The Sparsest Additive Spanner via Multiple Weighted BFS Trees

Ami Paz IRIF–CNRS & Paris Diderot University

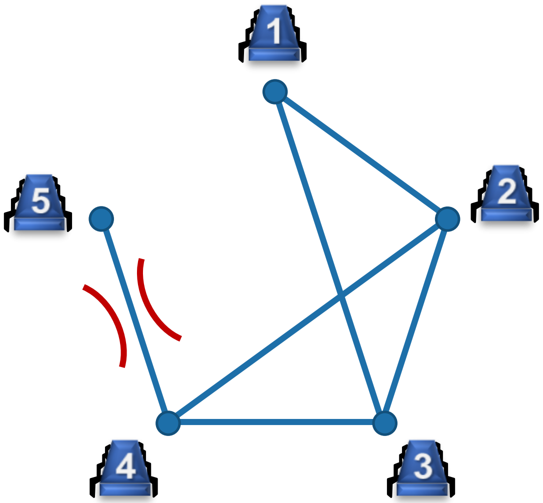
Joint work with:

Keren Censor-Hillel, Noam Ravid Technion

This project has received funding from the European Union's Horizon 2020 Research and Innovation Program under grant agreement no. 755839

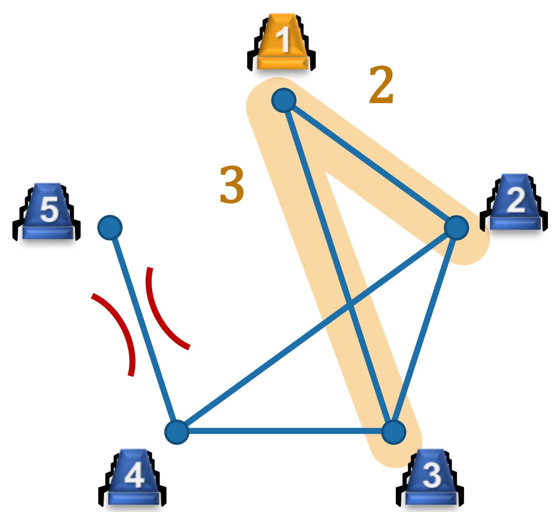
The CONGEST Model

- Communication graph on $|V| = n$ nodes
- Bounded messages, $O(\log n)$ bits
- Synchronous



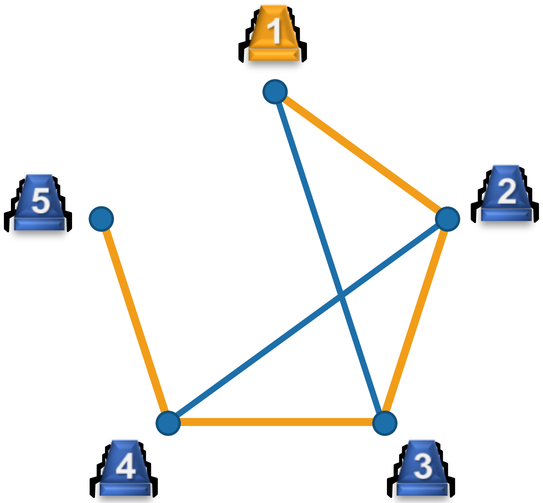
The CONGEST Model

- Communication graph on $|V| = n$ nodes
- Bounded messages, $O(\log n)$ bits
- Synchronous
- Input
  - Unique ID
  - Neighbors



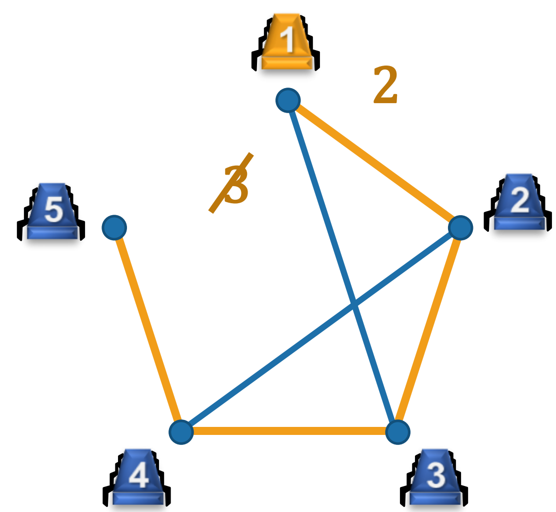
The CONGEST Model

- Communication graph on $|V| = n$ nodes
- Bounded messages, $O(\log n)$ bits
- Synchronous
- Input
  - Unique ID
  - Neighbors
- Output—subgraph
  - Neighbors in the subgraph

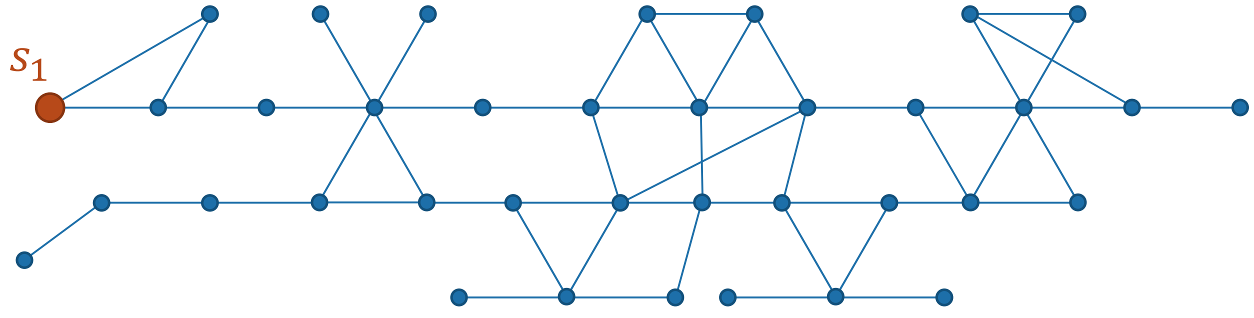


The CONGEST Model

- Communication graph on $|V| = n$ nodes
- Bounded messages, $O(\log n)$ bits
- Synchronous
- Input
  - Unique ID
  - Neighbors
- Output—subgraph
  - Neighbors in the subgraph

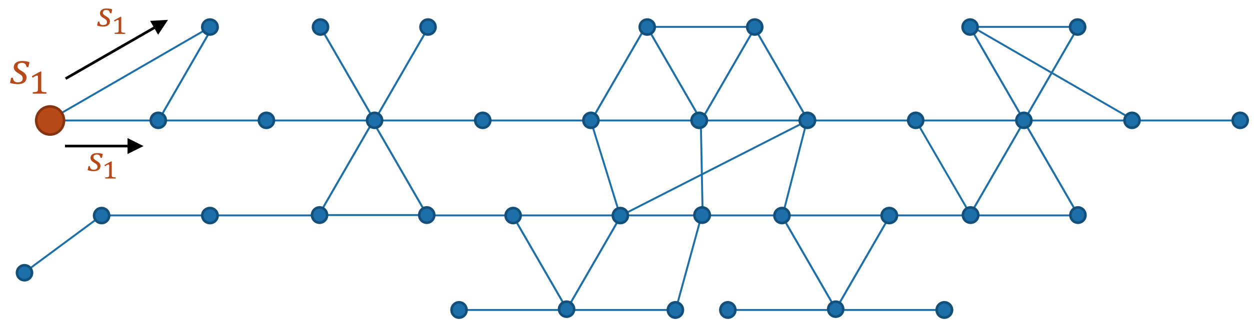


Example: Distributed BFS

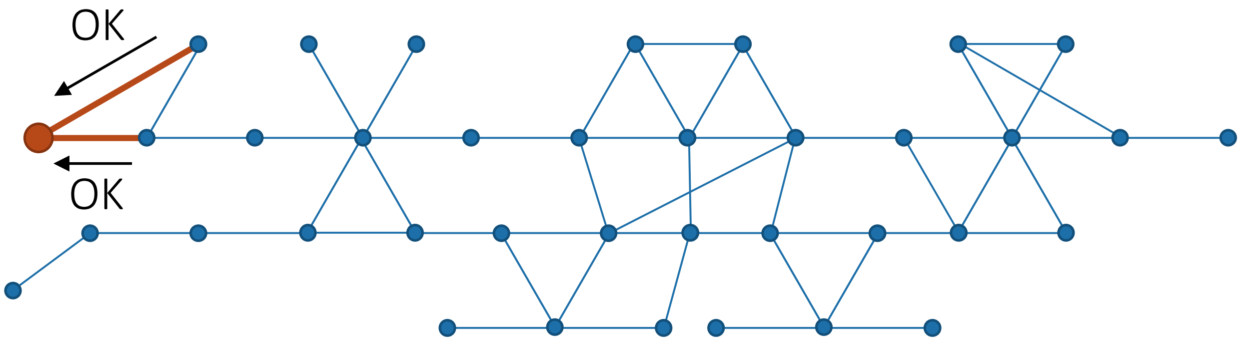


Example: Distributed BFS

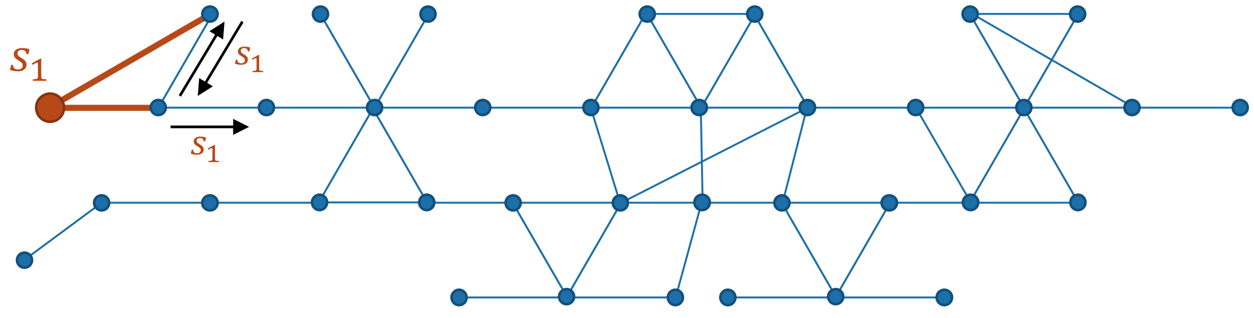
Message:
$source$



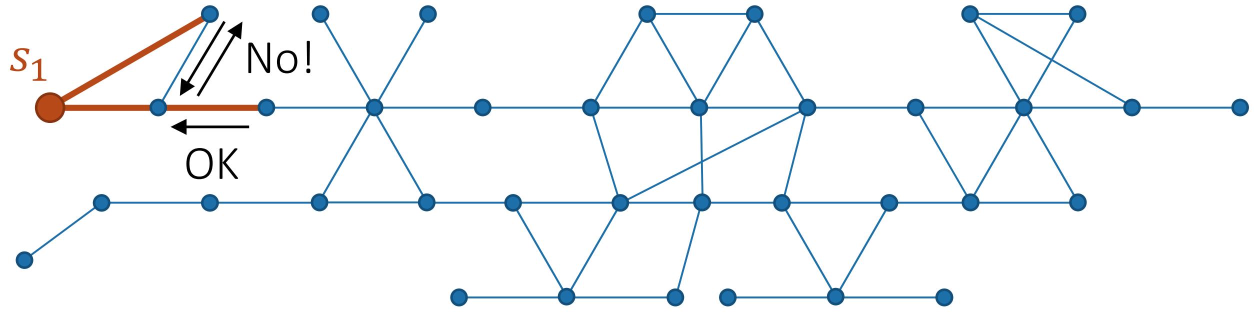
Example: Distributed BFS



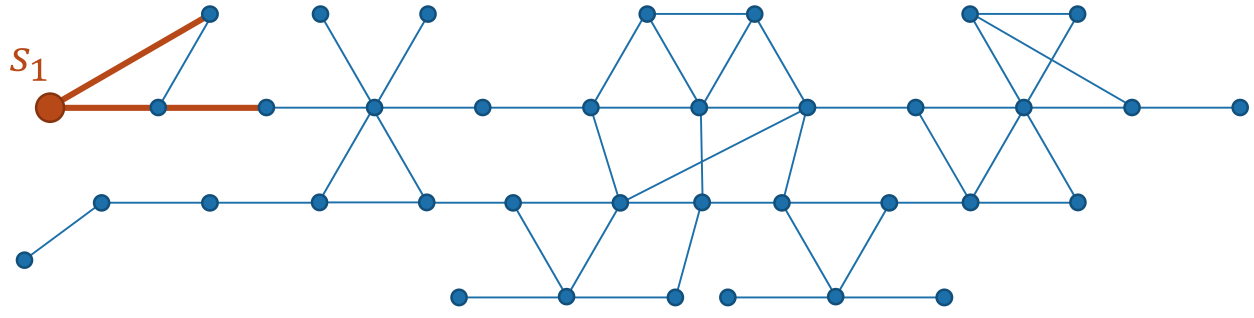
Example: Distributed BFS



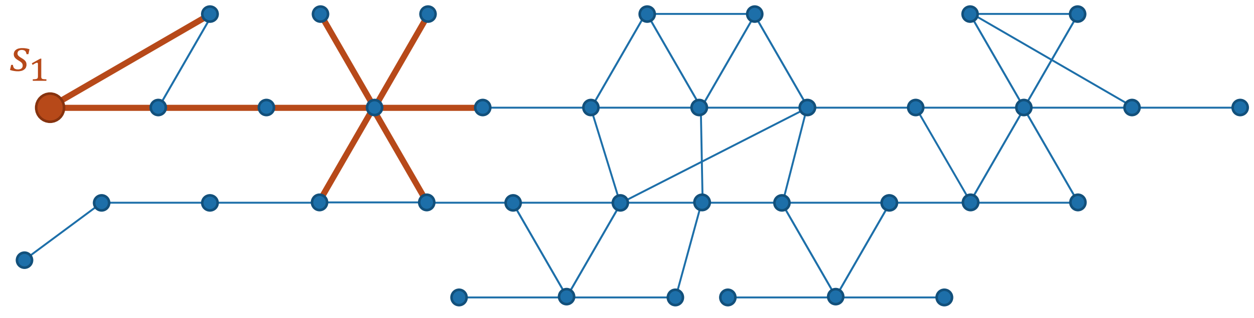
Example: Distributed BFS



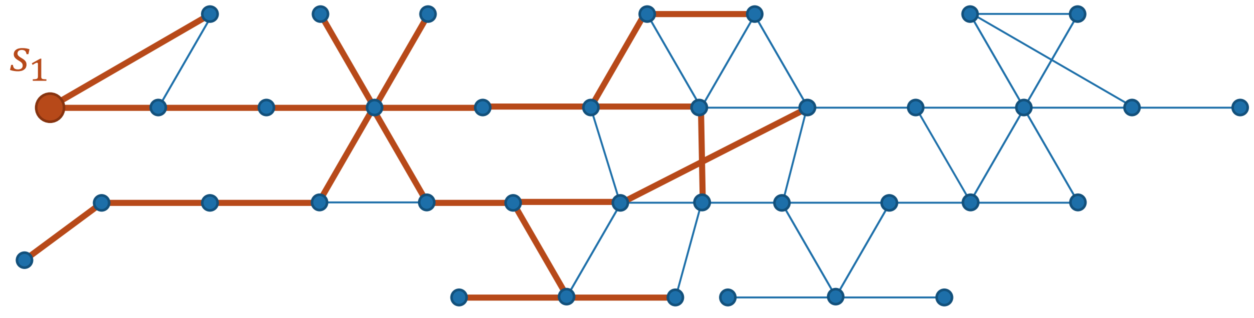
Example: Distributed BFS



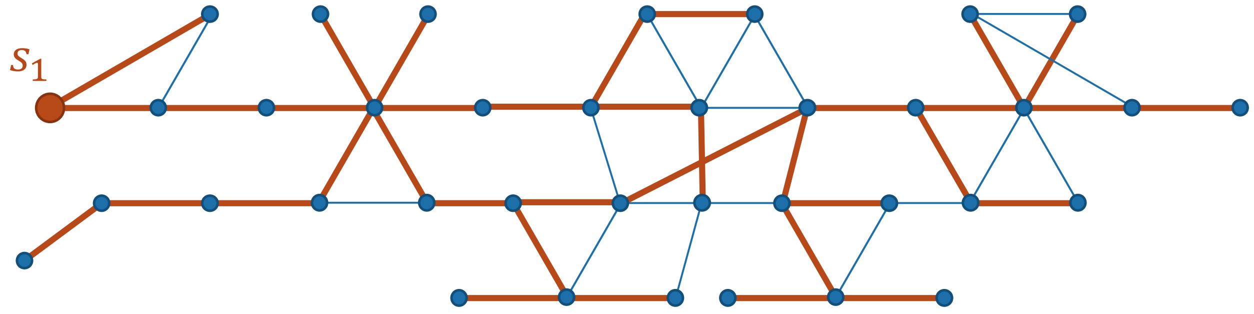
Example: Distributed BFS



Example: Distributed BFS

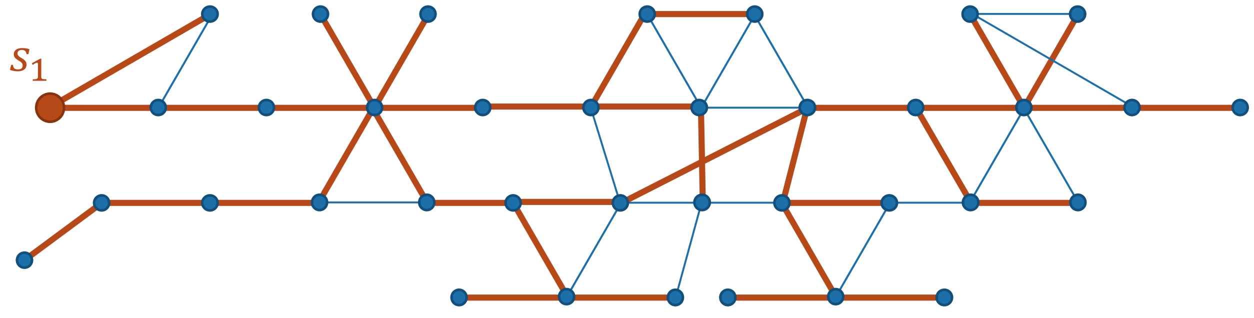


Example: Distributed BFS

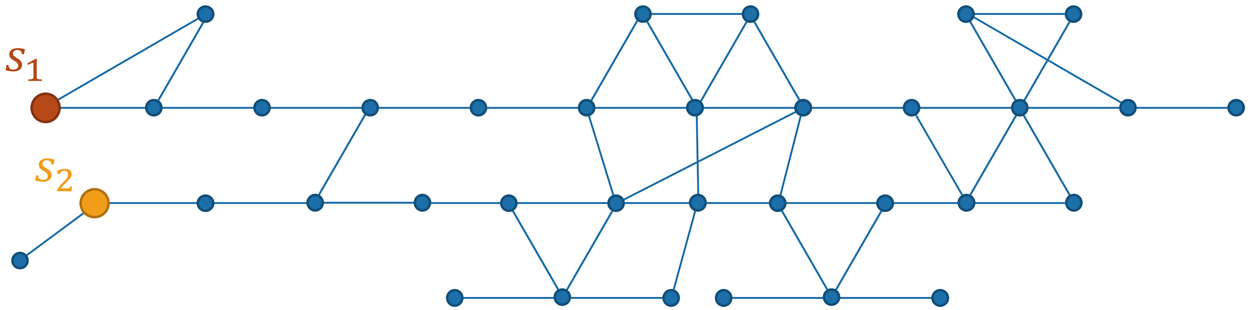


Example: Distributed BFS

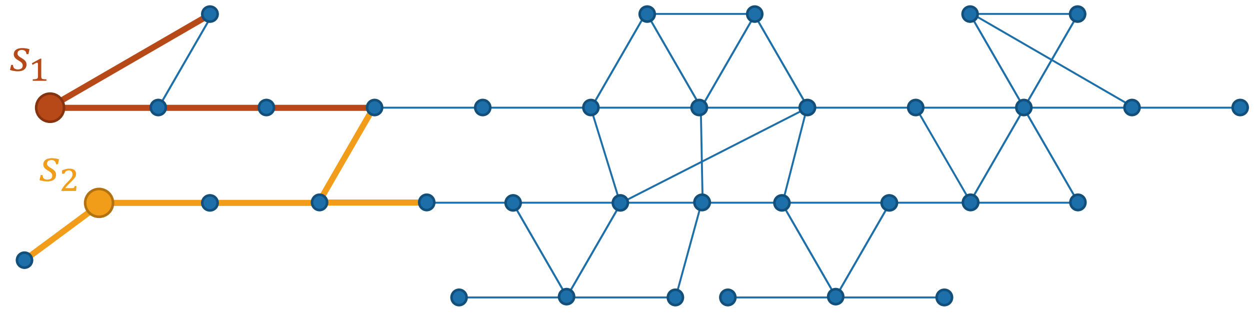
- BFS in $O(D)$ rounds



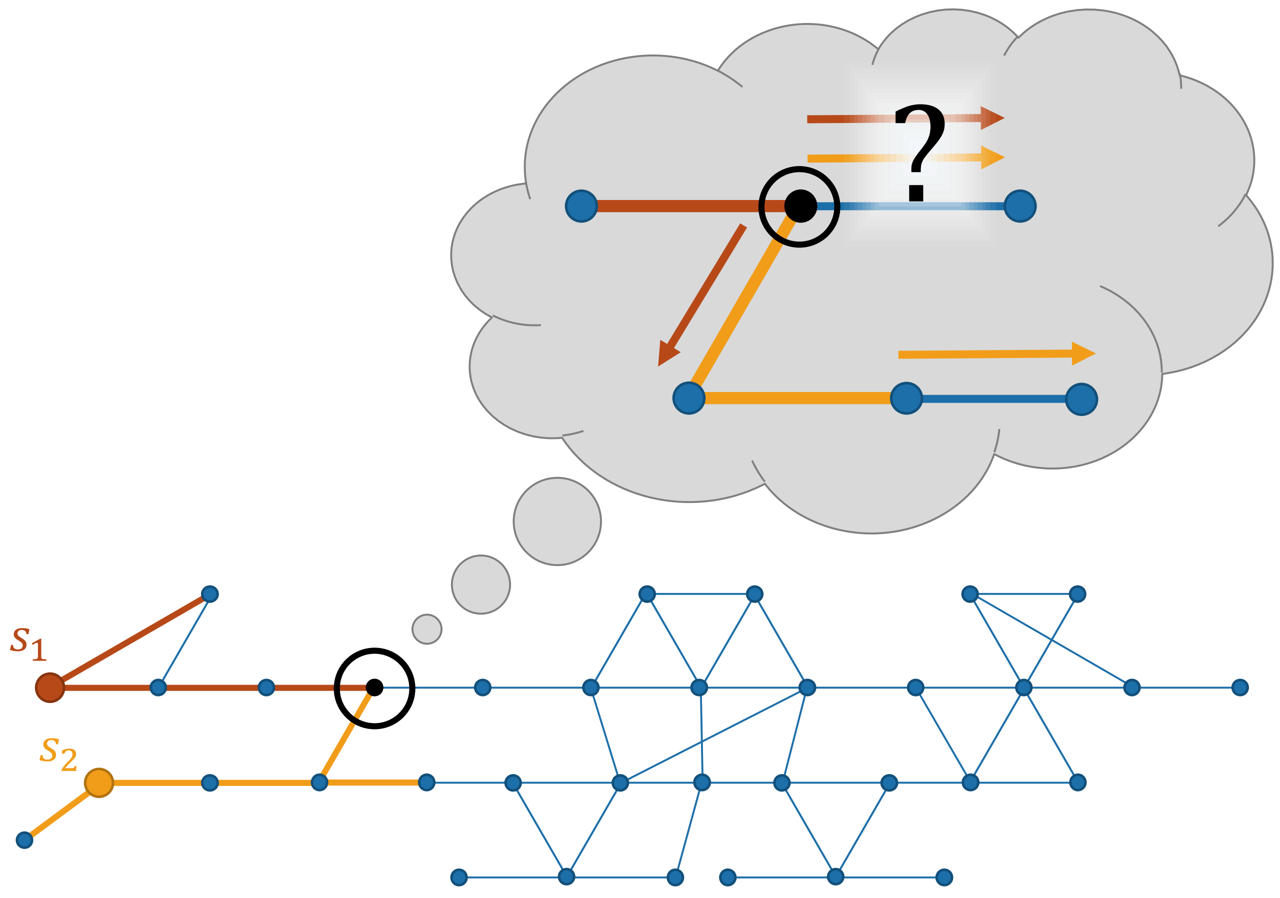
Example: Multiple BFS trees



Example: Multiple BFS trees

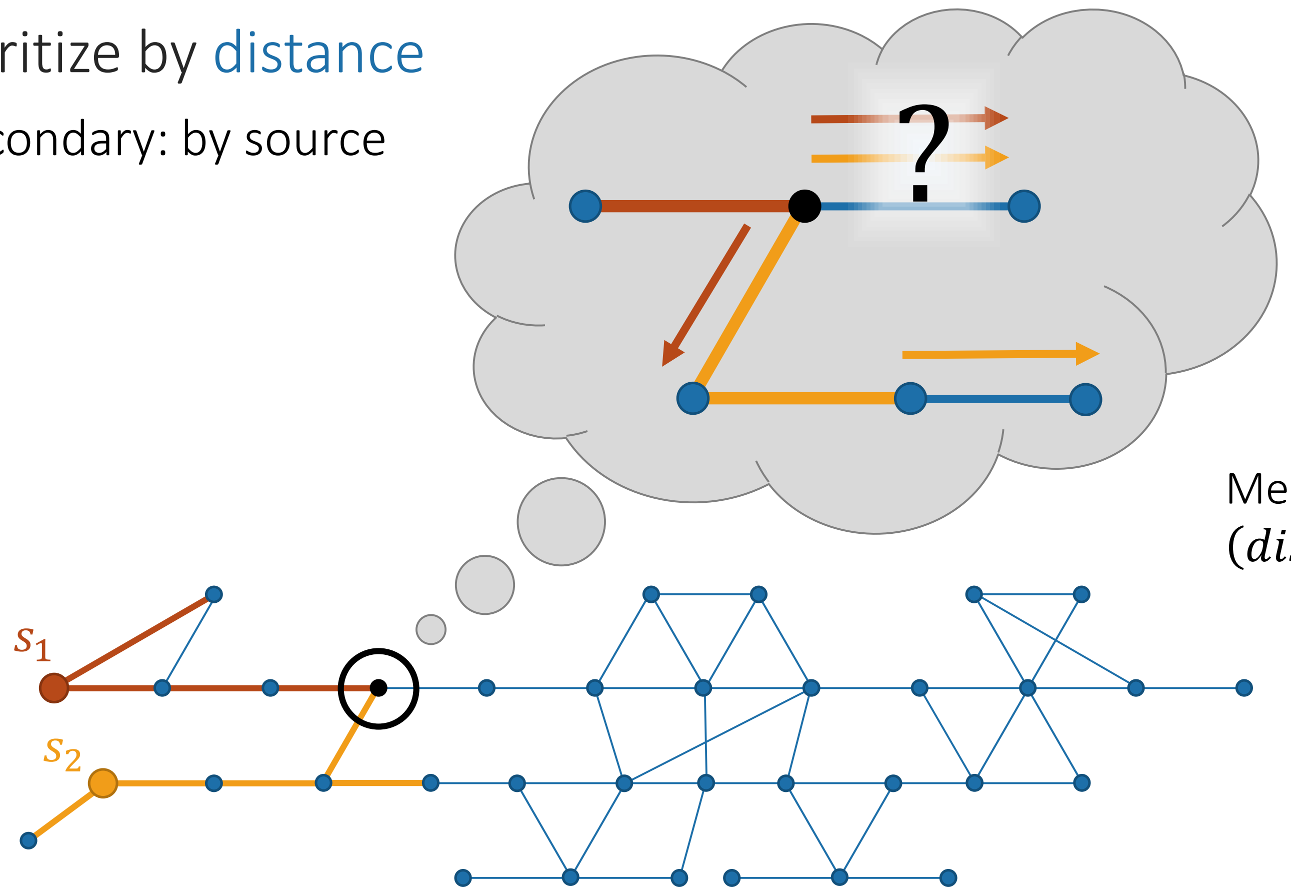


Example: Multiple BFS trees



Example: Multiple BFS trees

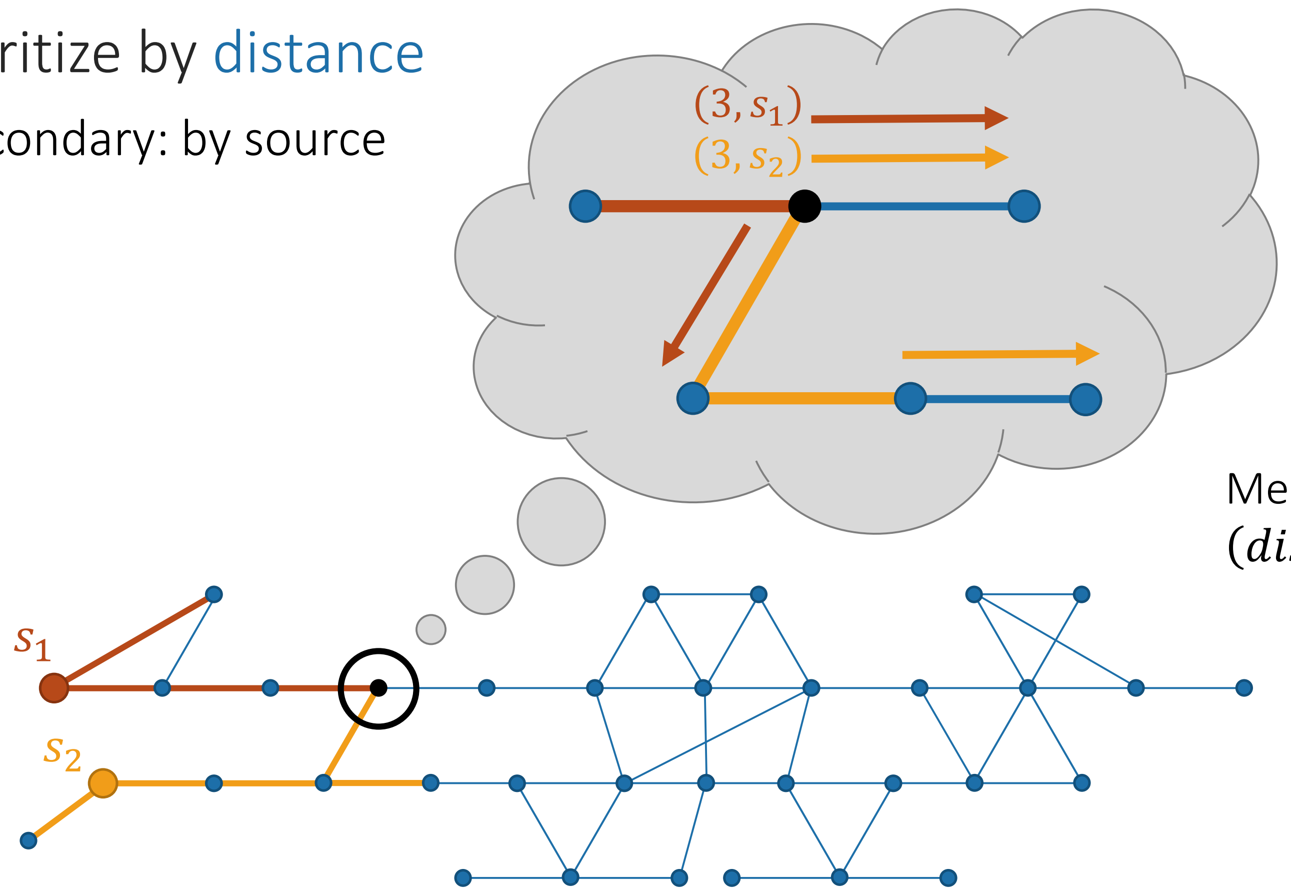
- Prioritize by distance
  - Secondary: by source



Message format:
$(dist, source)$

Example: Multiple BFS trees

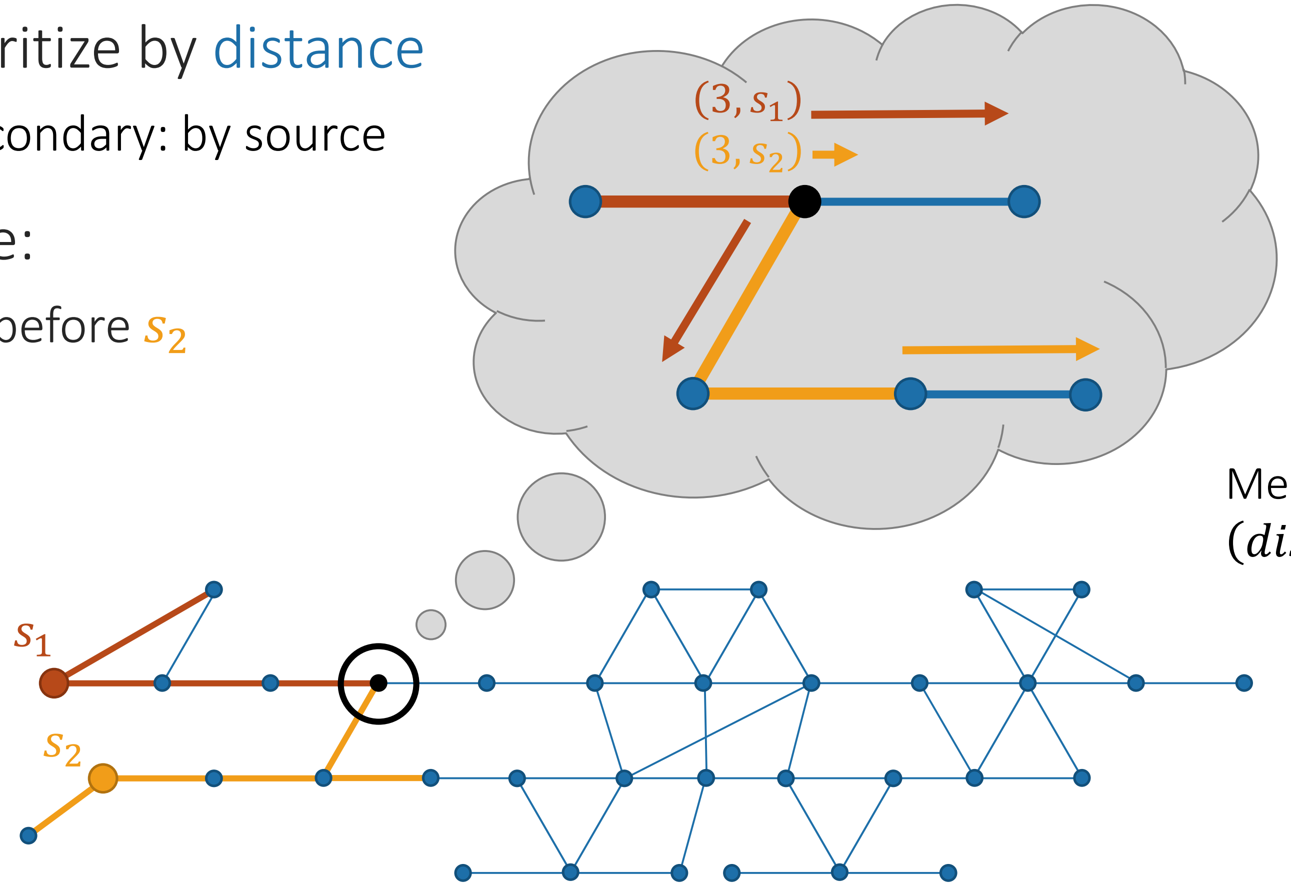
- Prioritize by distance
  - Secondary: by source



Message format:
$(dist, source)$

Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source
- Here:
  - s_1 before s_2

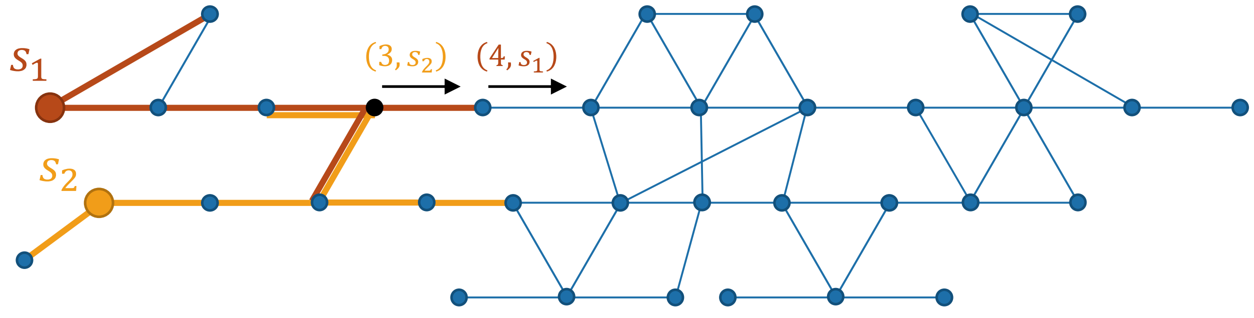


Message format:
$(dist, source)$

Example: Multiple BFS trees

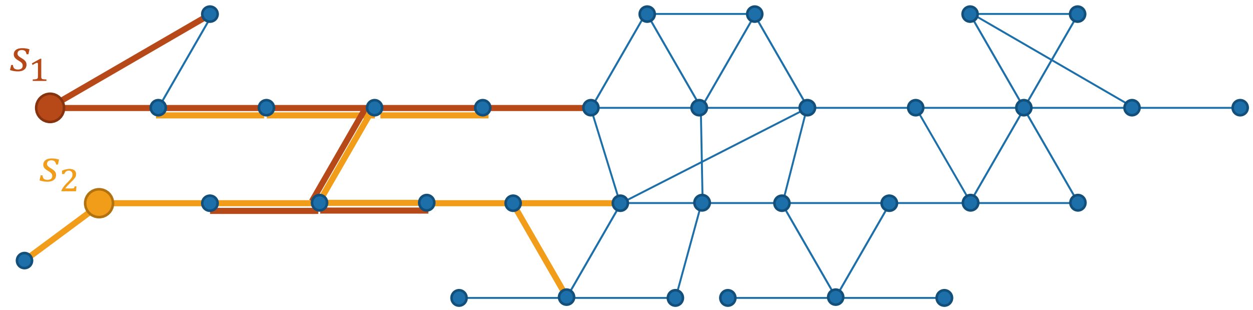
- Prioritize by distance
  - Secondary: by source

Message format: $(dist, source)$



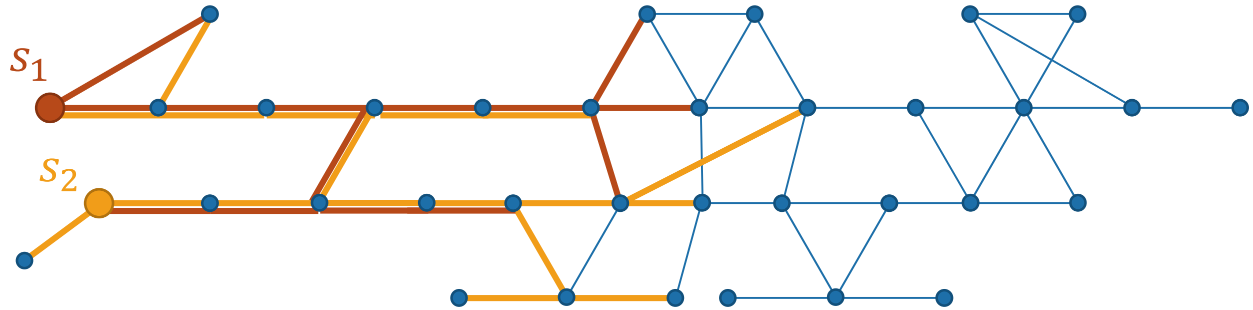
Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source



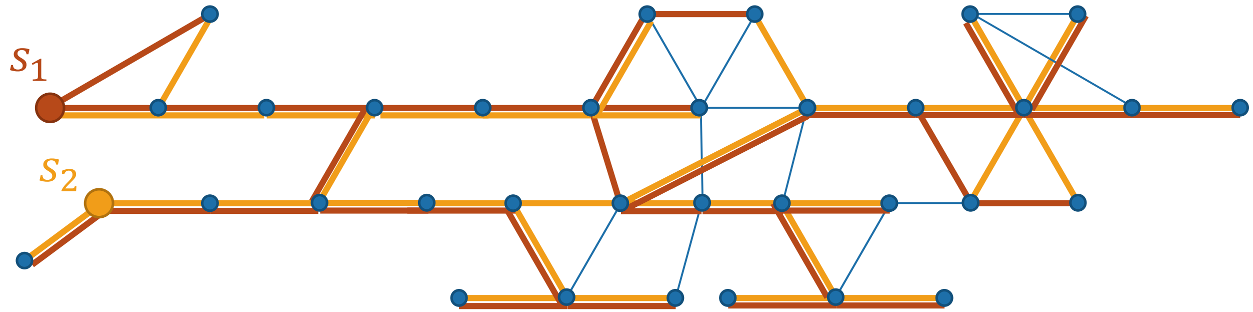
Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source



Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source



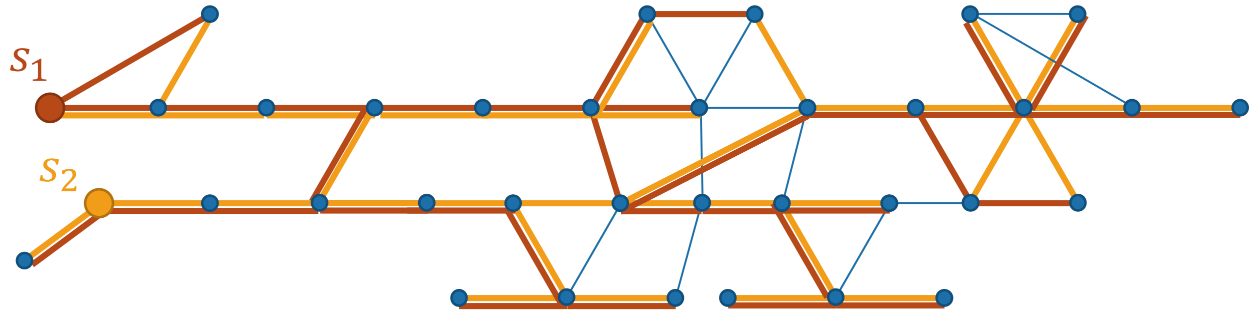
Example: Multiple BFS trees

BFS from τ sources

- Trivial: $O(\tau \cdot D)$ rounds

Theorem [LP13]

It is possible to construct BFS trees from τ sources in $O(\tau + D)$ rounds



Weighted BFS

G weighted graph, τ source

Want: a BFS tree with minimal-weight paths from s

That is: from all shortest (s, t)-paths, find the lightest

Weighted BFS

G weighted graph, τ source

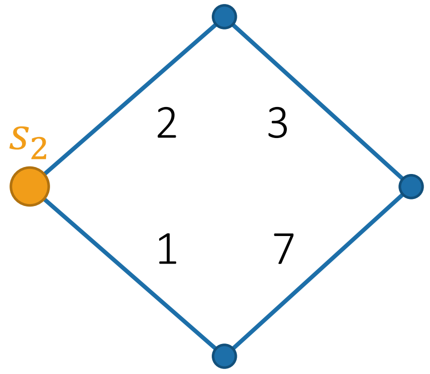
Want: a BFS tree with minimal-weight paths from s

- Is this a tree?
- Can we build it in CONGEST?
- Can we build multiple trees?

Weighted BFS

Claims:

- There is a tree with shortest-lightest paths
- It can be built in CONGEST

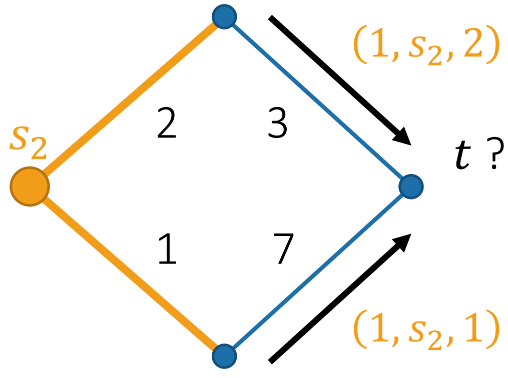


Weighted BFS

Claims:

- There is a tree with shortest-lightest paths
- It can be built in CONGEST

Message format:
$(dist, source, w\_dist)$

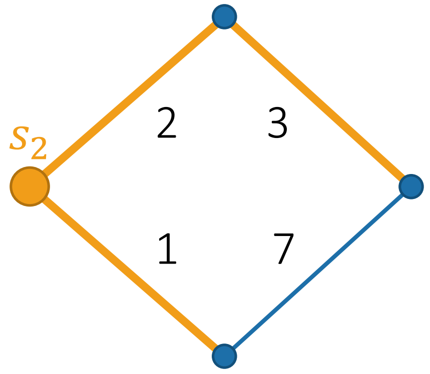


$2 + 3$, or
$1 + 7$?

Weighted BFS

Claims:

- There is a tree with shortest-lightest paths
- It can be built in CONGEST



Weighted BFS

G weighted graph, s source

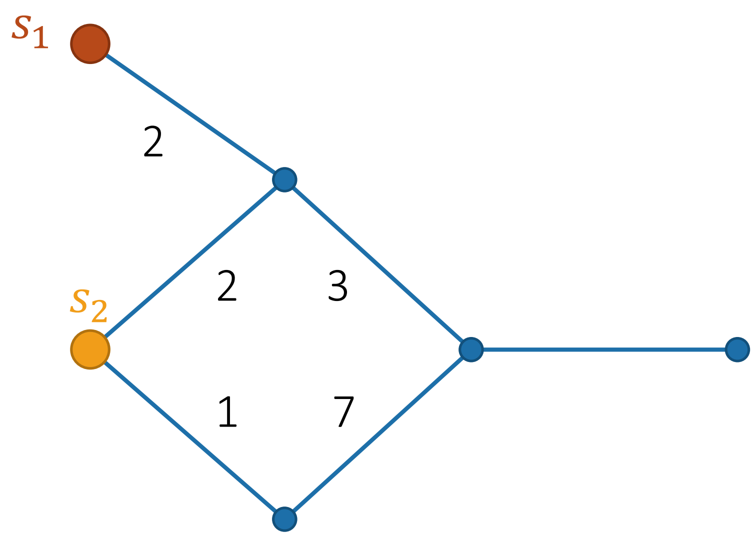
Want: a BFS tree with minimal weight paths from s

- Is this a tree? 🙂
- Can we build it in CONGEST? 🙂
- Can we build multiple trees?

Weighted BFS

Claim:

- We can be build multiple wBFS trees in CONGEST

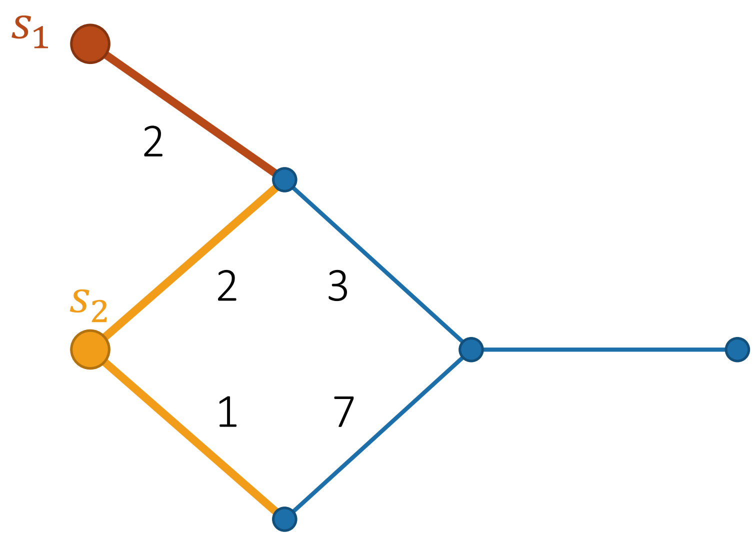


Message format:
$(dist, source, w\_dist)$

Weighted BFS

Claim:

- We can be build multiple wBFS trees in CONGEST

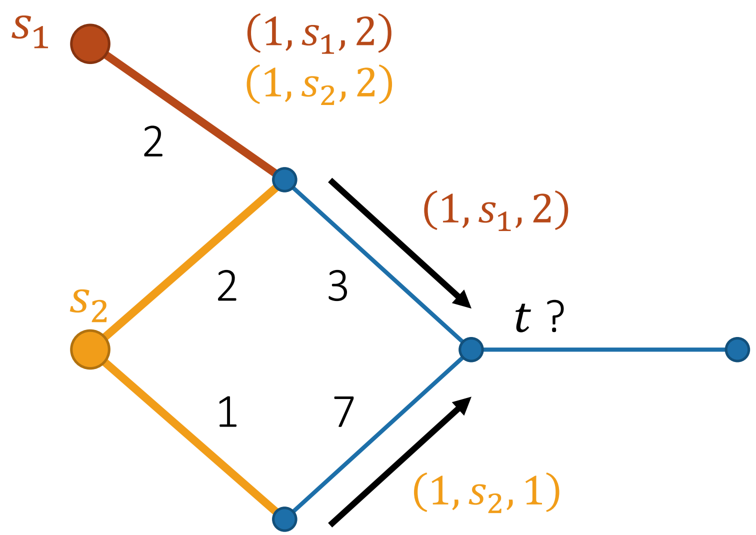


Message format:
$(dist, source, w\_dist)$

Weighted BFS

Claim:

- We can be build multiple wBFS trees in CONGEST

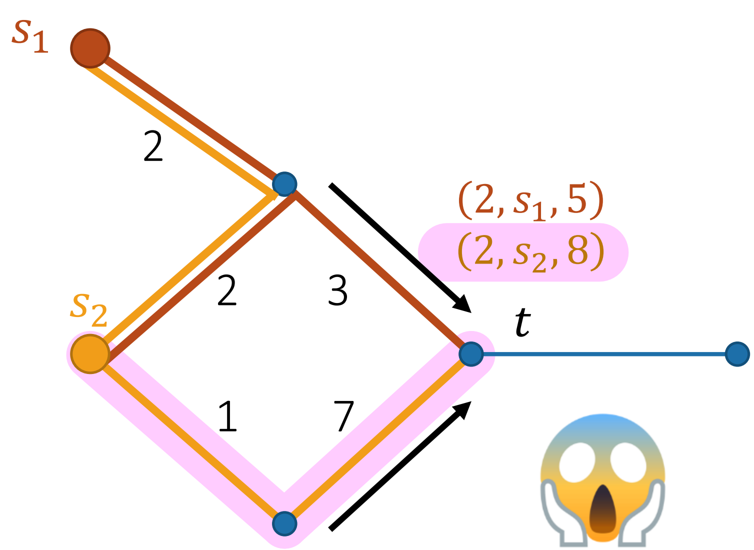


Message format:
$(dist, source, w\_dist)$

Weighted BFS

Claim:

- We can be build multiple wBFS trees in CONGEST

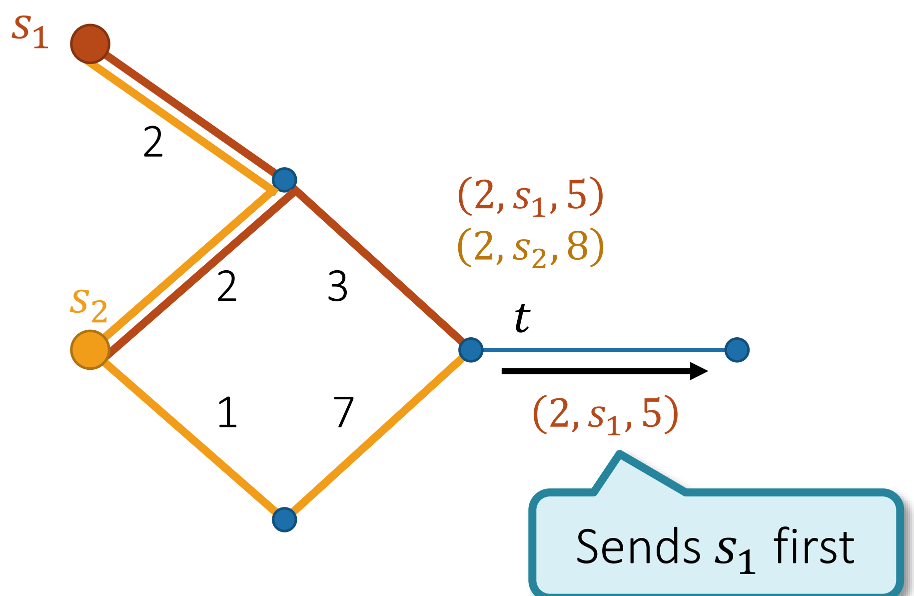


Message format:
$(dist, source, w\_dist)$

Weighted BFS

Claim:

- We can be build multiple wBFS trees in CONGEST

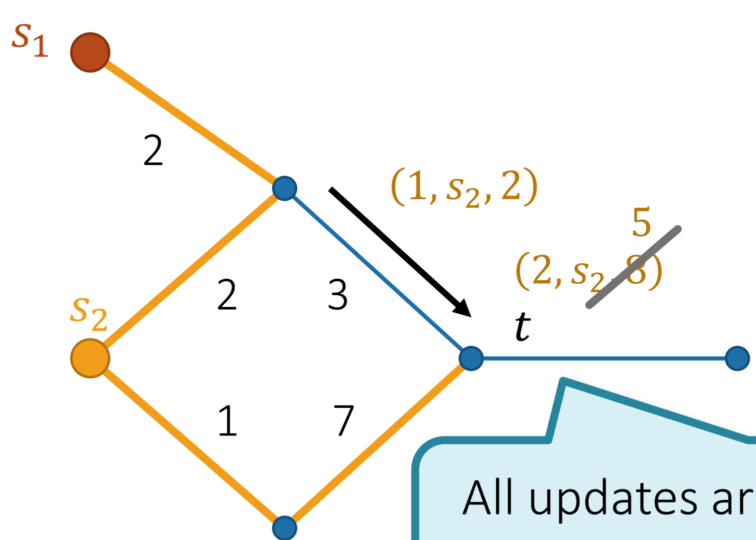


Message format:
$(dist, source, w\_dist)$

Weighted BFS

Claim:

- We can be build multiple wBFS trees in CONGEST



Message format:
$(dist, source, w\_dist)$

All updates arrive before t sends (nontrivial)

Weighted BFS

G weighted graph, s source

Want: a BFS tree with minimal weight paths from s

- Is this a tree? 🙂
- Can we build it in CONGEST? 🙂
- Can we build multiple trees? 🙂

Multiple Weighted BFS trees

Weighted BFS from τ sources

Theorem (New)

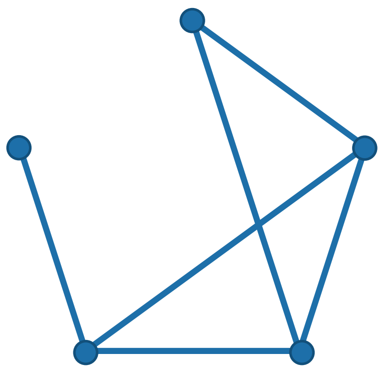
It is possible to construct weighted BFS trees from τ sources in $O(\tau + D)$ rounds

Spanners

A graph G on n nodes

Want: a subgraph H on the same nodes, that

- Approximately preserves distances
- Sparse

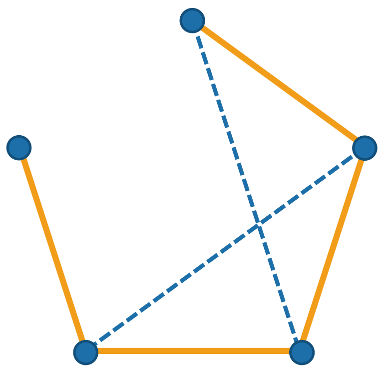


Spanners

A graph G on n nodes

Want: a subgraph H on the same nodes, that

- Approximately preserves distances
- Sparse



This talk:

only additive all-pairs spanners

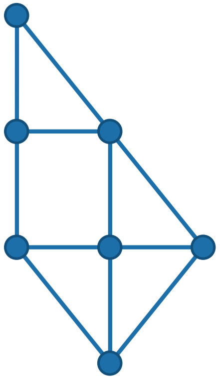
Spanners

A $(+\beta)$-spanner of G is a subgraph H

on the same nodes, s.t.

- for all $(u, v) \in V \times V$:

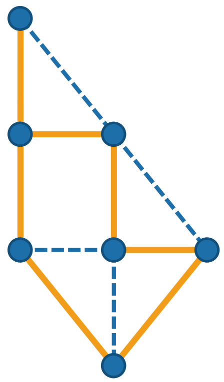
  $\text{dist}_H(u, v) \leq \text{dist}_G(u, v) + \beta$



Spanners

A $(+\beta)$-spanner of G is a subgraph H on the same nodes, s.t.

- for all $(u, v) \in V \times V$:
  $\mathrm{dist}_H(u, v) \leq \mathrm{dist}_G(u, v) + \beta$

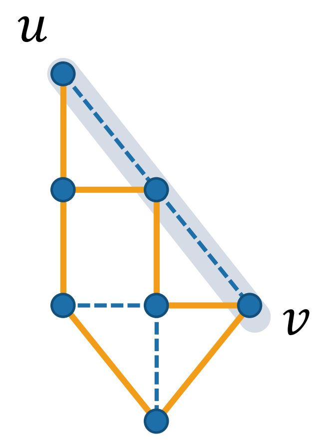


Spanners

A $(+\beta)$-spanner of G is a subgraph H on the same nodes, s.t.

- for all $(u, v) \in V \times V$:

  $\text{dist}_H(u, v) \leq \text{dist}_G(u, v) + \beta$

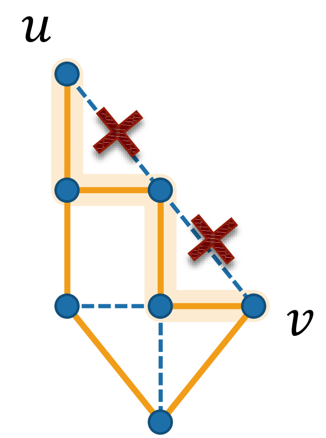


Spanners

A $(+\beta)$-spanner of G is a subgraph H on the same nodes, s.t.

- for all $(u, v) \in V \times V$:

  $\text{dist}_H(u, v) \le \text{dist}_G(u, v) + \beta$



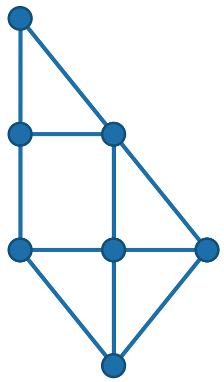
(+2)-spanner

Applications

- Synchronizers [Awe85,PU89]
- Information dissemination [CHHKM12]
- Compact routing schemes [PU89,TZ01,Che13]
- And many more…

Sequential Spanners

- Constructions
  - (+2): $O\left(n^{3/2}\right)$ edges [ACIM99]
  - (+4): $\tilde{O}\left(n^{7/5}\right)$ edges [Che13]
  - (+6): $O\left(n^{4/3}\right)$ edges [BKMP10]



- Lower bound
  - Any: $n^{4/3}/2^{\Omega\left(\sqrt{\log n}\right)}$ edges [AB16]

Goal:
Networks that build their own spanners

Distributed Additive Spanners

Spanner	Number of edges
	Sequential
$(+2)$-spanner	$O(n^{3/2})$ [ACIM99]
$(+4)$-spanner	$\tilde{O}(n^{7/5})$ [Che13]
$(+6)$-spanner	$O(n^{4/3})$ [BKMP10]
$(+8)$-spanner	
$(+?)$-spanner	

Optimal

Distributed Additive Spanners

Spanner	Number of edges	
	Sequential	Distributed
$(+2)$-spanner	$O\left(n^{3/2}\right)$ [ACIM99]	$\tilde{O}\left(n^{3/2}\right)$ [LP13]
$(+4)$-spanner	$\tilde{O}\left(n^{7/5}\right)$ [Che13]	$\tilde{O}\left(n^{7/5}\right)$ [CH+17]
$(+6)$-spanner	$O\left(n^{4/3}\right)$ [BKMP10]	
$(+8)$-spanner		$\tilde{O}\left(n^{15/11}\right)$ [CH+17]
$(+?)$-spanner		$O\left(n^{4/3}\right)$ (???)

Distributed Additive Spanners

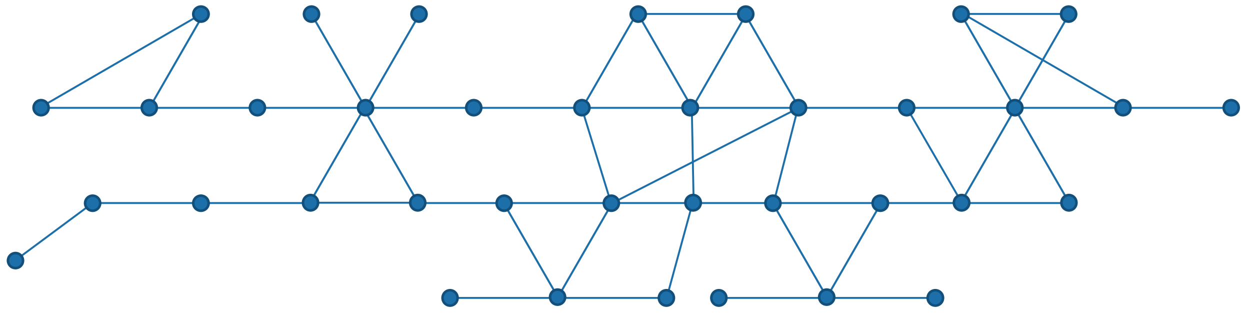
Spanner	Number of edges	
	Sequential	Distributed
(+2)-spanner	$O(n^{3/2})$ [ACIM99]	$\tilde{O}(n^{3/2})$ [LP13]
(+4)-spanner	$\tilde{O}(n^{7/5})$ [Che13]	$\tilde{O}(n^{7/5})$ [CH+17]
(+6)-spanner	$O(n^{4/3})$ [BKMP10]	$\tilde{O}(n^{4/3})$
(+8)-spanner		$\tilde{O}(n^{15/11})$ [CH+17]
(+?)-spanner		$O(n^{4/3})$ (???)

New!

Spanner Construction

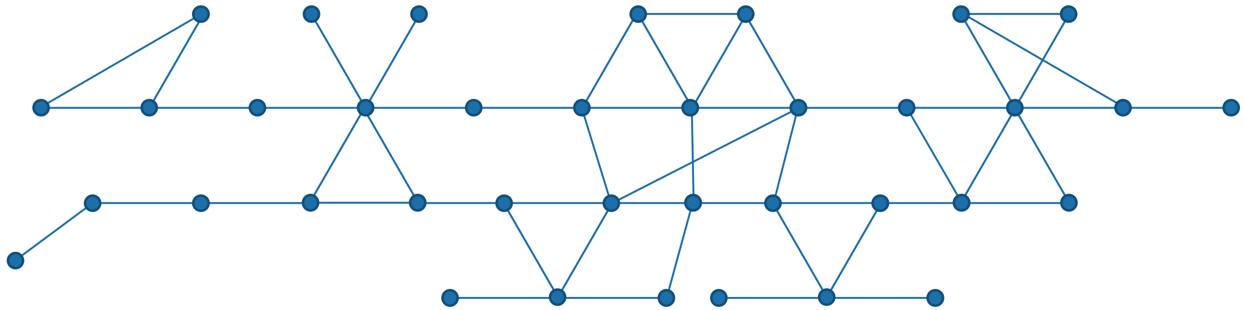
Two phases:

- Clustering
- Path buying



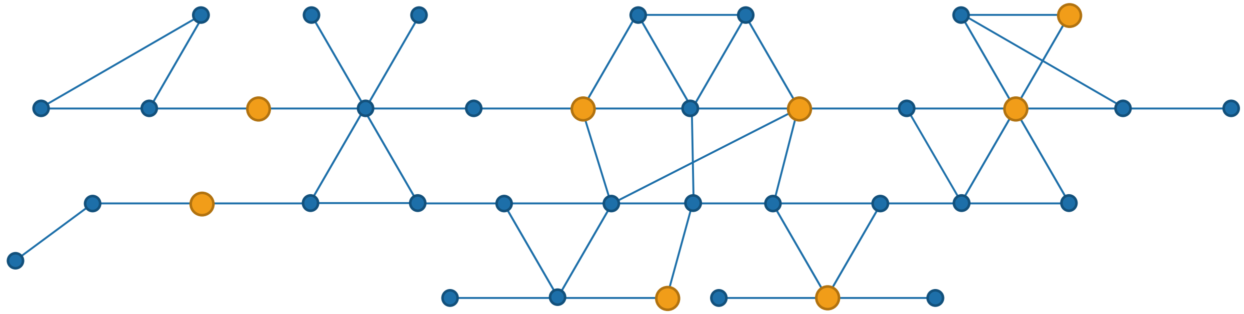
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes



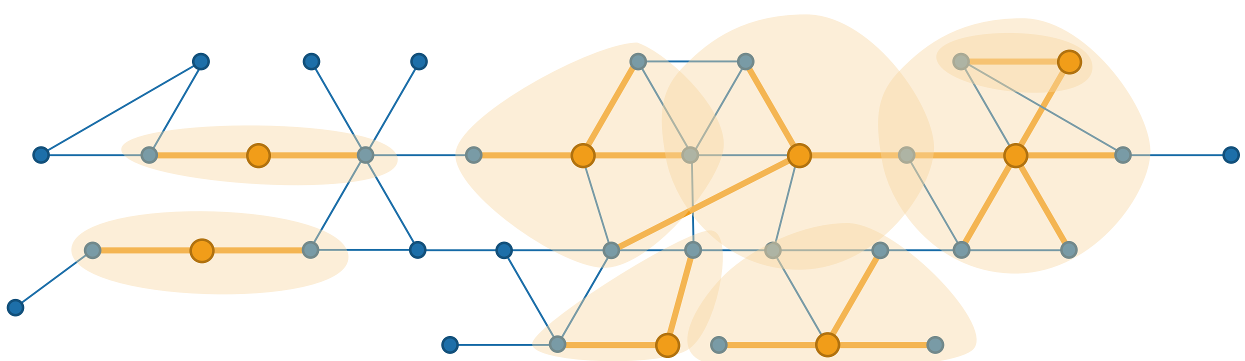
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes



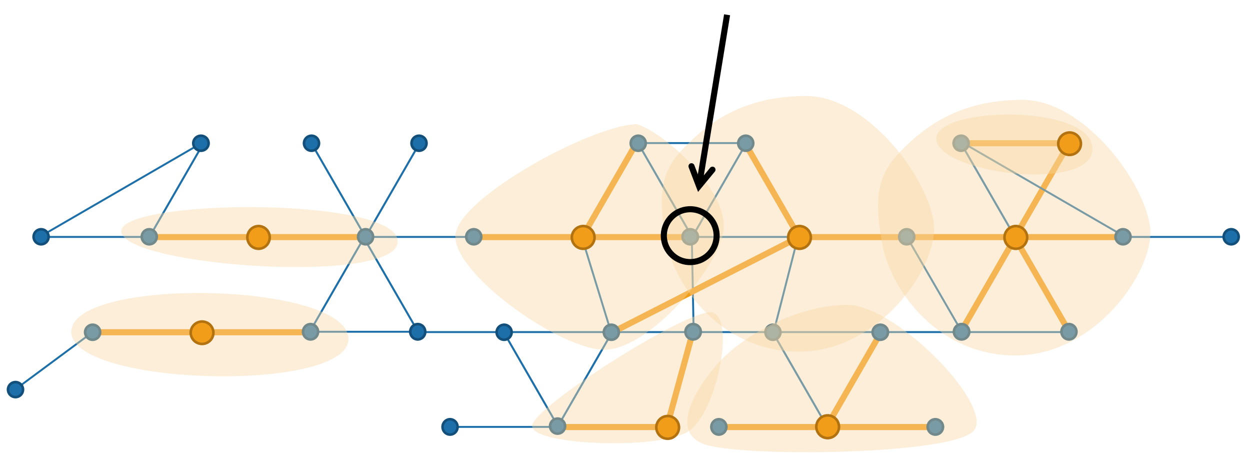
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes



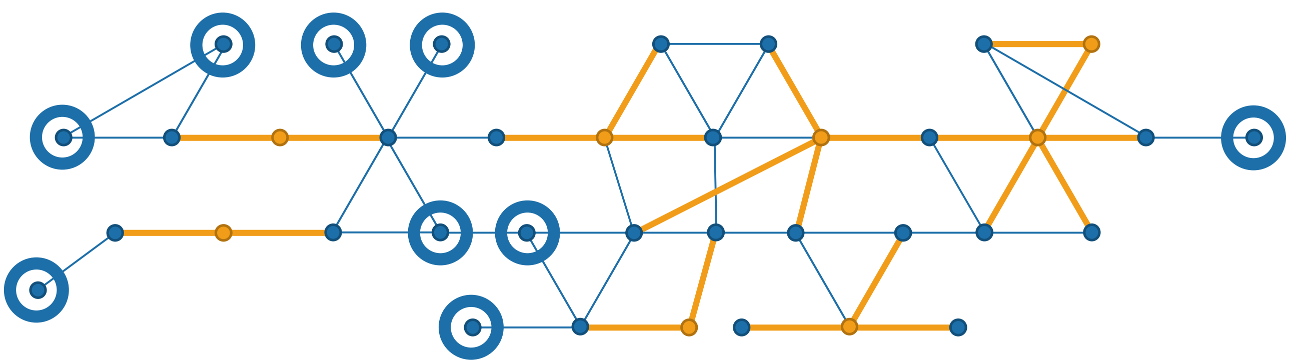
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes



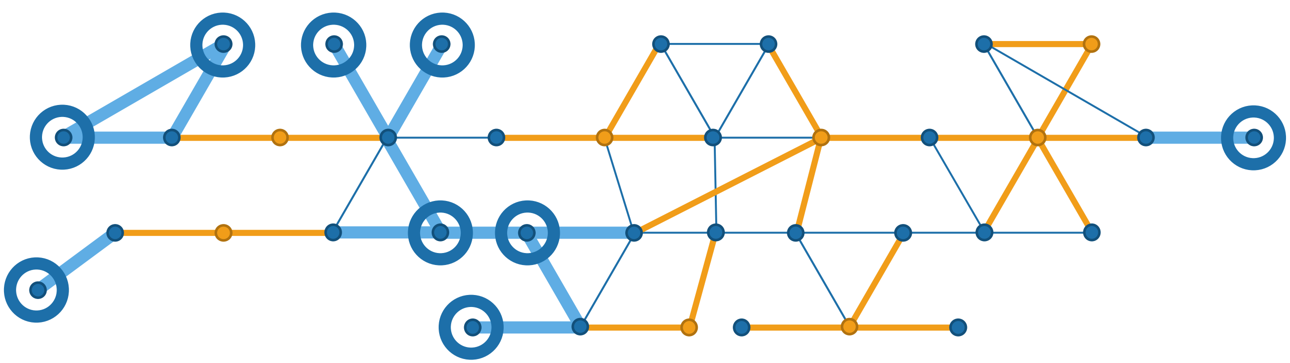
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes



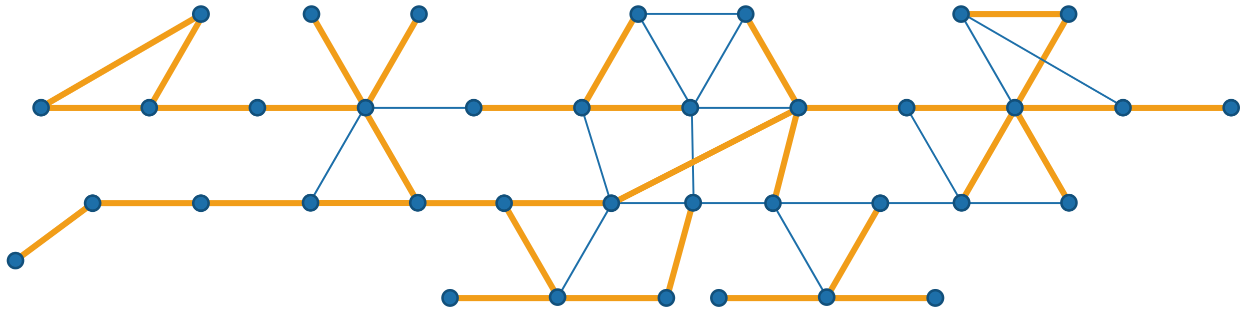
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes



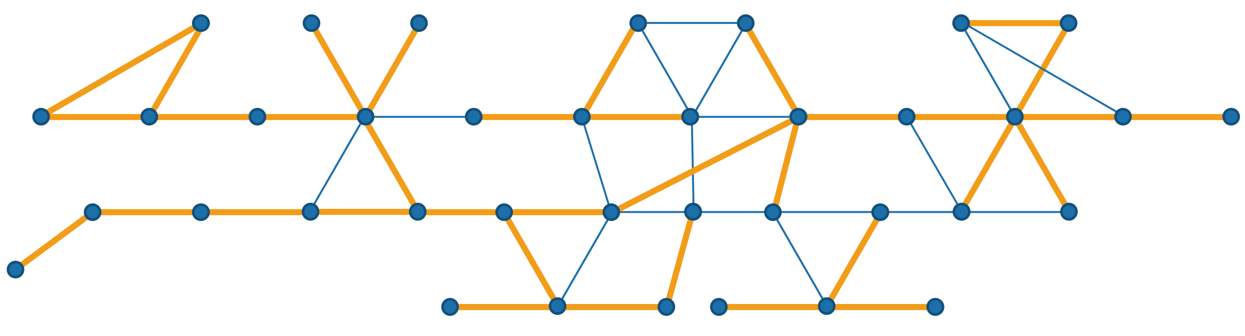
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes



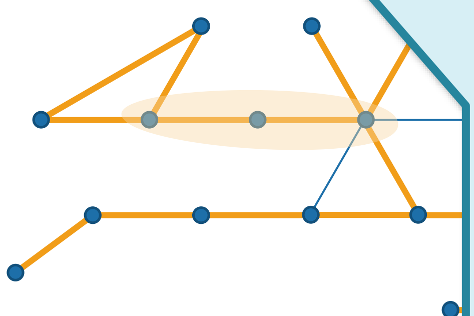
Path Buying

- For $k = 1,2,4,8,\ldots,n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges



Path Buying

- For $k = 1,2,4,8,\ldots,n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges

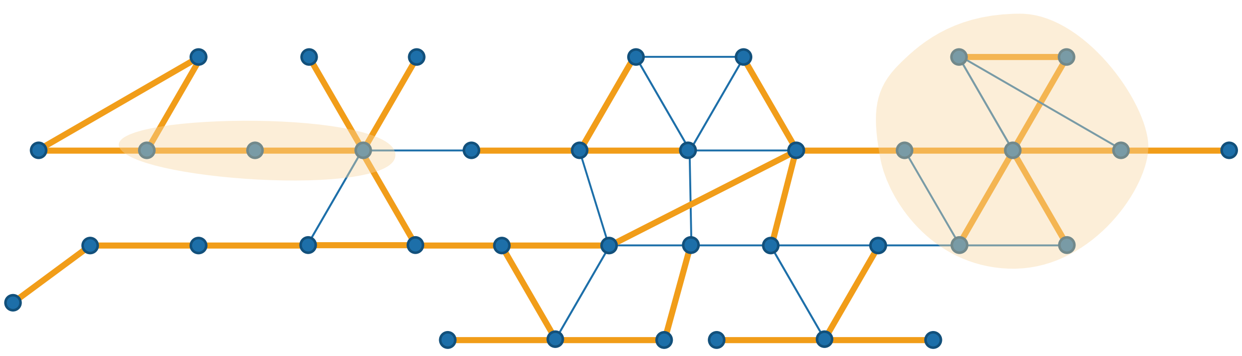


That is, for each (c_i, C_j):

1. $A \leftarrow \emptyset$
2. For each $v \in C_j$, if there is a (c_i, v)-path that is shortest and misses $\leq 2k$ edges add one to A
3. If $A \neq \emptyset$, add a shortest path from A

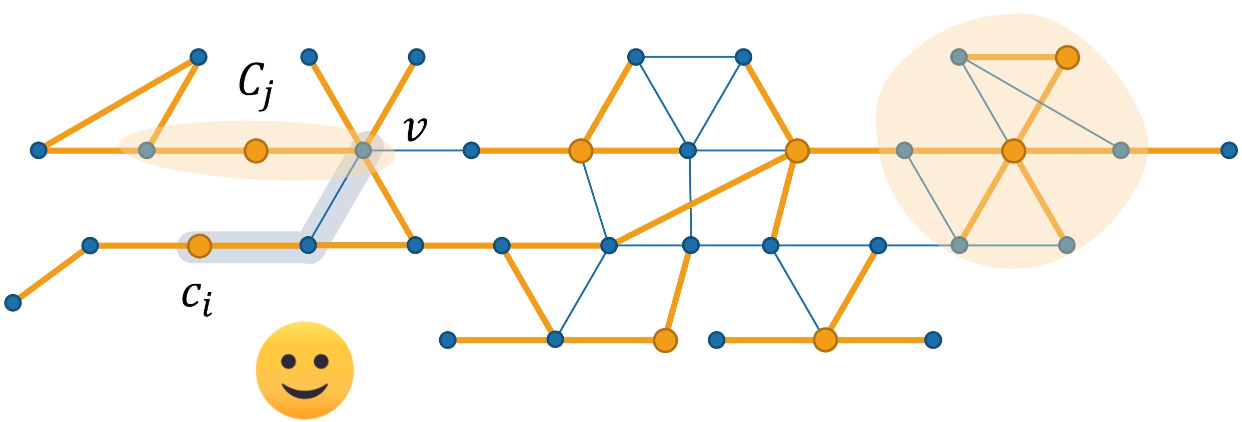
Path Buying

- For $k = 1,2,4,8,\ldots,n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges



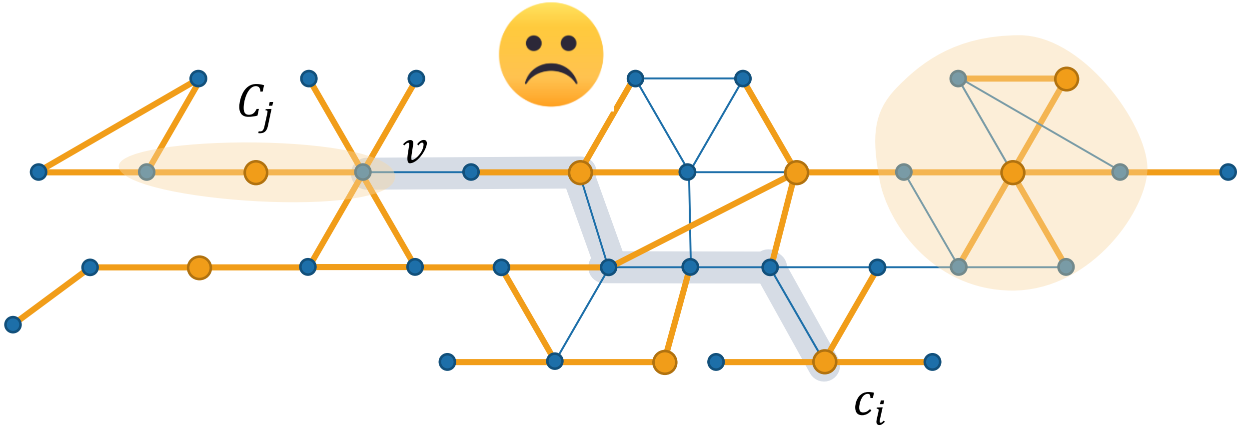
Path Buying

- For $k = 1,2,4,8, \ldots, n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges



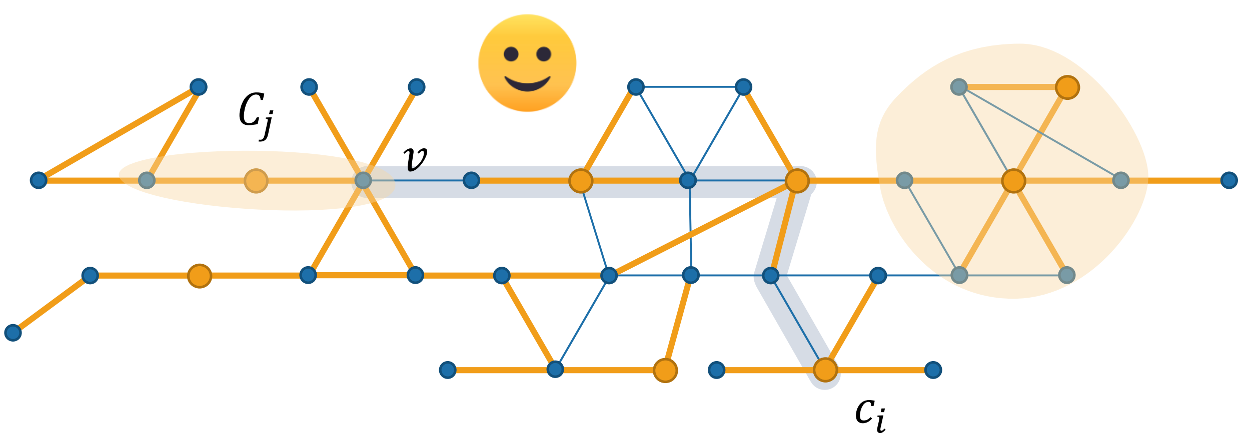
Path Buying

- For $k = 1,2,4,8, \ldots, n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges



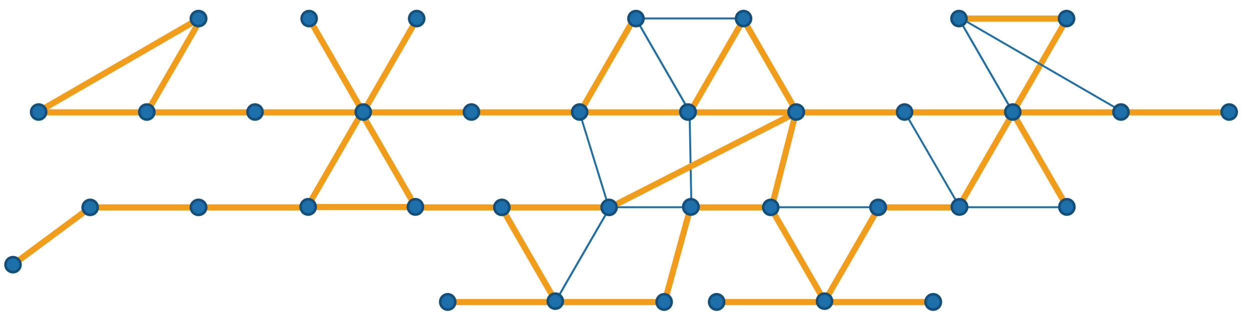
Path Buying

- For $k = 1,2,4,8, \ldots, n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges



Path Buying

- For $k = 1,2,4,8, \dots, n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges

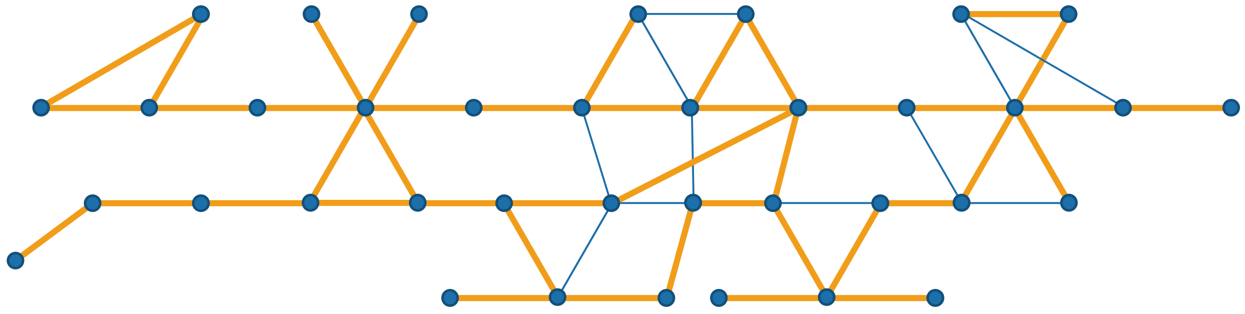


Distributed Spanner Construction

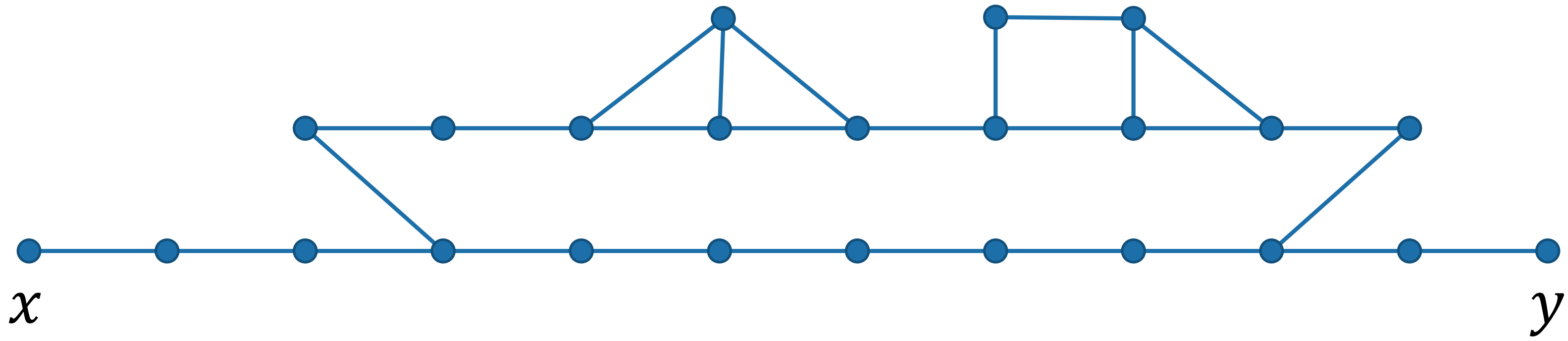
Theorem (New)

It is possible to construct:

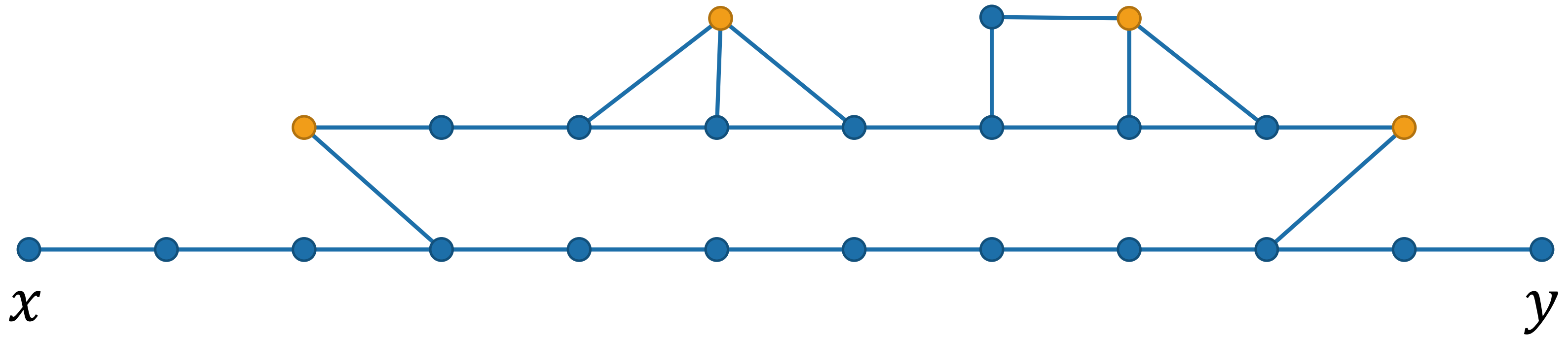
- A $(+6)$-spanner
- With $\tilde{O}(n^{4/3})$ edges
- In $\tilde{O}(n^{2/3} + D)$ rounds



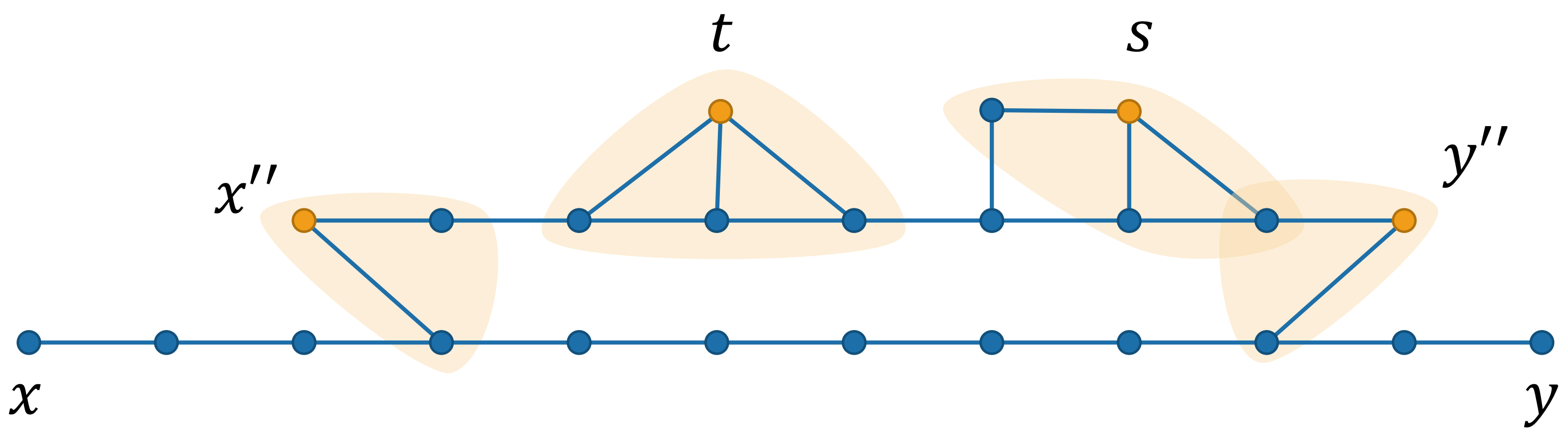
Stretch



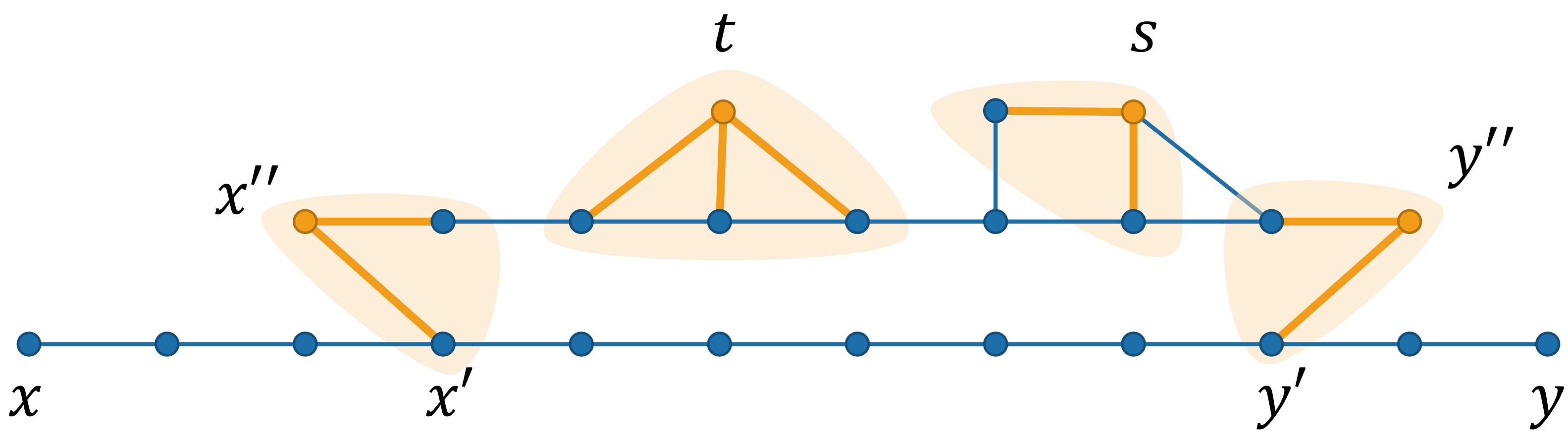
Stretch



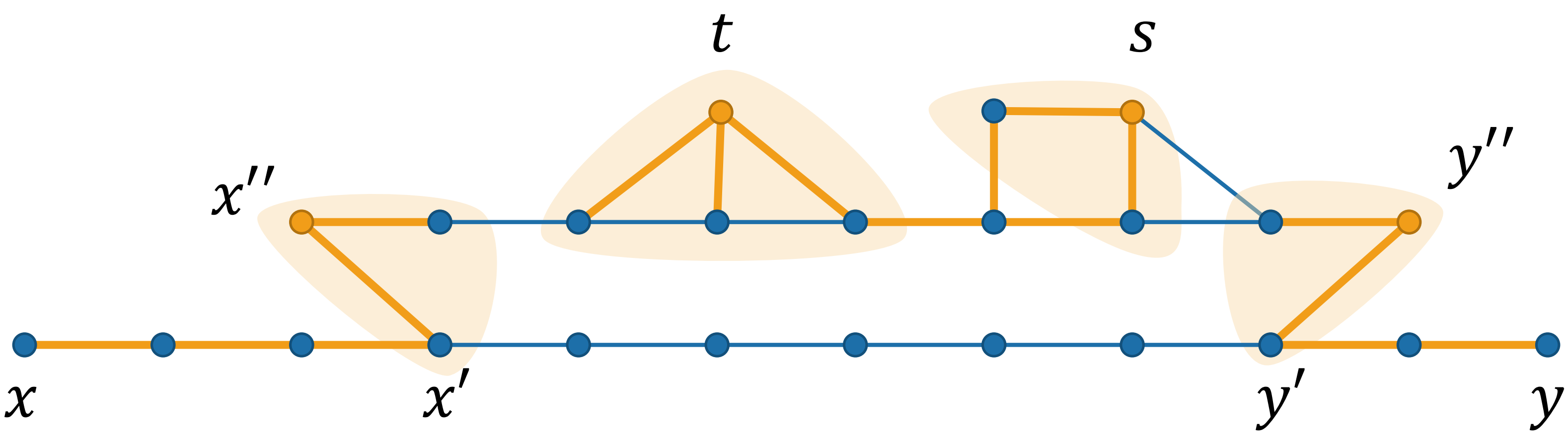
Stretch



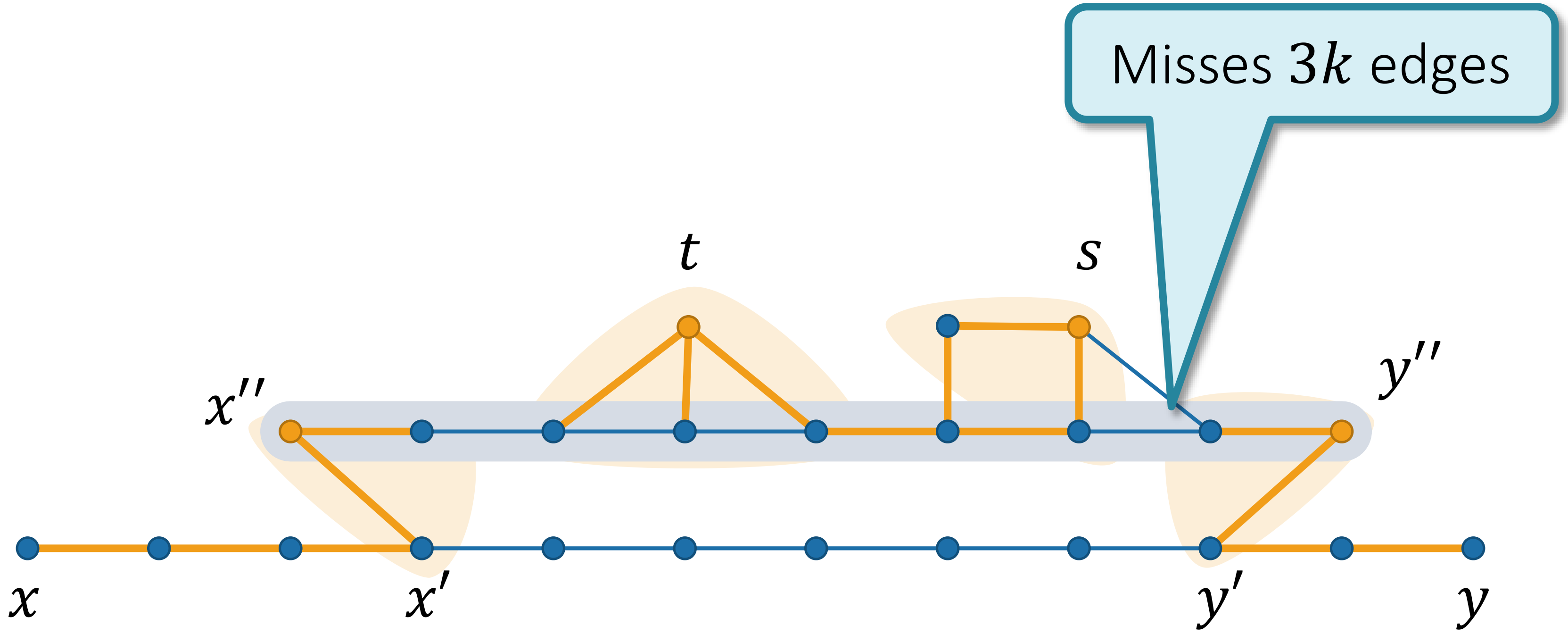
Stretch



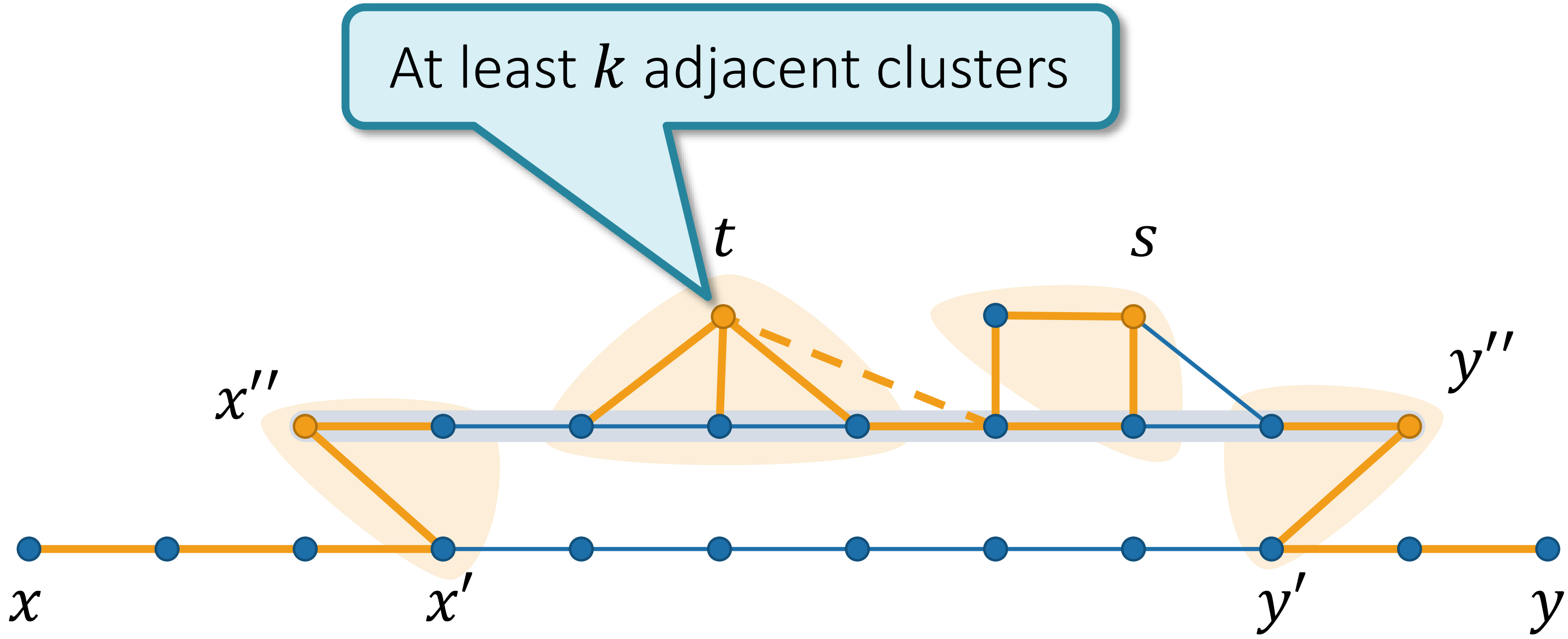
Stretch



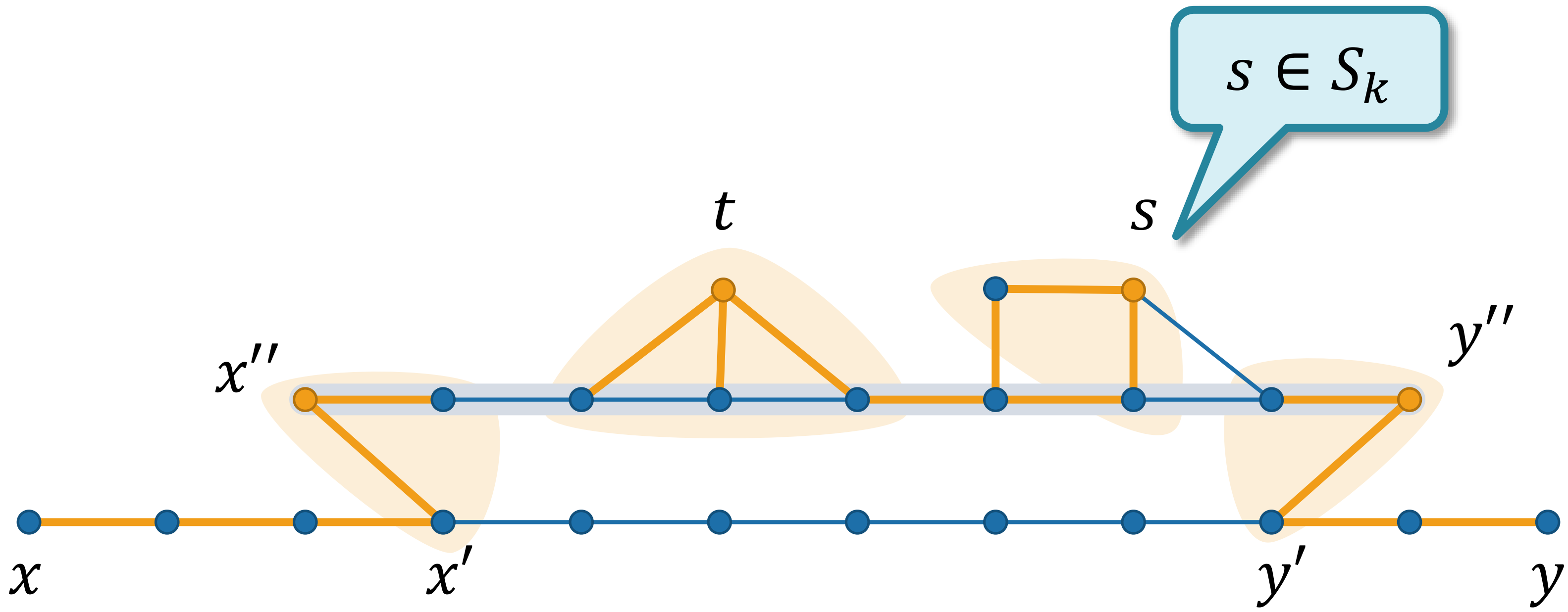
Stretch



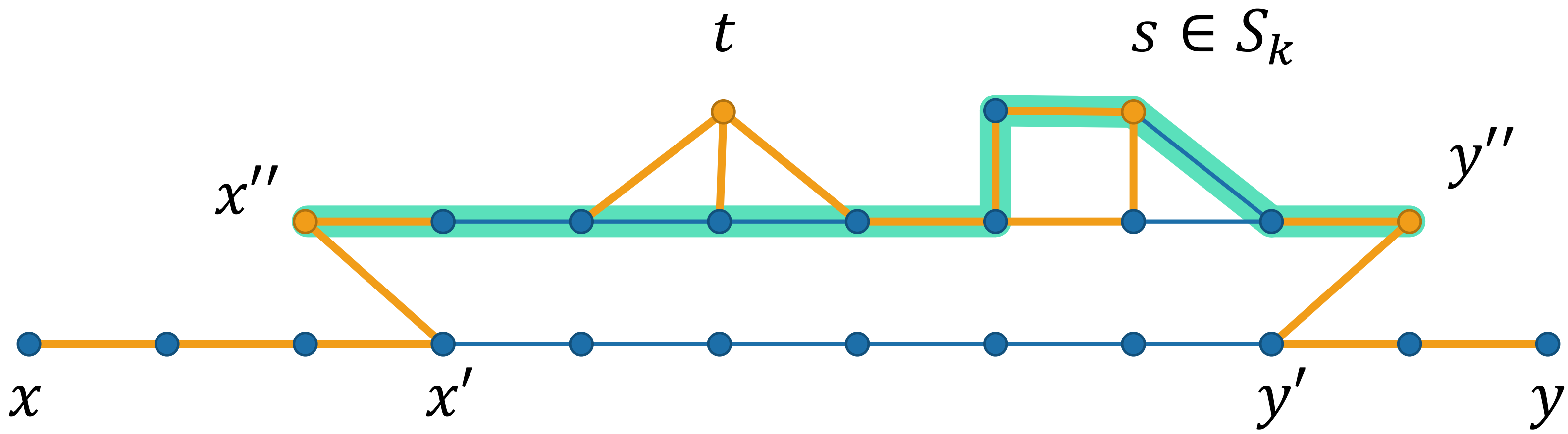
Stretch



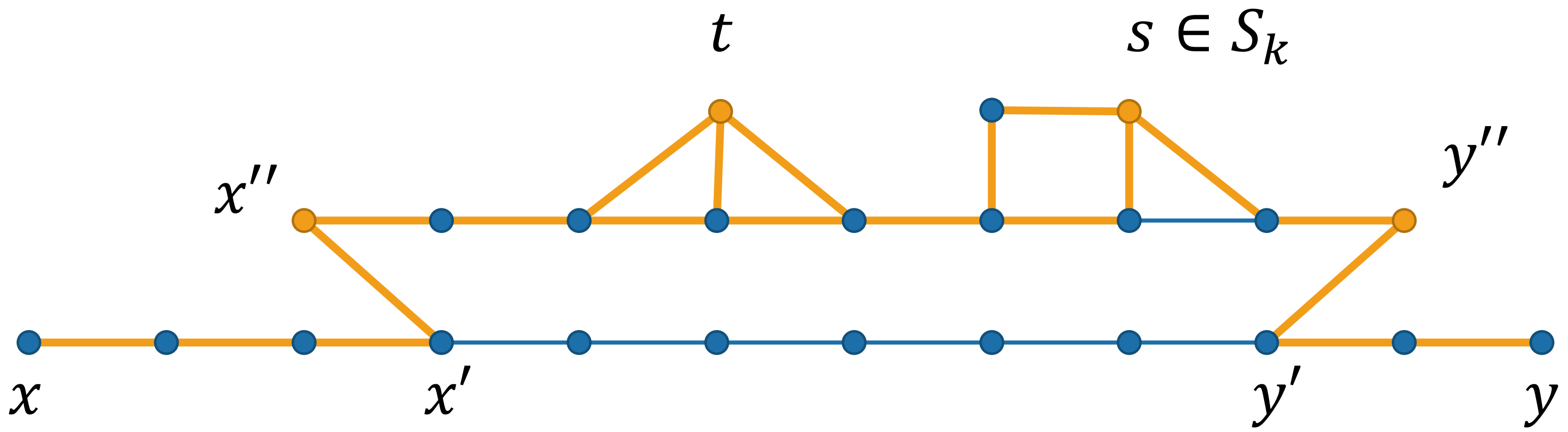
Stretch



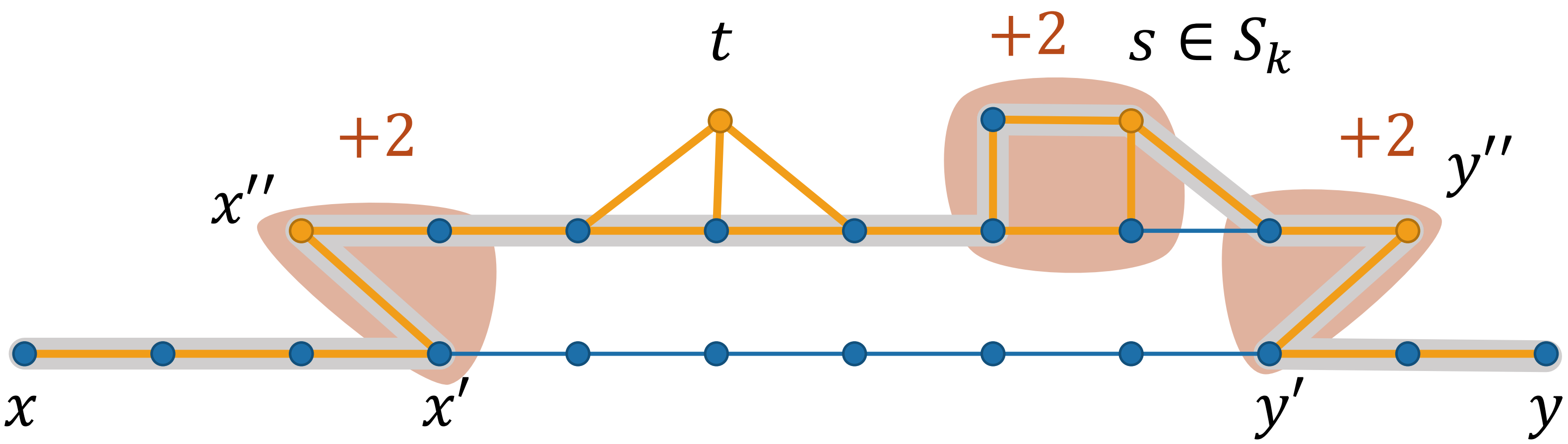
Stretch



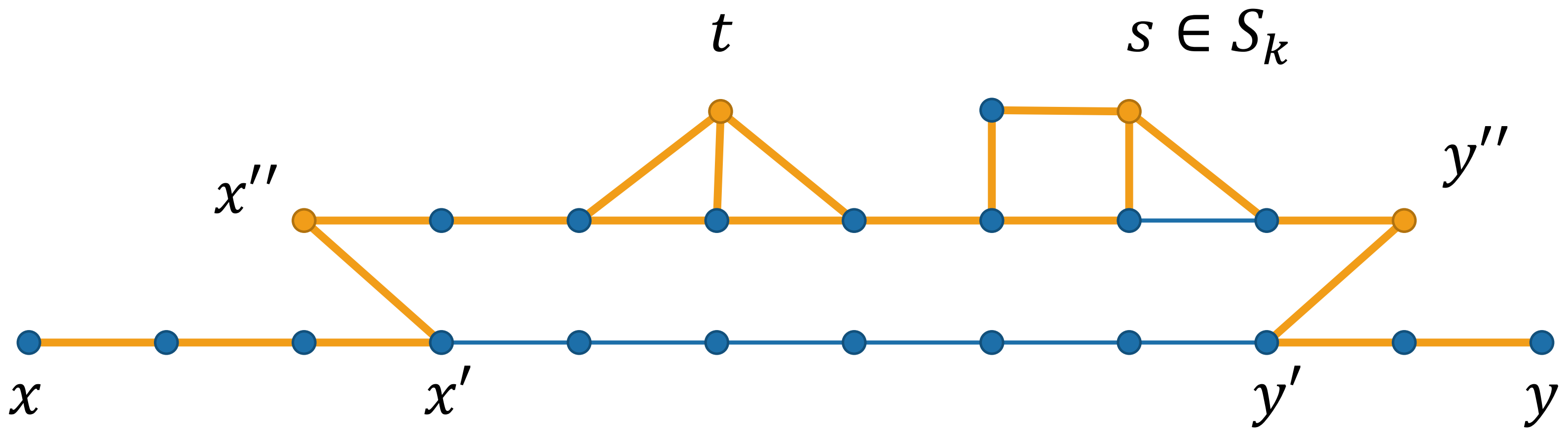
Stretch



Stretch



Stretch



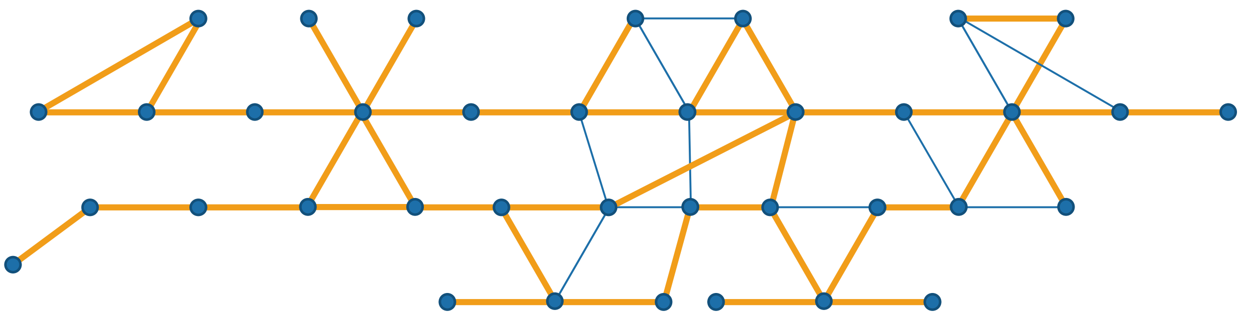
+6 streatch

Distributed Spanner Construction

Theorem (New)

It is possible to construct:

- A $(+6)$-spanner
- With $\tilde{O}(n^{4/3})$ edges
- In $\tilde{O}(n^{2/3} + D)$ rounds



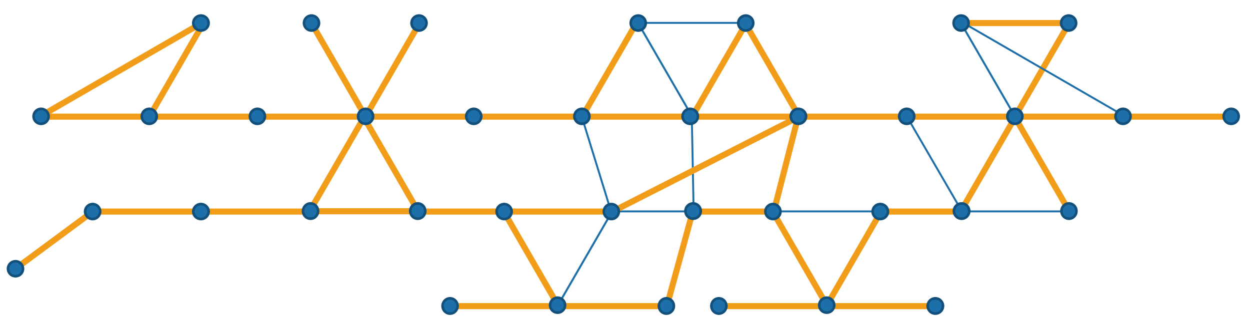
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
- Add all edges of un-clustered nodes

Locally

Talk to neighbors

Talk to neighbors

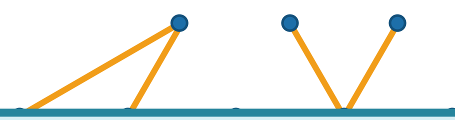


Path Buying

- For $k = 1,2,4,8,\ldots,n^{2/3}$ do:
  - Build a set S_k of $\sim 1/k$ of the clusters
  - For each center c_i and a cluster $C_j \in S_k$
    - Add a shortest path from c_i to some $v \in C_j$
    - But only if it misses at most $2k$ edges

Join locally to S_k

For each (c_i, C_j), for each $v \in C_j$, need to find the shortest (c_i, v)-path that misses minimal num. of edges



Note: Graph and spanner are **unweighted**
Only use weights for the alg.

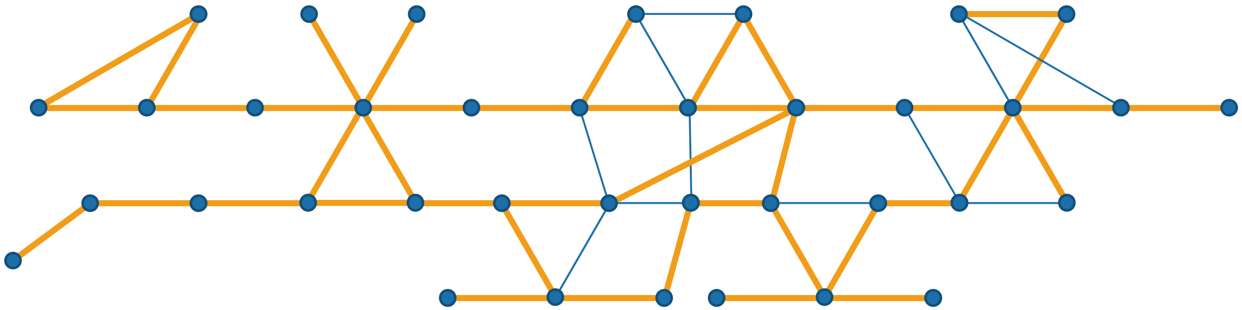
Weight edges: missing=1, others=0
Run wBFS from each c_i

Distributed Spanner Construction

Theorem (New)

It is possible to construct:

- A $(+6)$-spanner
- With $\tilde{O}(n^{4/3})$ edges
- In $\tilde{O}(n^{2/3}+D)$ rounds



Conclusion

- New sequential algorithm for $(+6)$-spanners
- New distributed implementation
  - Gives an almost-optimal $(+6)$-spanner
- New distributed algorithm: weighted-BFS

- Open: lower bounds for distributed construction time

Thank You!