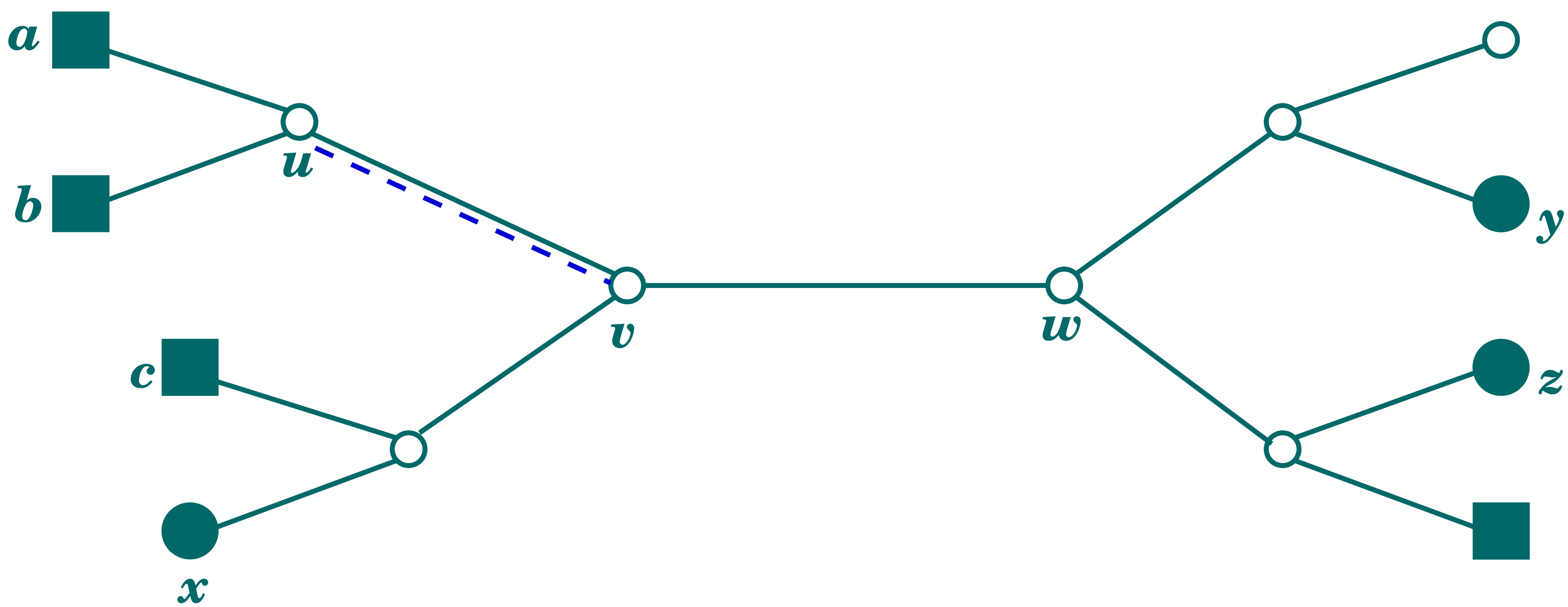
Unique solution per edge!

Tree Algorithm — Structural Continuity



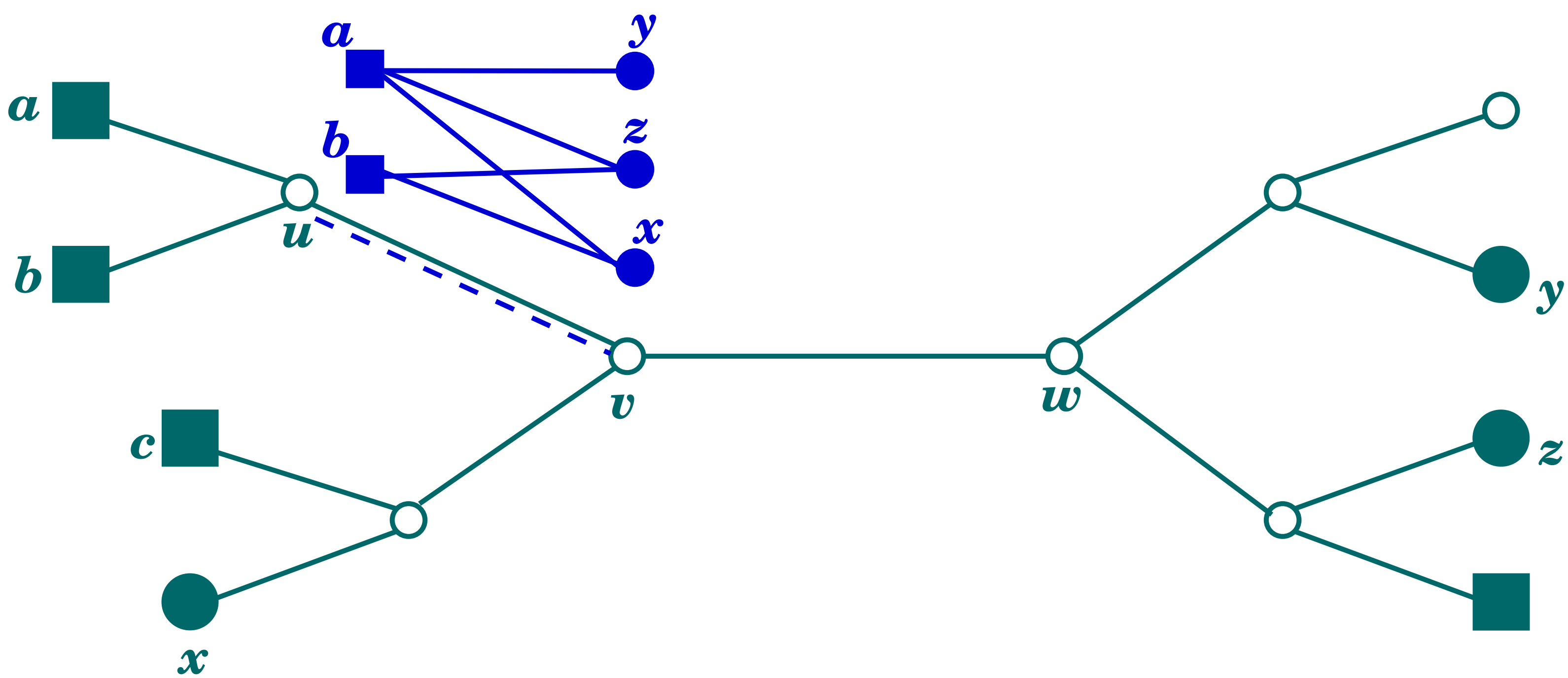
- Interest sets — recall: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.

Tree Algorithm — Structural Continuity



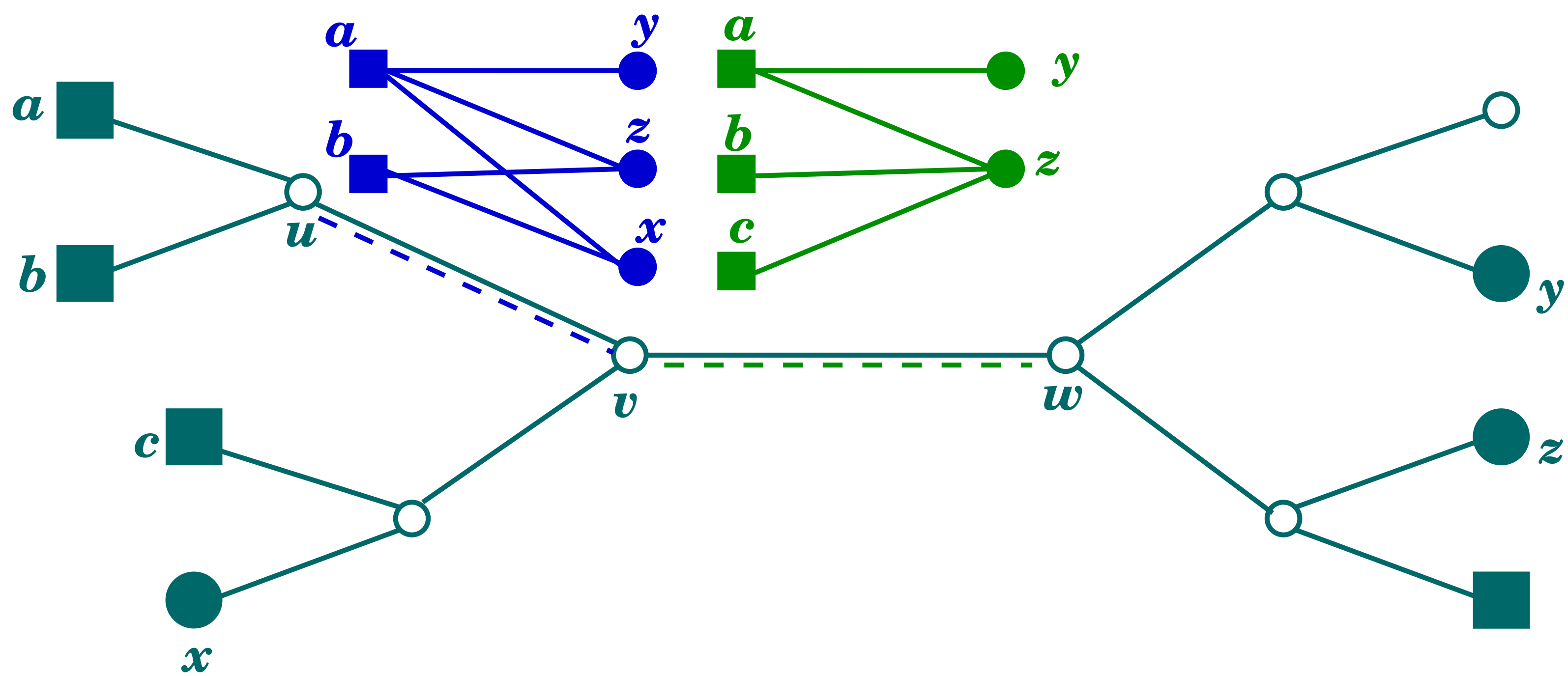
- Interest sets — recall: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.
- What about G_{uv}? Clearly different.

Tree Algorithm — Structural Continuity



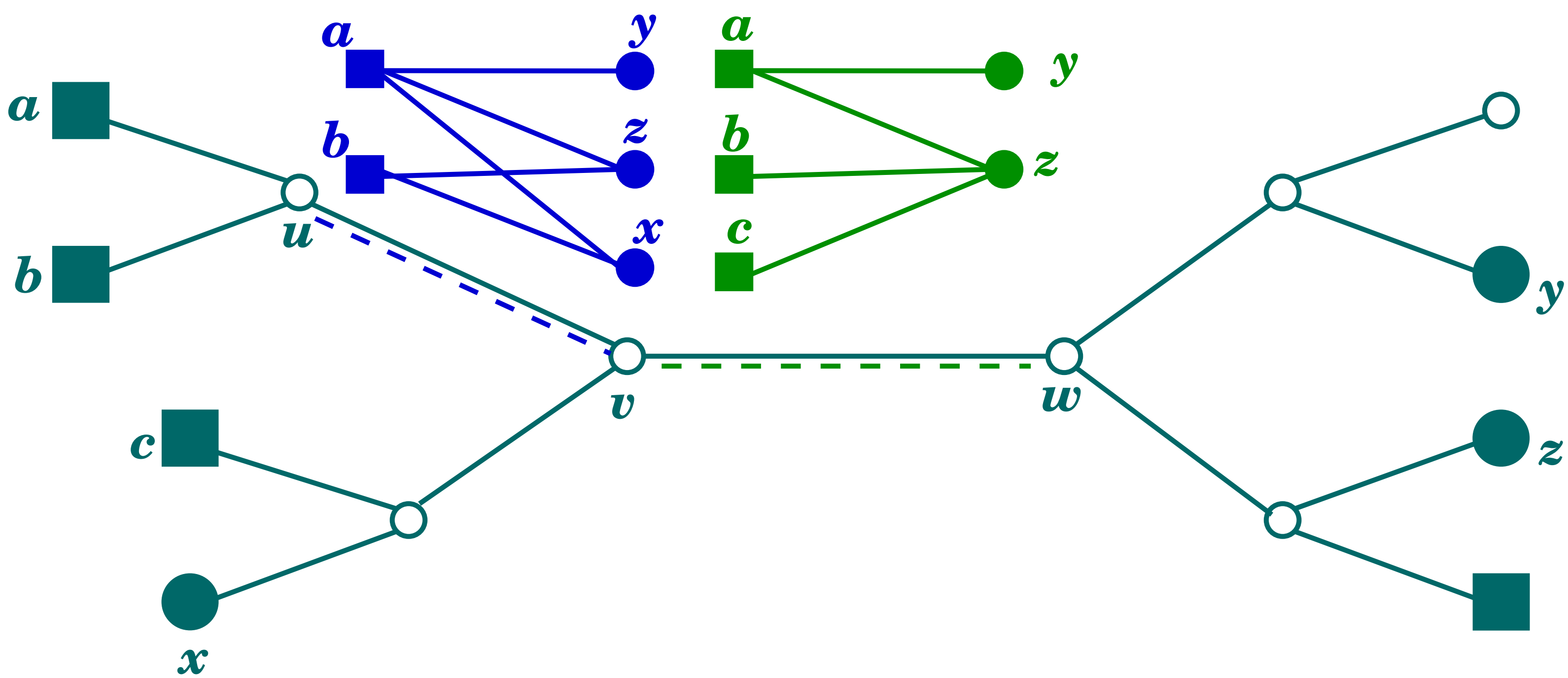
- Interest sets — recall: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.
- What about G_{uv}? Clearly different.

Tree Algorithm — Structural Continuity



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

Tree Algorithm — Structural Continuity

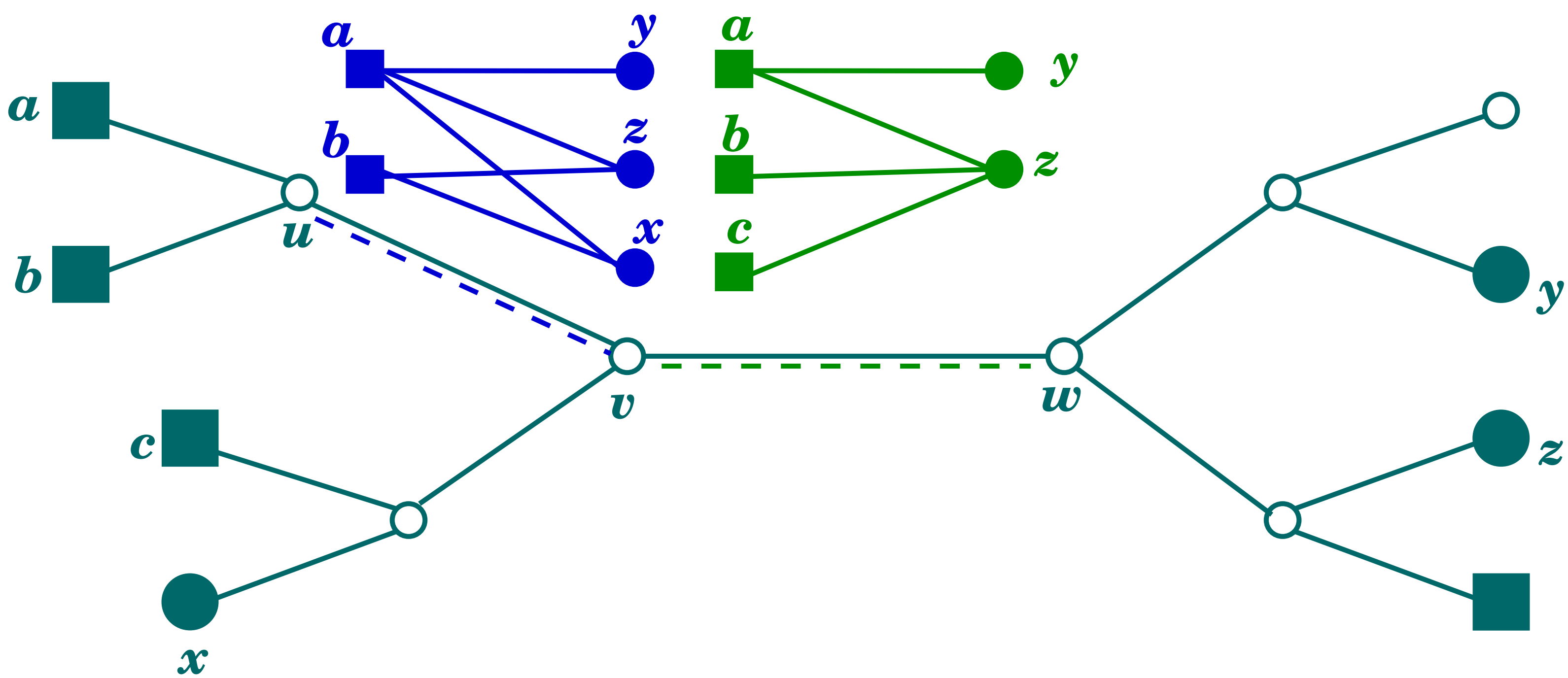


- Interest sets — recall: $\{x, z\}$ want $\{a, b, c\}$; y wants only a.
- What about G_{uv}? Clearly different.
- **Are push trees, pull trees and response paths connected?**

Lemma 3. *If we compute push-maximum MWVC for every edge, then Push and Pull subtrees are connected.*

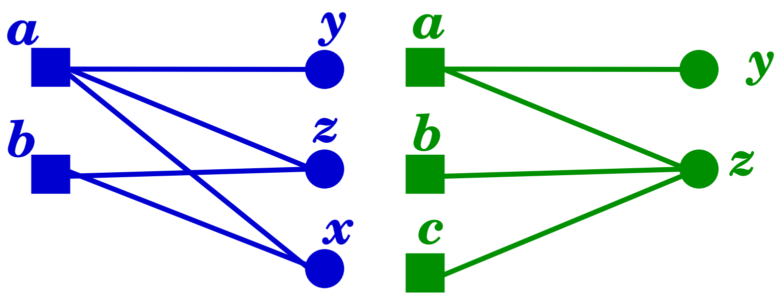
Structural Continuity Solution

Yes!!!



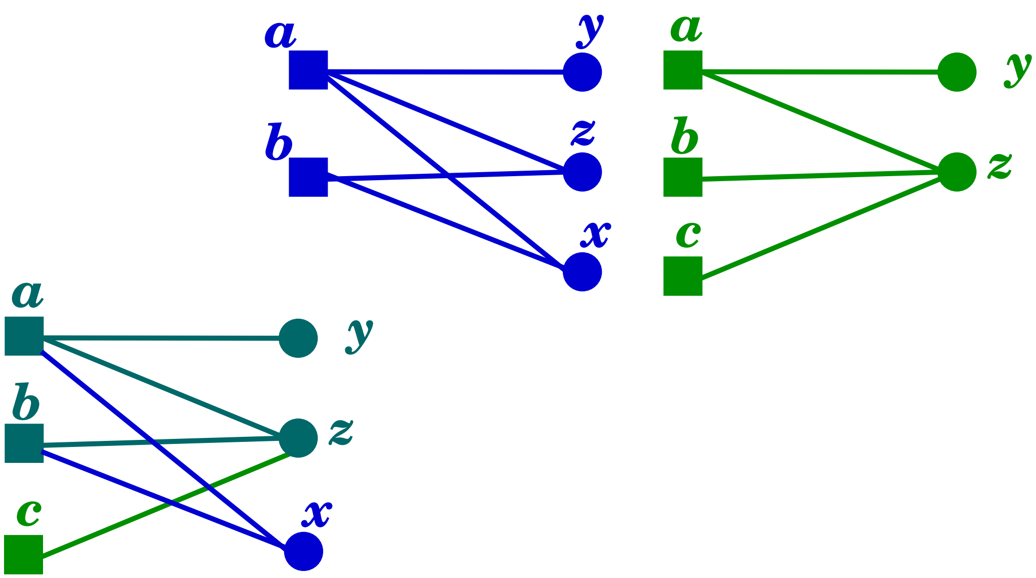
Structural Continuity Solution

Yes!!!



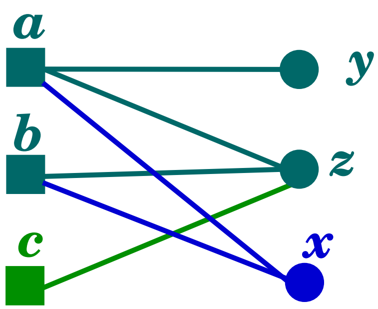
Structural Continuity Solution

Yes!!!



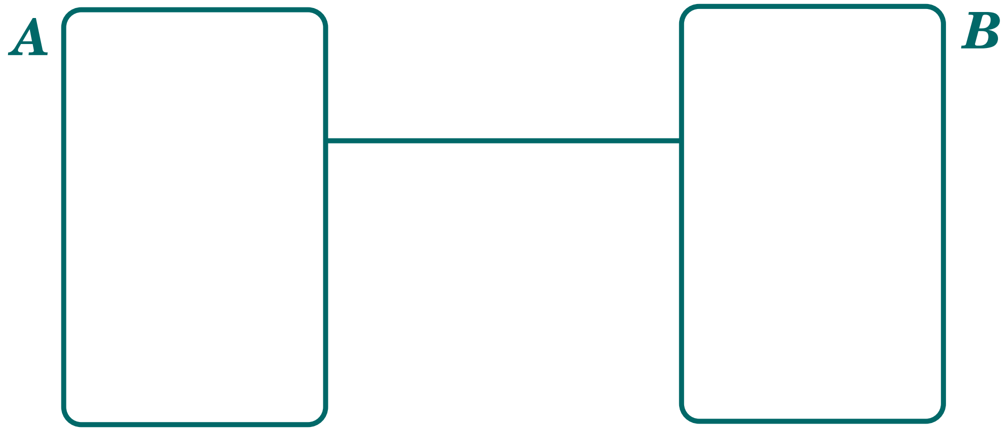
Structural Continuity Solution

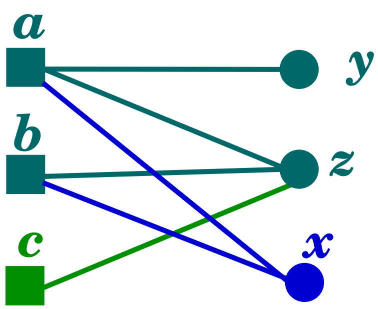
Yes!!!



Structural Continuity Solution

Yes!!!

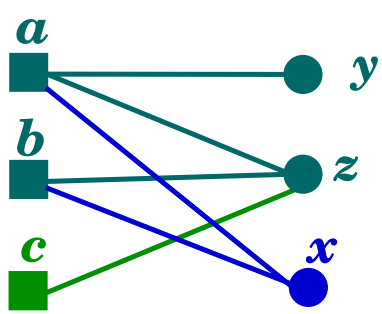


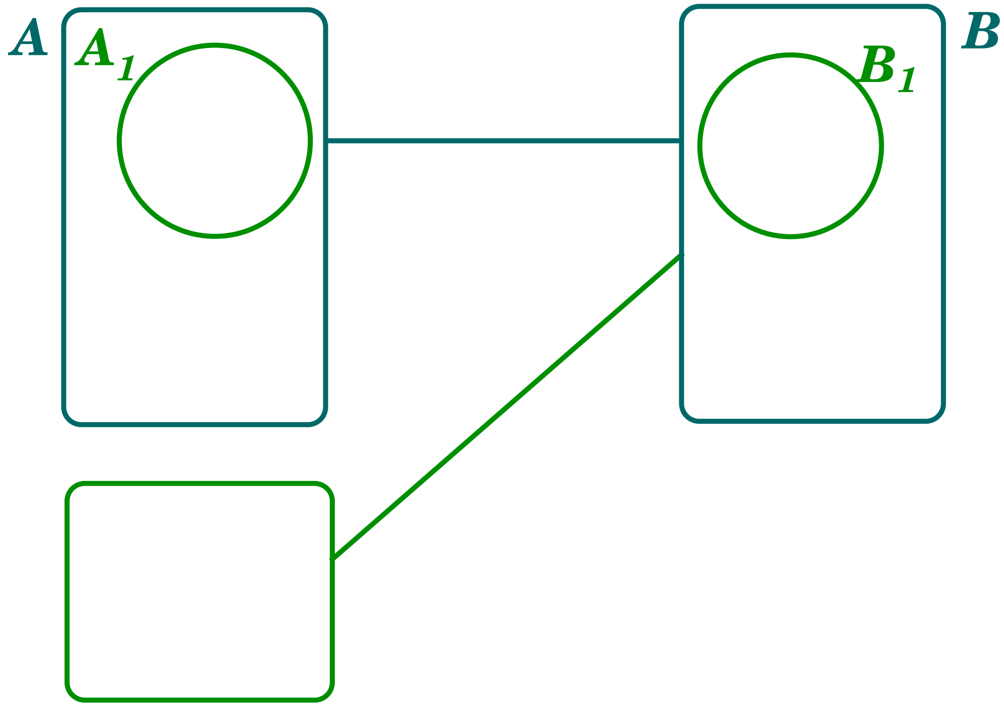


Lemma 4. *Let uvw be two consecutive edges, let A be the set of push nodes in G_{uv}, and let B be the set of (non-push) nodes in G_{vw}.*

Structural Continuity Solution

Yes!!!



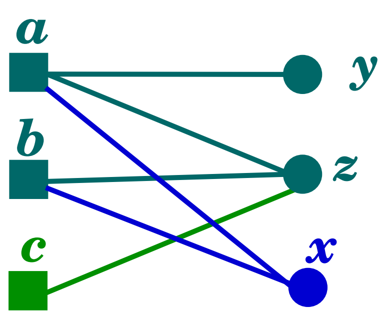


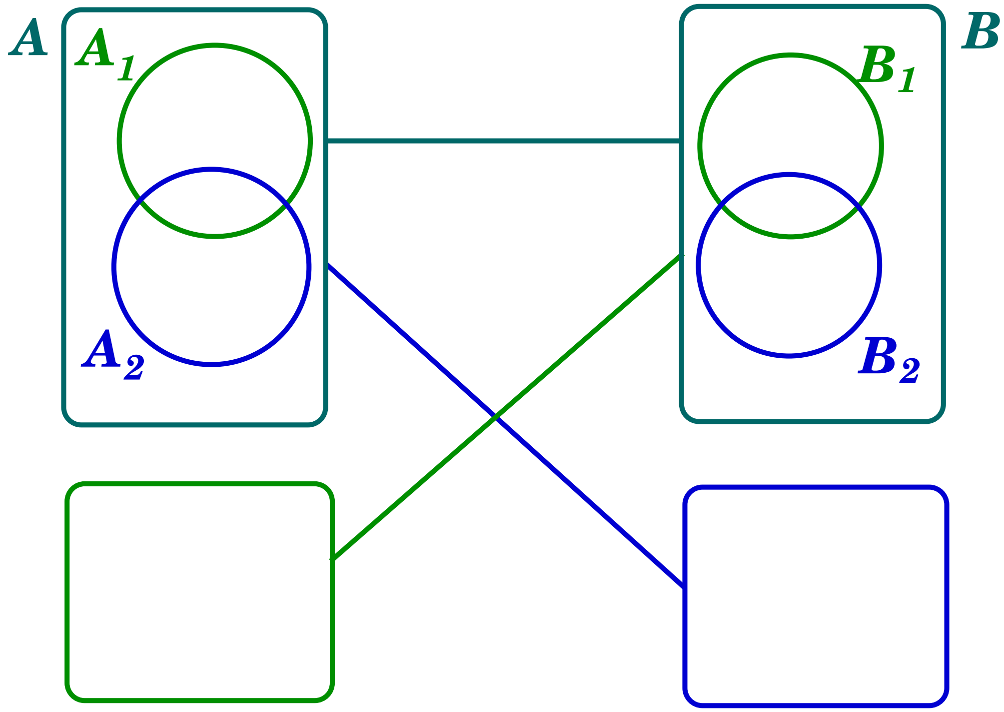
Lemma 4. *Let* uvw *be two consecutive edges, let* A *be the set of push nodes in* G_{uv}*, and let* B *be the set of (non-push) nodes in* G_{vw}*. Let*

- A_1, B_1 *be parts of push-maximum MWVC of* G_{vw} *in* A, B *resp., and*

Structural Continuity Solution

Yes!!!



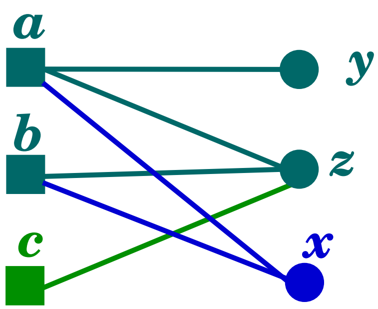


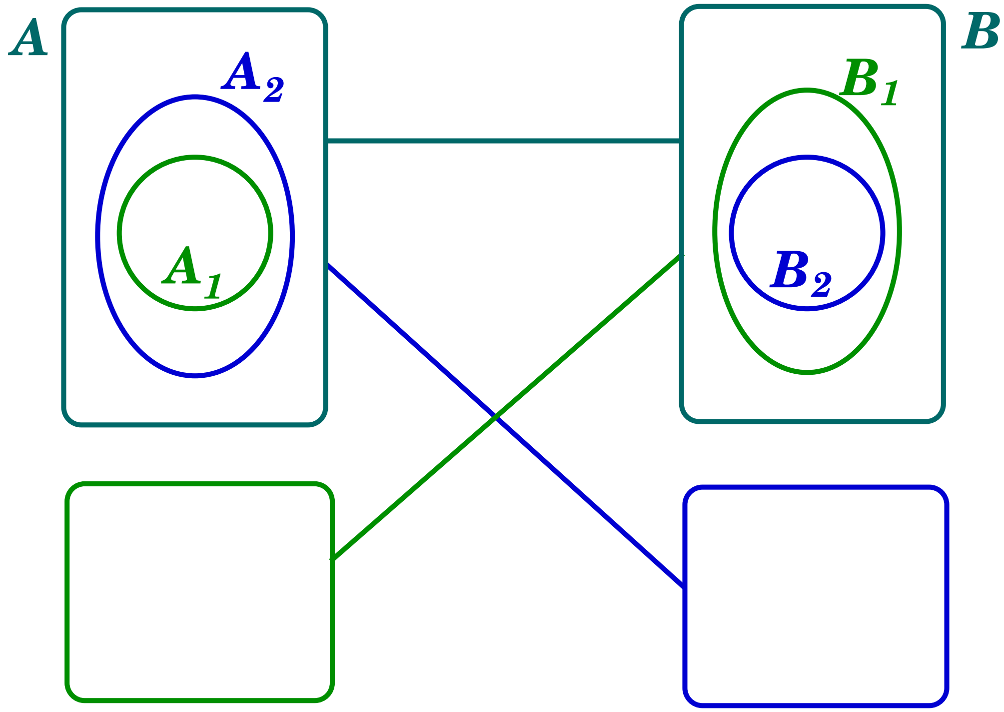
Lemma 4. *Let uvw be two consecutive edges, let A be the set of push nodes in G_{uv}, and let B be the set of (non-push) nodes in G_{vw}. Let*

- *A_1, B_1 be parts of push-maximum MWVC of G_{vw} in A, B resp., and*
- *A_2, B_2 be parts of push-maximum MWVC of G_{uv} in A, B resp.*

Structural Continuity Solution

Yes!!!





Lemma 4. *Let* uvw *be two consecutive edges, let* A *be the set of push nodes in* G_{uv}*, and let* B *be the set of (non-push) nodes in* G_{vw}*. Let*

- A_1, B_1 *be parts of push-maximum MWVC of* G_{vw} *in* A, B *resp., and*
- A_2, B_2 *be parts of push-maximum MWVC of* G_{uv} *in* A, B *resp.*
- **then** $A_1 \subseteq A_2$ **and** $B_1 \supseteq B_2$**.**

Push/Pull subtrees, Response paths are connected!

Tree Algorithm

for each directed edge uv
  construct the graph G_{uv}
  find its canonical minimum cut C_{uv}
  for all $i \in P_{uv}$
    if $i \in C_{uv}$ then include uv in T_i
  for all $j \in Q_{vu}$
    if $j \in C_{uv}$ then include uv in T'_j
  for all $(i, j) \in X_{uv}$
    if $x_{ij} \in C_{uv}$ then include uv 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. **Chlebìk & Chlebìkovà SWAT'02**

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

Multicast Model – General Graph Approximation algorithm

- Reduction from Min Steiner Tree; NP-hard to **approximate** within 96/95. **Chlebìk & Chlebìkovà 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.*

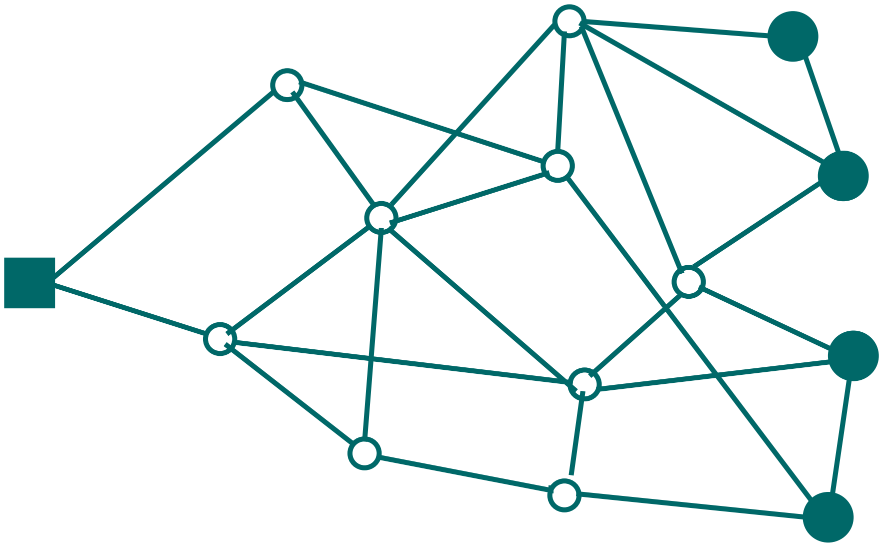
General Graph Approximation algorithm ctd.

- Bound Derivation
  - ∗ Choose T randomly from distribution of metric-spanning trees.
  - ∗ Project structures in G into T. Obtain feasible solution for T.
  - ∗ $OPT(T) \leq O(\log n) \cdot OPT(G)$.
- Approximation Algorithm
  - ∗ Solve T exactly using our algorithm.
  - ∗ Project structures in T into G. Obtain feasible solution for G.
  - ∗ $ALG(G) \leq 2 \cdot OPT(T)$

General Graph Approximation algorithm ctd.

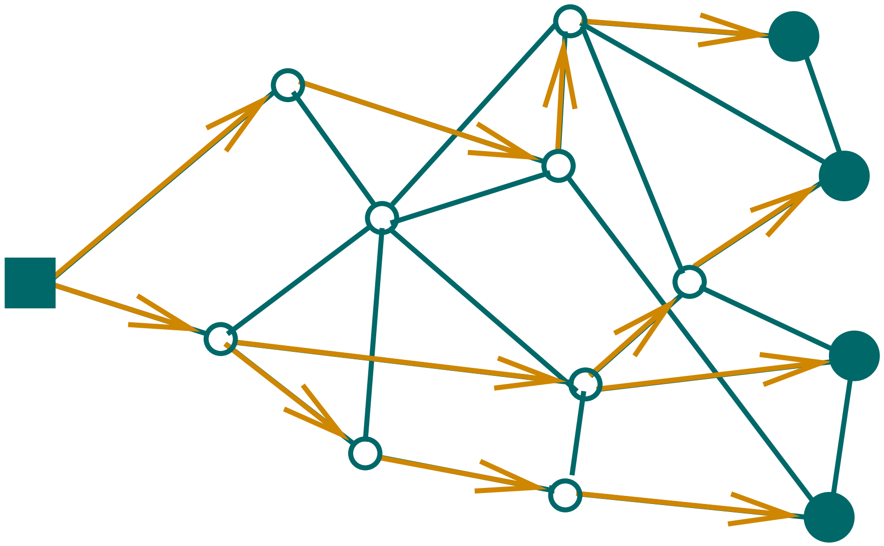
- Bound Derivation
  - $*$ Choose T randomly from distribution of metric-spanning trees.
  - $*$ Project structures in G into T. Obtain feasible solution for T.
  - $*$ $OPT(T) \le O(\log n) \cdot OPT(G)$.
- Approximation Algorithm
  - $*$ Solve T exactly using our algorithm.
  - $*$ Project structures in T into G. Obtain feasible solution for G.
  - $*$ $ALG(G) \le 2 \cdot OPT(T) \le O(\log n) \cdot OPT(G)$.

Multicast Model – Hardness



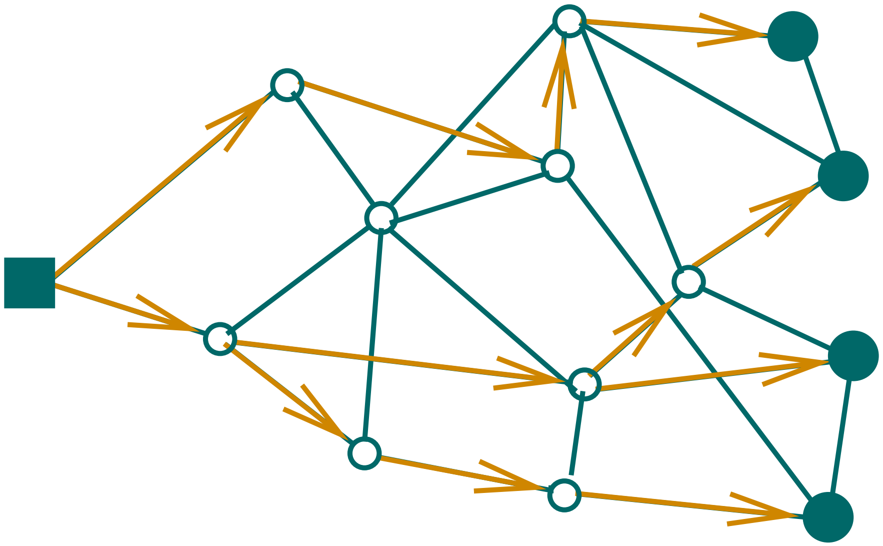
- Multicast problem with(out) aggregation: easy reduction from **Min Steiner tree**.
  - Arbitrary node becomes low-freq source
  - Rest become high-freq Sink nodes
  - Each interested in Source

Multicast Model – Hardness



- Multicast problem with(out) aggregation: easy reduction from **Min Steiner tree**.
  - ∗ Arbitrary node becomes low-freq source
  - ∗ Rest become high-freq Sink nodes
  - ∗ Each interested in Source

Multicast Model – Hardness



- Multicast problem with(out) aggregation: easy reduction from **Min Steiner tree**.
  - Arbitrary node becomes low-freq source
  - Rest become high-freq Sink nodes
  - Each interested in Source
- Min Steiner Tree NP-hard to **approximate** within 96/95. **Chlebìk & Chlebìkovà SWAT'02**

The Unicast Model

The Unicast Model

- Given (non-)metric distances d_{uv} for every pair $(u, v) \in V \times V$.
- $\mathsf{SetC}(u, S) = \sum_{k \in S} d_{uk}$
- find push-sets P_i and pull-sets Q_j that minimize total communication cost:

$$\sum_{i \in \mathcal{P}} p_i \sum_{k \in P_i} d_{ik} + \sum_{j \in \mathcal{Q}} q_j \sum_{k \in Q_j} d_{kj} + \sum_{j \in \mathcal{Q}} q_j \cdot \mathsf{RespC}(j),$$

- and satisfies: for all $i \in I_j$, $P_i \cap Q_j \neq \emptyset$
- where

$$\mathsf{RespC}(j) = \begin{cases} \mathsf{SetC}(j, Q_j) & \text{(aggregation model)} \\ \sum_{i \in I_j} \mathsf{MinC}(P_i \cap Q_j, j) & \text{otherwise.} \end{cases}$$

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: $$\sum_{i \in \mathcal{P}} p_i \sum_{k \in V} d_{ik} x_{ik} + \sum_{j \in \mathcal{Q}} q_j \sum_{k \in V} d_{kj} y_{kj}$$

subject to $$\begin{cases} r_{ijk} \leq x_{ik} \\ r_{ijk} \leq y_{kj} \\ \sum_k r_{ijk} \geq 1 \end{cases}$$, where $x_{ik}, y_{kj}, r_{ijk} \in \{0, 1\}$.

Unicast Model with Aggregation Nonmetric Case via Randomized Rounding

- Convert to LP: Use ≥ 0 instead of $\in \{0, 1\}$
- Solve and discard values $\leq 1/n^2$ and scale by $n/(n-1)$
- Round values up to powers of $1/2$, obtain $(\tilde{x}, \tilde{y}, \tilde{z})$
- For node k and $0 \leq p < 2 \log n$, define X_{pk} as i such that $\tilde{x}_{ik} \geq 1/2^p$.
- $\forall p, k$: with probability $\min\{1, (\log n)/2^p\}$ add k to P_i and Q_j for all $i \in X_{pk}$ and $j \in Y_{pk}$.

Theorem 3. *With high probability, solution is feasible, with cost* $O(\log n) \cdot OPT_{LP}$.

Unicast Model with Aggregation Nonmetric Case via Randomized Rounding – Proof

- Since $i \in X_{\log(\tilde{r}_{ijk})k}$ and $j \in Y_{\log(\tilde{r}_{ijk})k}$, $\Pr[k \in P_i \cap Q_j] \geq \min\{1, \tilde{r}_{ijk} \log(n)\}$.

- Clearly $\Pr[k \in P_i] \leq \sum_{p: i \in X_{pk}} (\log n)/2^p = 2\tilde{x}_{ik} \log n$ $\leq \min\{1, 2\tilde{x}_{ik} \log n\}$.

- $\Pr[P_i \cap Q_j = \emptyset] = \prod_k (1 - \tilde{r}_{ijk} \log n) \leq e^{-\sum_k \tilde{r}_{ijk} \log n} \leq 1/n^2$.

- Define r.v. C_i as push cost for i, and r.v. C_{ik} takes value d_{ik} with probability $\min\{1, 2\tilde{x}_{ik} \log n\}$.

- Chernoff-Hoeffding: w.h.p. $\sum_k C_{ik} \leq O(\log n) \cdot \sum_k d_{ik} \tilde{x}_{ik}$.

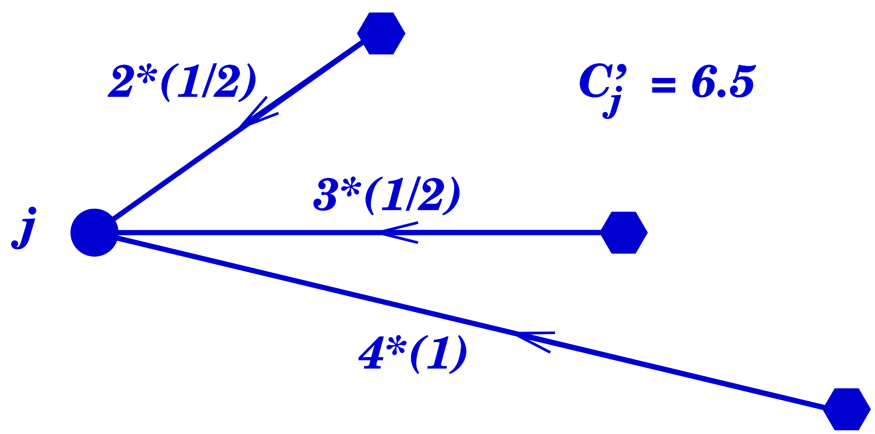
- Summing over all sources, sinks gives cost bound w.h.p.

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**

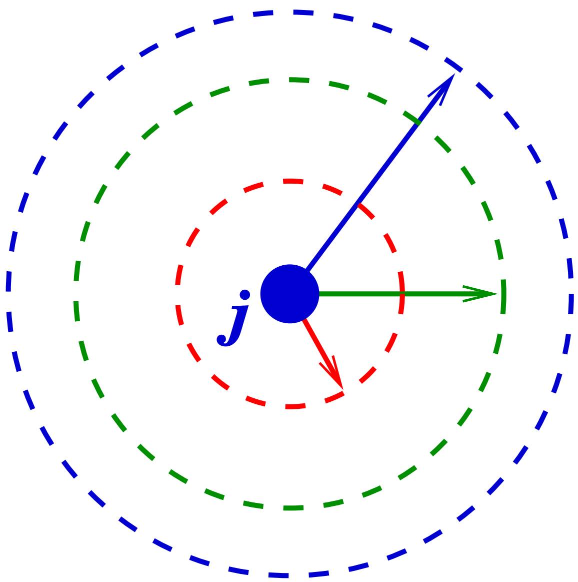
Unicast Model with Aggregation
Uniform Interest Sets in Metric Case — Intro

- **Basic definitions**
  - ∗ Optimal solution to the LP is (x^*, y^*, r^*).
  - ∗ LP gives cost lower bounds $C_i = \sum_k d_{ik} x^*_{ik}$ and $C'_j = \sum_k d_{kj} y^*_{kj}$



Unicast Model with Aggregation
Uniform Interest Sets in Metric Case — Intro

- **Basic definitions**
  - Optimal solution to the LP is (x^*, y^*, r^*).
  - LP gives cost lower bounds $C_i = \sum_k d_{ik} x^*_{ik}$ and $C'_j = \sum_k d_{kj} y^*_{kj}$
  - For node u, $r > 0$, define $B_u(r) = \{v : d_{uv} \leq r\}$.
  - Let $1 < \alpha < \beta$. Clearly $B_j(C'_j) \subseteq B_j(\alpha C'_j) \subseteq B_j(\beta C'_j)$

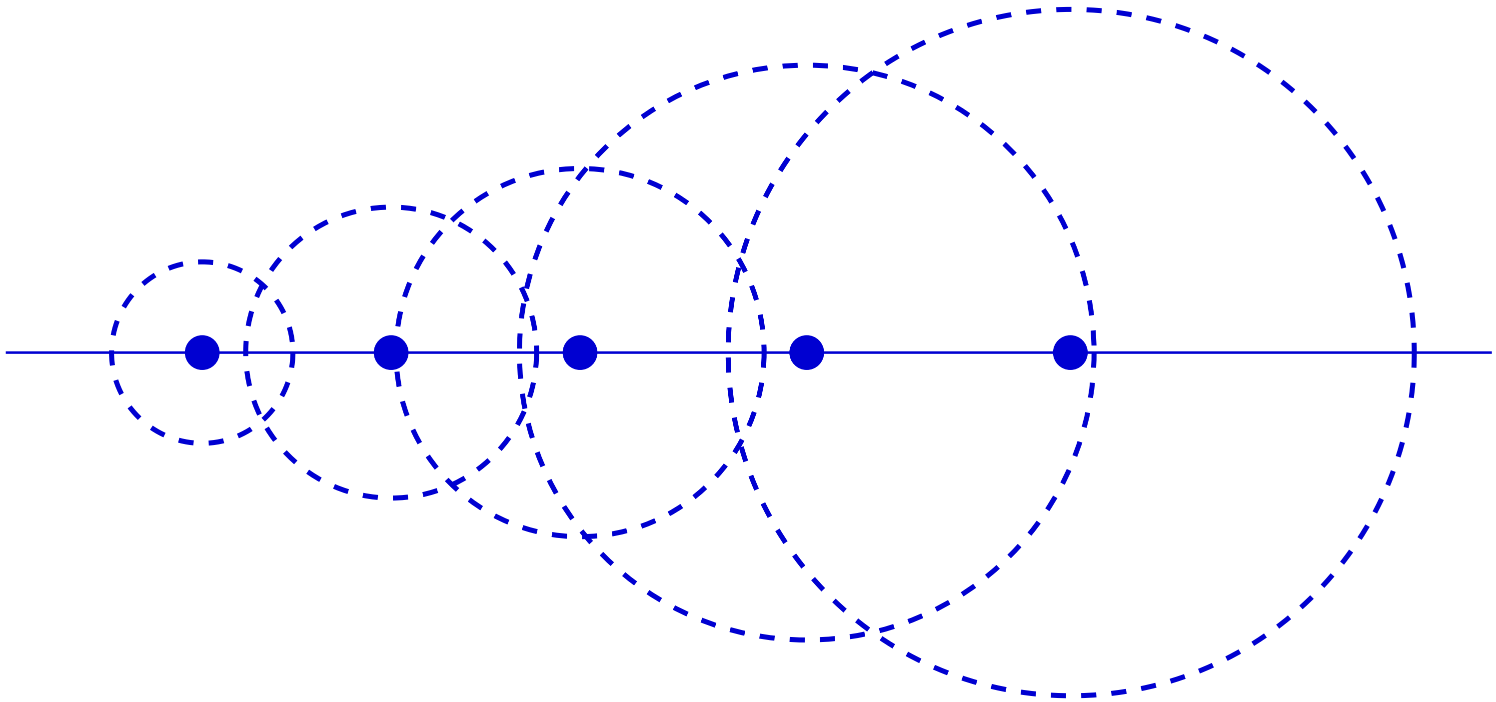


Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.

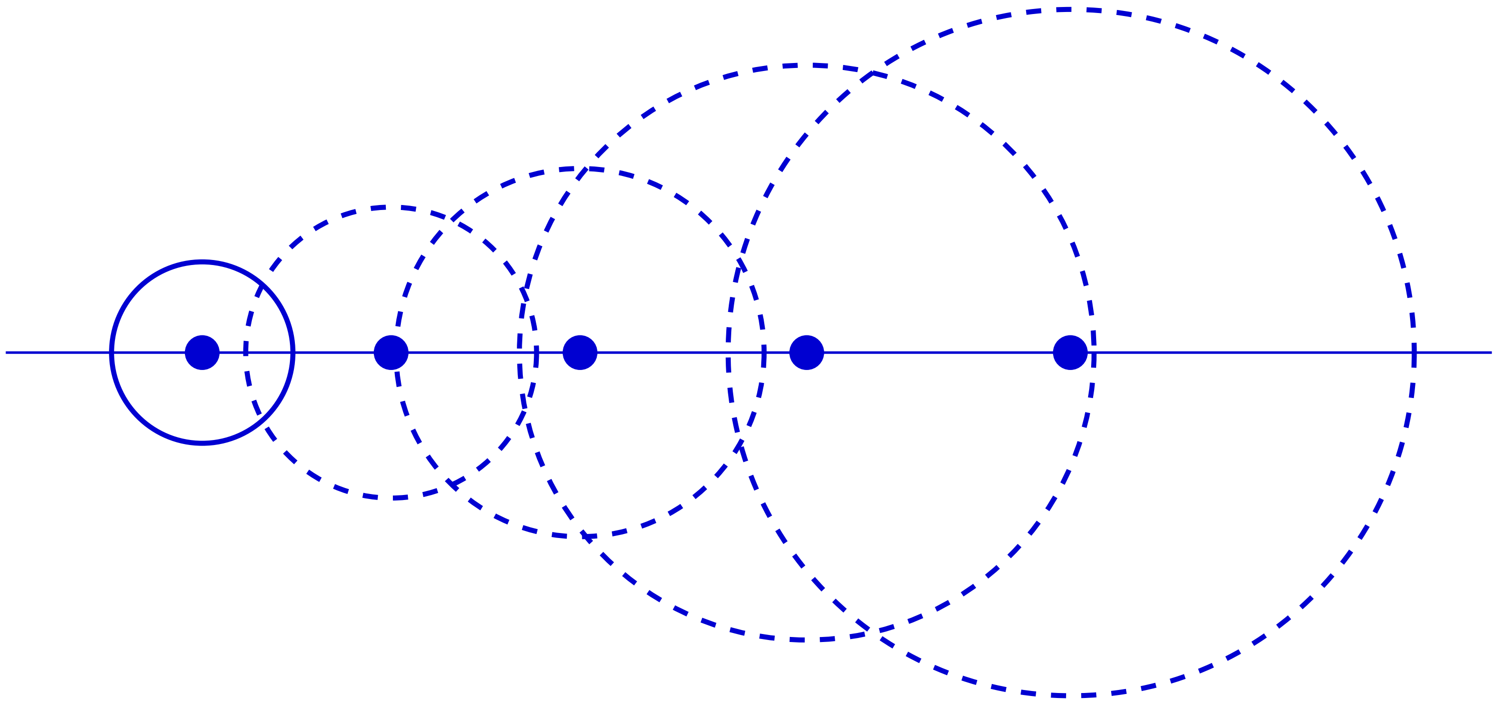
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



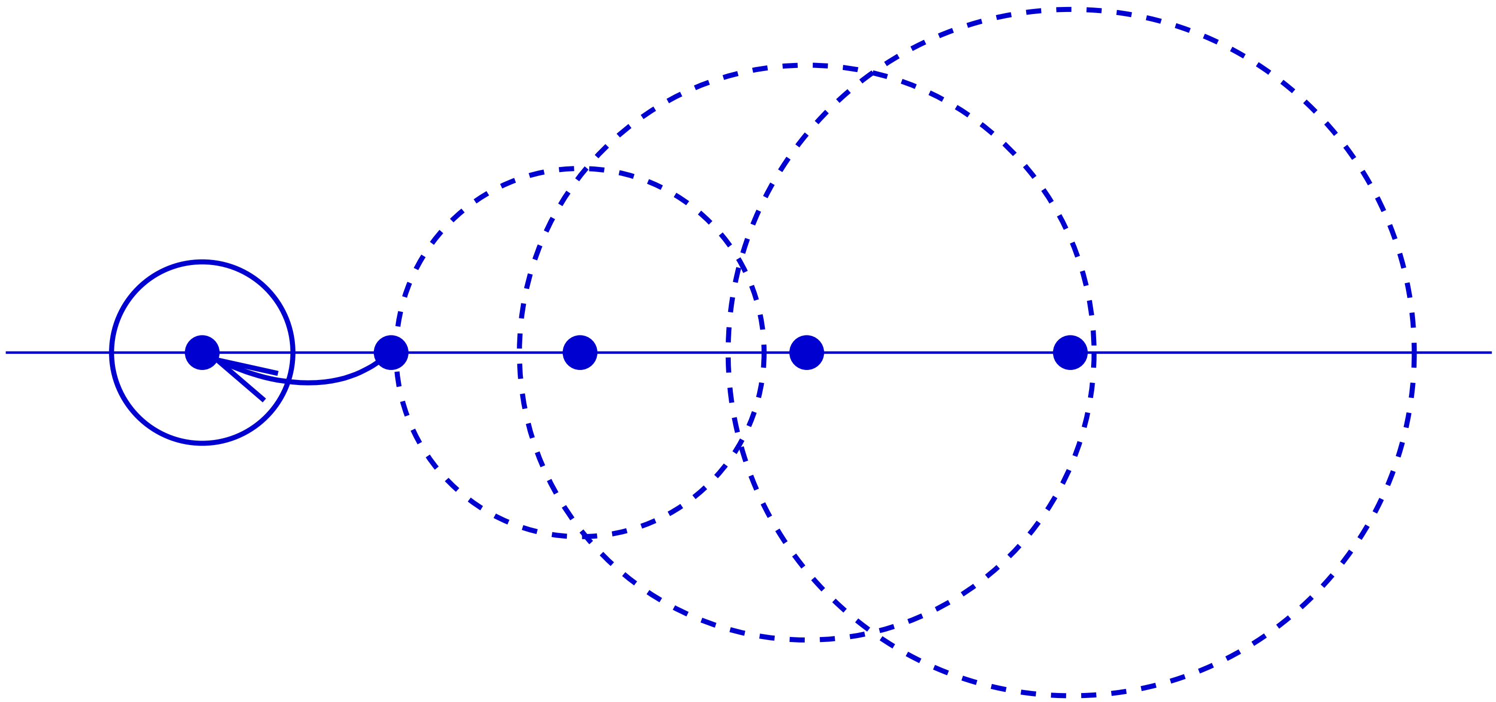
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



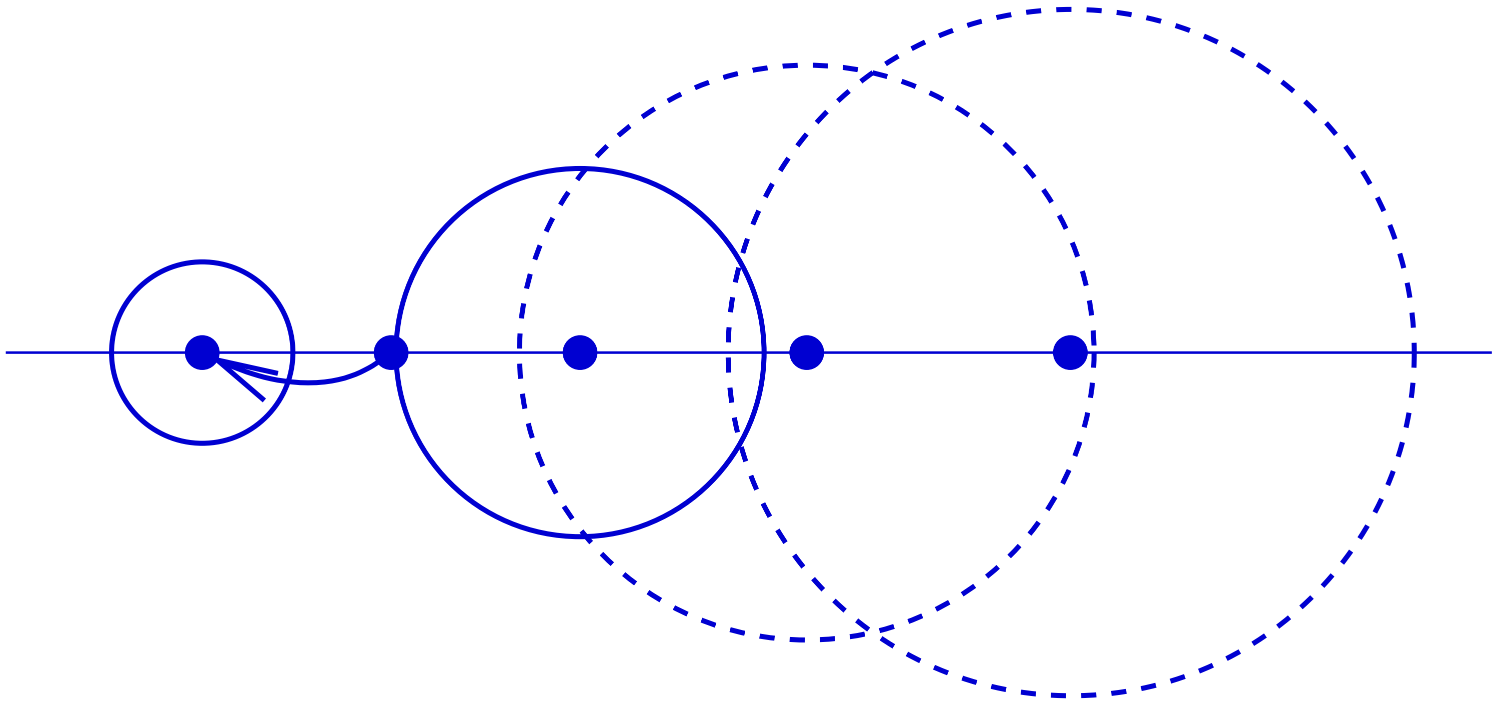
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



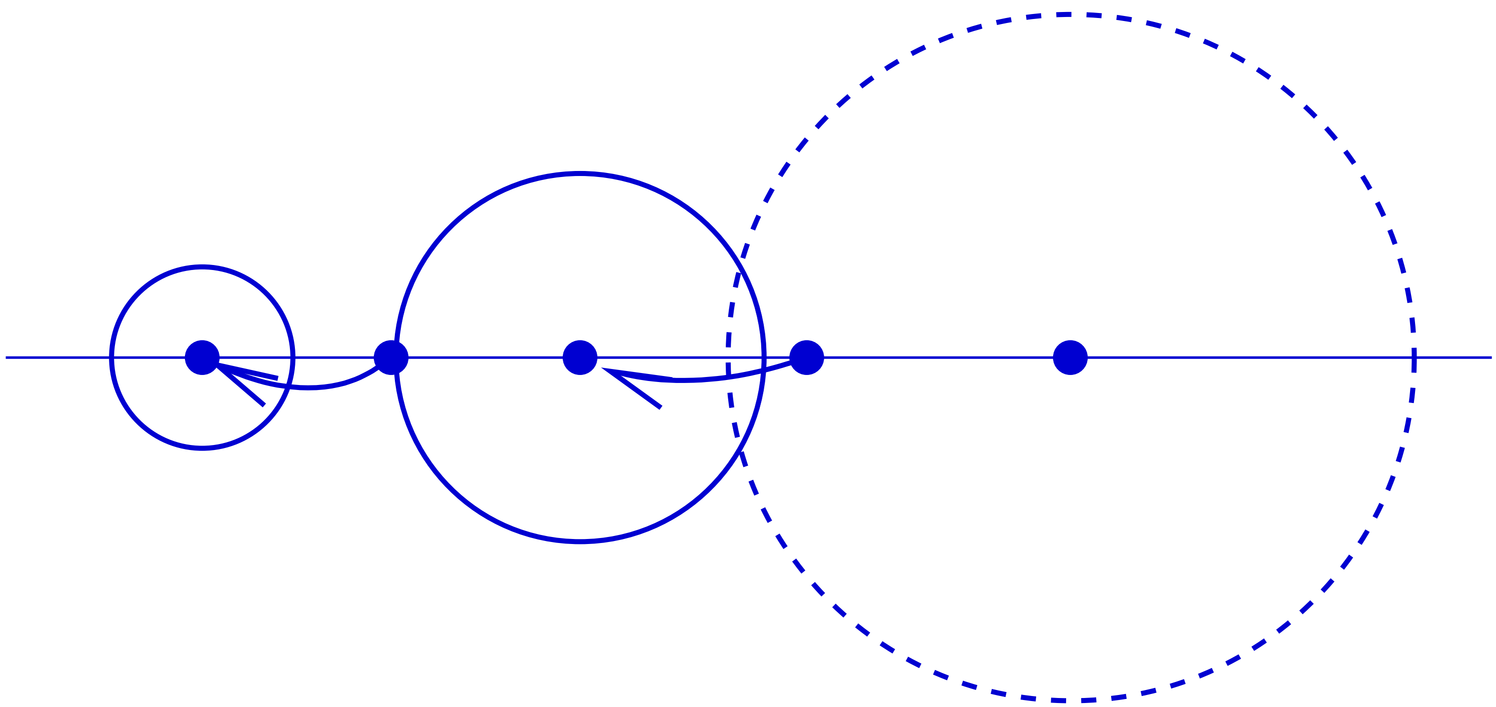
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



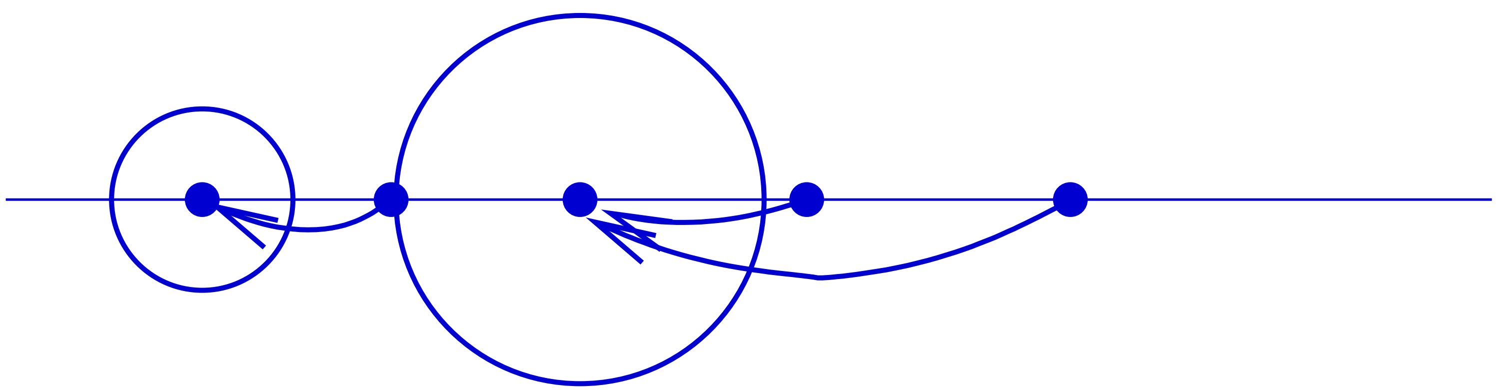
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



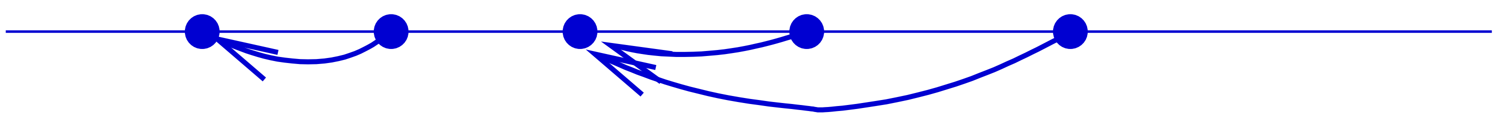
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



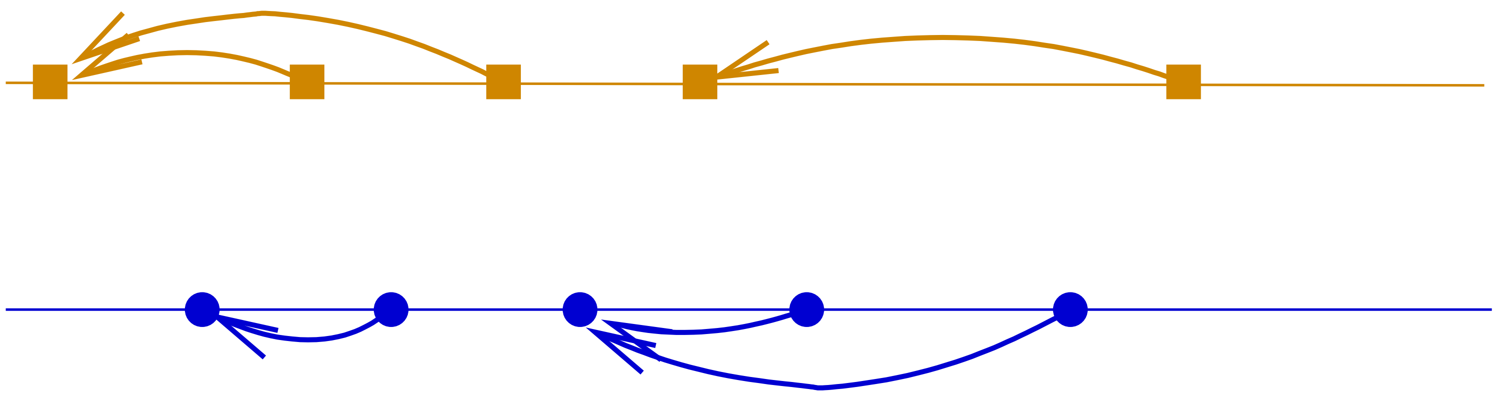
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



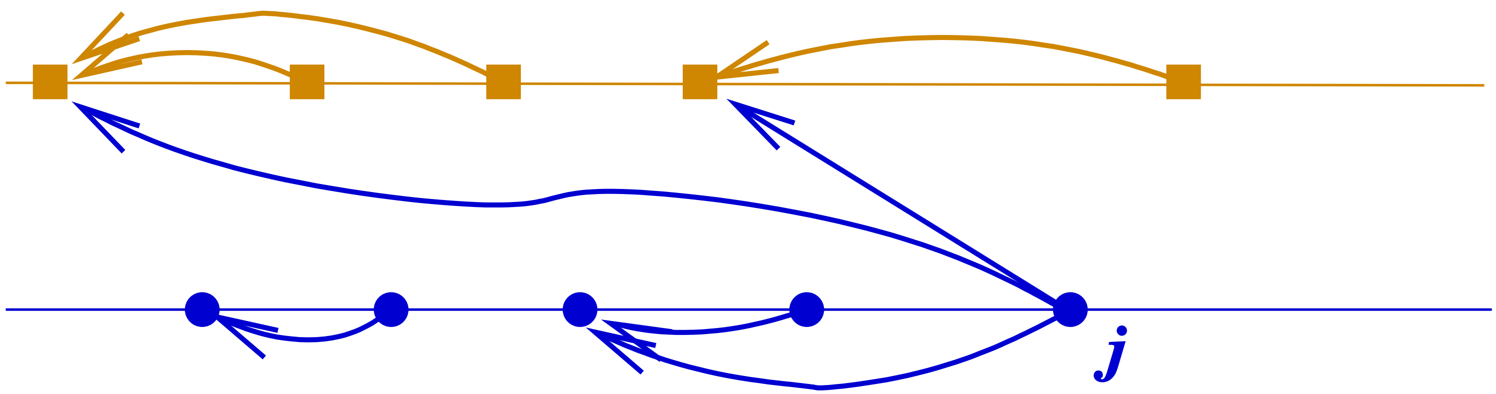
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-balls, by nondecreasing cost.



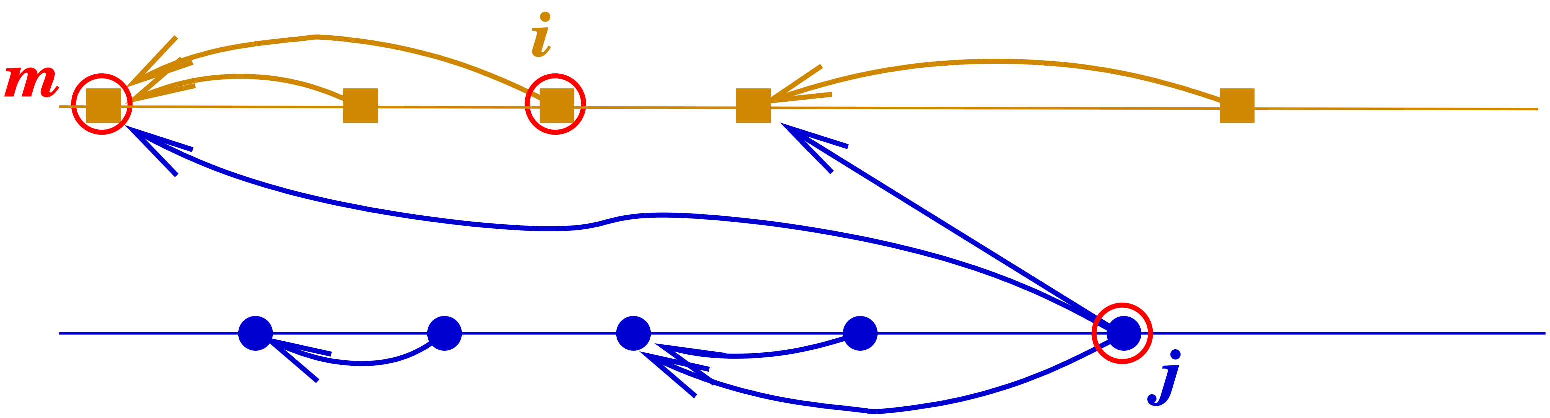
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-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: $Q_j = \{j\} \cup \{\ell'_j\} \cup \{i : i \in S \text{ and } C_i < C'_j\}$.



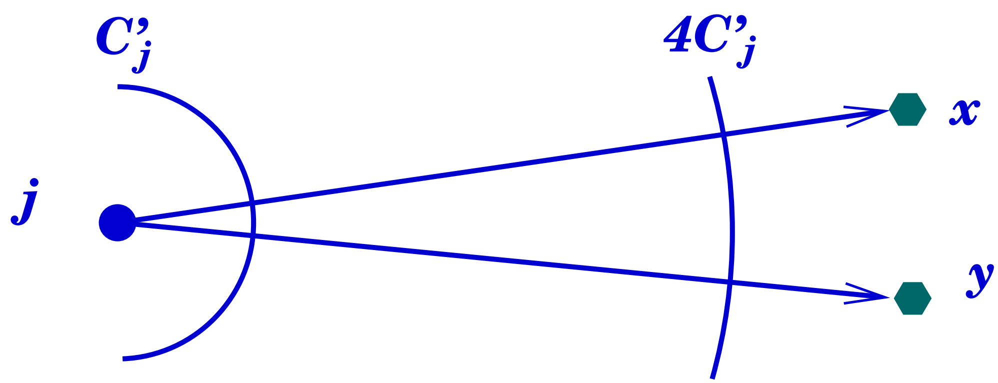
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm

- Choose leaders: nodes with disjoint β-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 $j \in \mathcal{Q}$, $P_i \cap Q_j \neq \emptyset$.



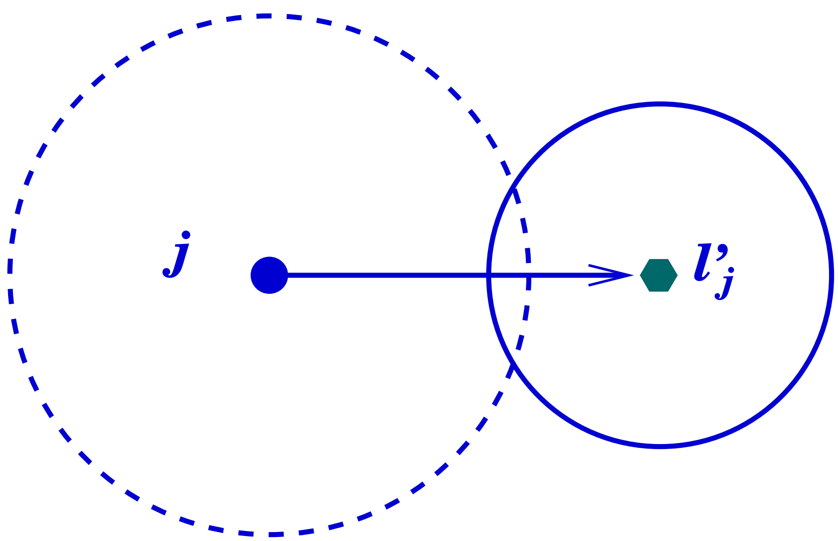
Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm Proof

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



Unicast Model with Aggregation
Uniform Interest Set / Metric — Algorithm Proof

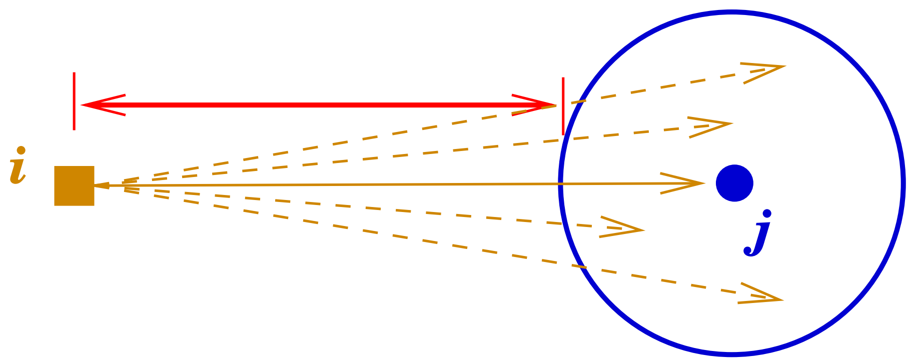
- Relative distance limits total push extent:
  For $i \in \mathcal{P}$, $\alpha > 1$, $\sum_{k \notin B_i(\alpha C_i)} x^*_{ik} \leq 1/\alpha$
- Derive Approximation Ratio.
  * Recall: $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S'$ and $C'_j \leq C_i\}$
  * Cost to i's leader ℓ_i: $2\beta C_i$



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: $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S'$ and $C'_j \leq C_i\}$
  - Cost to i's leader ℓ_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 $i \in \mathcal{P}$, $\alpha > 1$, $\displaystyle\sum_{k \notin B_i(\alpha C_i)} x^*_{ik} \leq 1/\alpha$

- Derive Approximation Ratio.
  - $*$ Recall: $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S'$ and $C'_j \leq C_i\}$
  - $*$ Cost to i's leader ℓ_i: $2\beta C_i$
  - $*$ Cost to (other) leaders S_i:

$$\begin{aligned} C_i &\geq \textstyle\sum_{j \in S_i} (d_{ij} - \alpha C'_j) \sum_{k \in B_j(\alpha C'_j)} r^*_{ijk} \\ &\geq \textstyle\sum_{j \in S_i} d_{ij} \left[1 - \frac{\alpha}{\beta}\right] \left[1 - \frac{1}{\alpha}\right] \\ &= \textstyle\frac{(\beta-\alpha)(\alpha-1)}{\alpha\beta} \sum_{j \in S_i} d_{ij}. \end{aligned}$$

  - $*$ $\alpha = 1.69$ and $\beta = 2.86$ obtains 14.57-approximation.

Conclusions and Open Problems

- Nonuniform Packet Lengths
- 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!

- Chakinala, Kumarasubramanian, and Manokaran: partial support — generous gift from Northeastern University alumnus Madhav Anand.

- Rajaraman, partial support — NSF grant IIS-0330201.

- Laing, partial support — the Mellon Foundation and the Faculty Research Awards Committee of Tufts University, while visiting Northeastern University.