Constant-Round Near-Optimal Spanners in Congested Clique

Shiri Chechik

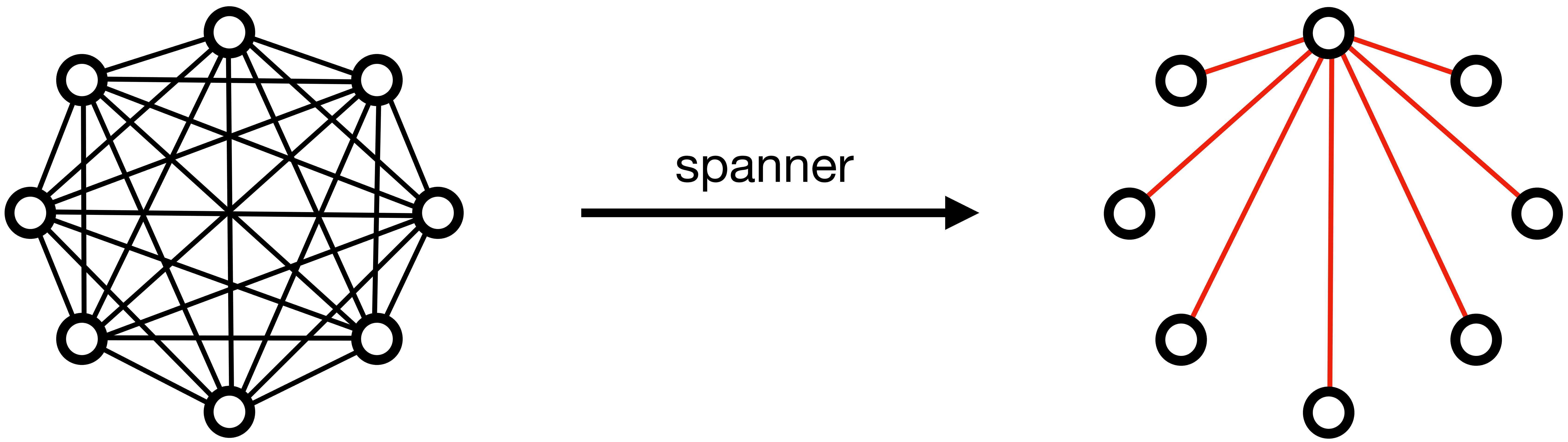
Tianyi Zhang

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

Graph Spanner

- **Input:**
  $G = (V, E, \omega)$ an undirected weighted graph

- **Output:**
  $H \subseteq G$ a **subgraph** of small size that approximately preserves distances



Graph Spanner

- **Input:**
  $G = (V, E, \omega)$ an undirected weighted graph

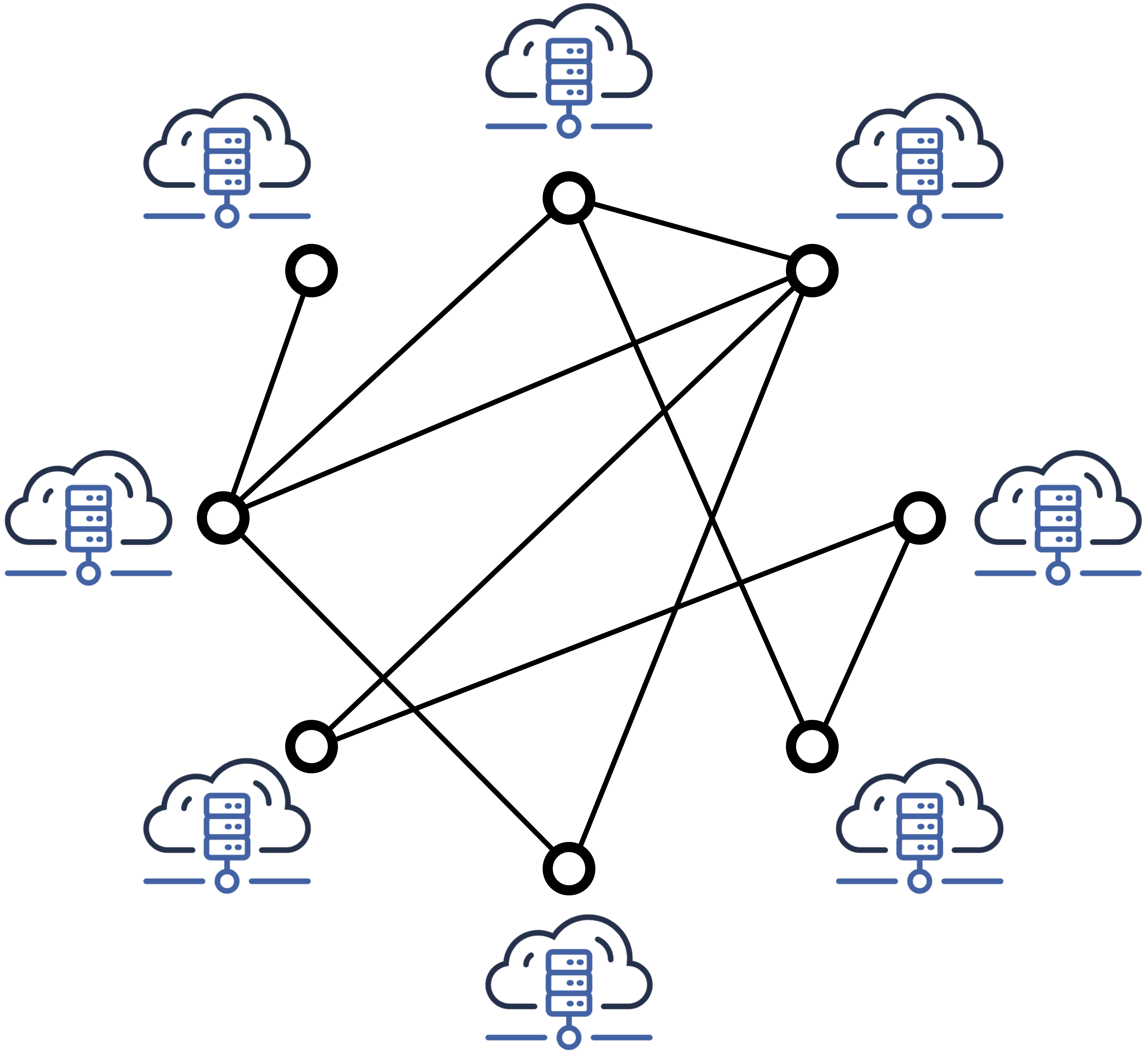
- **Output:**
  $H \subseteq G$ a **subgraph** of small size that approximately preserves distances

Multiplicative stretch:

- **Size:** $|H| = O(n^{1+1/k})$
- **Approximation:** $\forall s, t \in V, \text{dist}_G(s, t) \leq \text{dist}_H(u, v) \leq (2k-1)\text{dist}_G(s, t)$
- Optimal under the girth conjecture

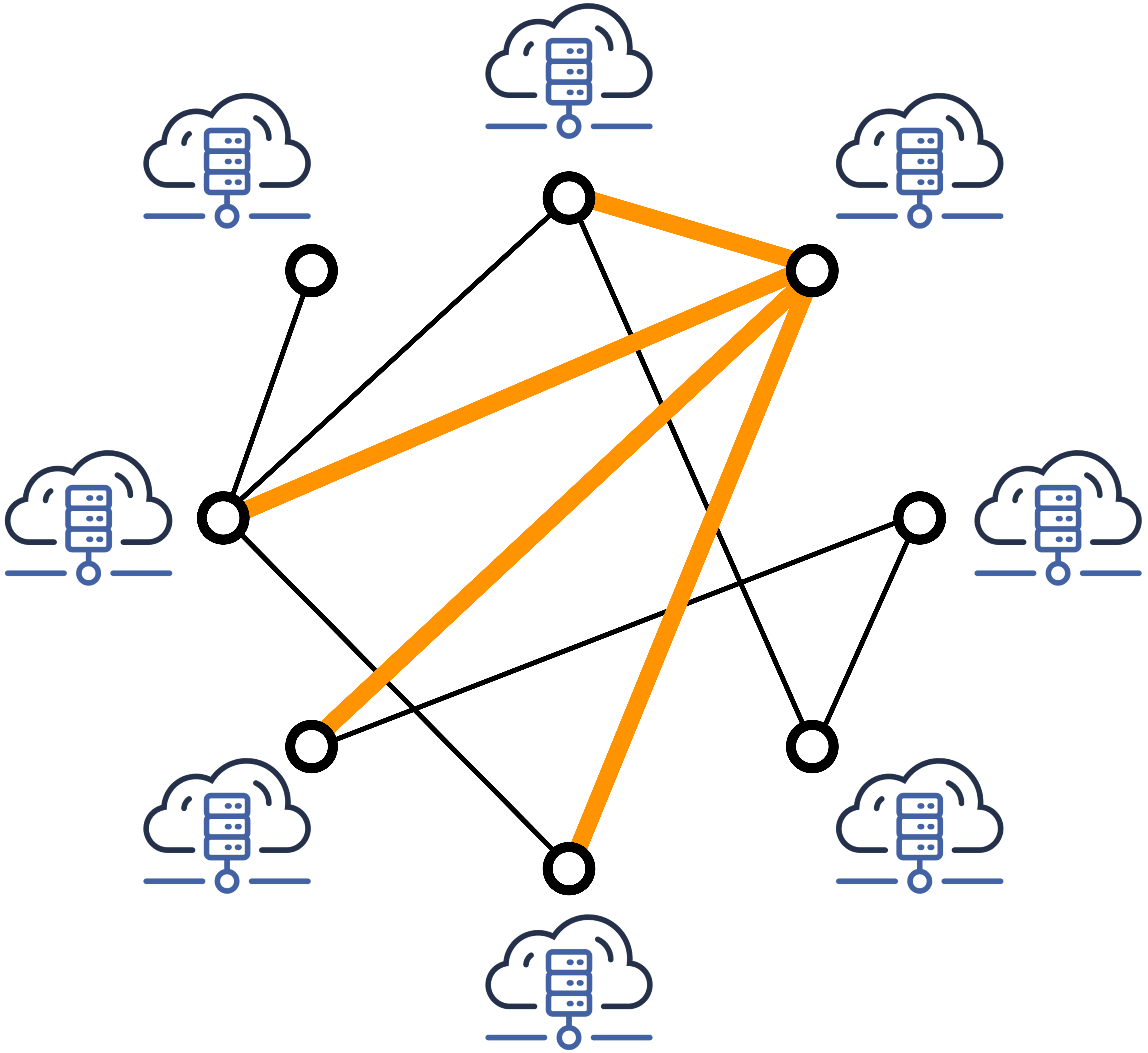
Congested Clique

- Input graph is stored distributively
- Synchronous, all-to-all communication
- Message size = $O(\log n)$ bits
- Runtime = #communication rounds



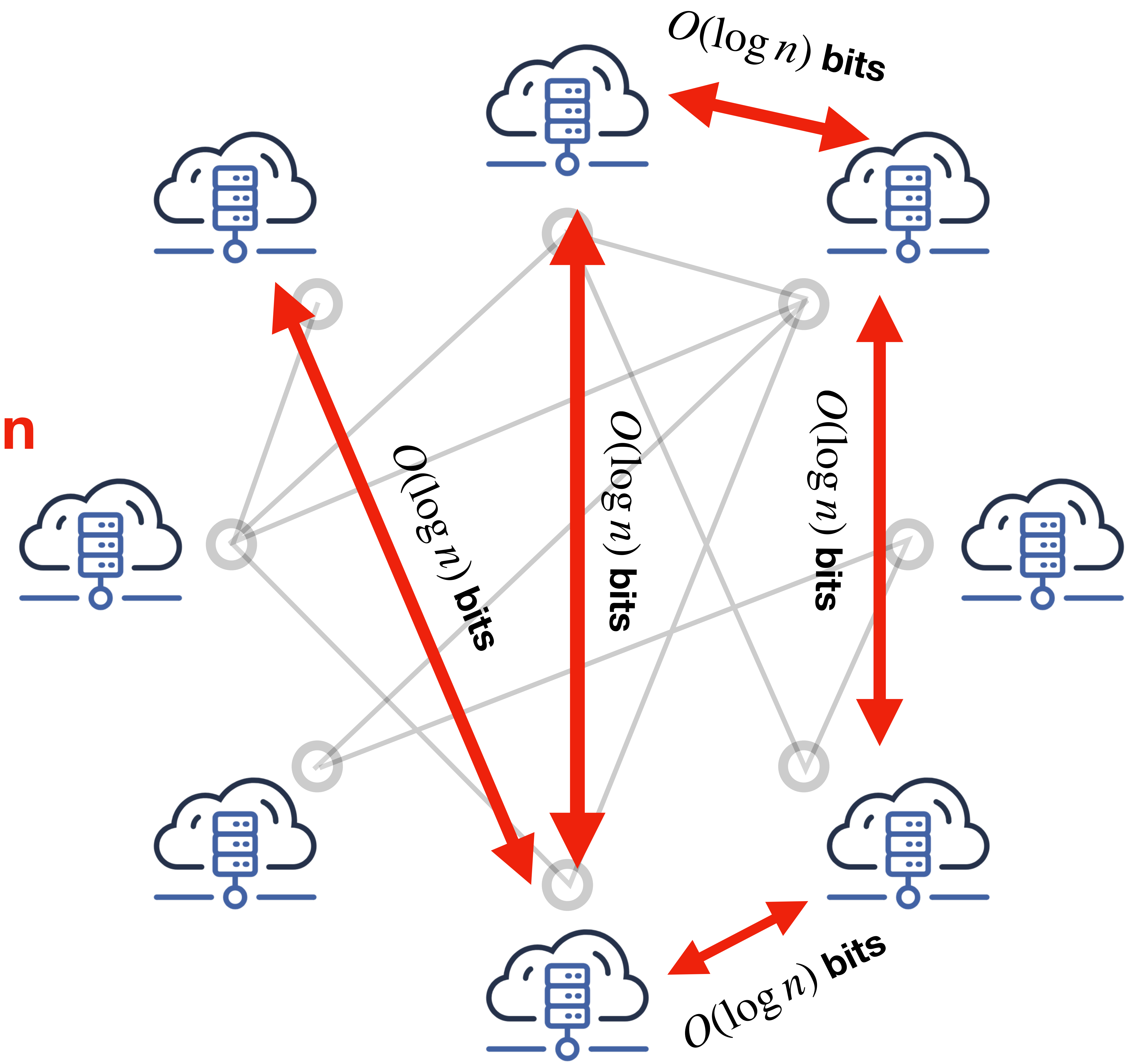
Congested Clique

- **Input graph is stored distributively**
- Synchronous, all-to-all communication
- Message size = $O(\log n)$ bits
- Runtime = #communication rounds



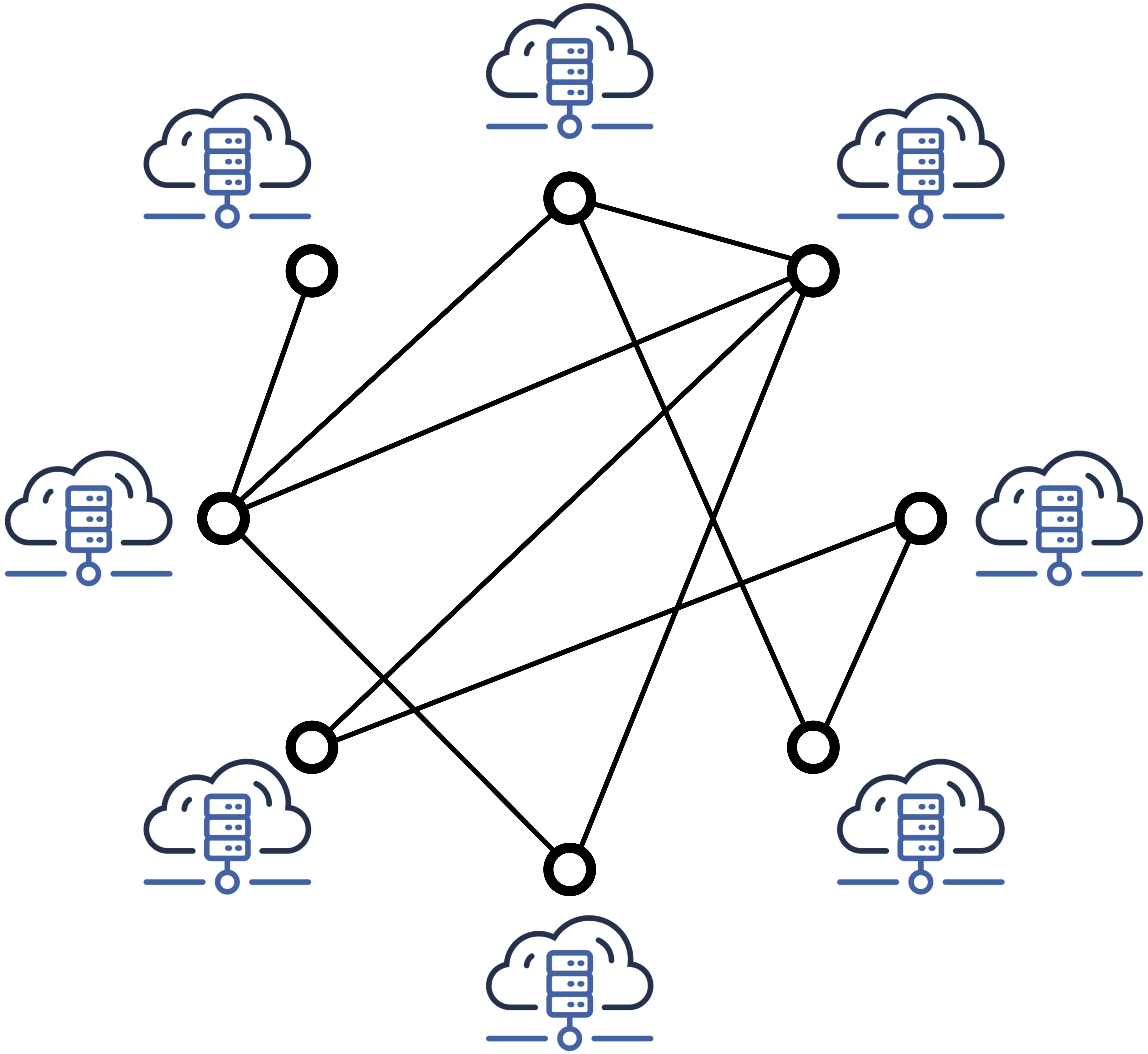
Congested Clique

- Input graph is stored distributively
- **Synchronous, all-to-all communication**
- **Message size = $O(\log n)$ bits**
- Runtime = #communication rounds



Congested Clique

- Input graph is stored distributively
- Synchronous, all-to-all communication
- Message size = $O(\log n)$ bits
- **Runtime = #communication rounds**
- **What is the runtime for spanners?**



History

reference	stretch	#edges	#rounds	input type
Baswana, Sen 2007	$2k-1$	$O(n^{1+1/k}+kn)$	$O(k)$	unweighted
Baswana, Sen 2007	$2k-1$	$O(kn^{1+1/k})$	$O(k)$	weighted
Parter, Yogev 2018	$2k-1$	$O(n^{1+1/k}\log^2 n)$	$O(\log k)$	unweighted
Biswas et al 2021	$k^{1+o(1)}$	$O(n^{1+1/k}\log k)$	$\log^{O(1)} k$	weighted

large space high runtime

History

reference	stretch	#edges	#rounds	input type
Dory et al 2021	$O(k)$	$O(n^{1+1/k})$	O(1)	unweighted
Dory et al 2021	$O(k)$	$O(n^{1+1/k} \log n)$	O(1)	weighted
Dory et al 2021	$O(k \log n)$	$O(n^{1+1/k})$	O(1)	weighted

large stretch **sub-opt space** **opt**

Our results

reference	stretch	#edges	#rounds	input type
new	$2k-1$	$O(n^{1+1/k})$	O(1)	unweighted
new	$(1+\epsilon)(2k-1)$	$O(n^{1+1/k})$	O(1)	weighted
new	$2k-1$	$O(kn^{1+1/k})$	O(1)	weighted
	near-opt	**near-opt**	**opt**	

Our results

	reference	stretch	#edges	#rounds	input type
This talk	**new**	$2k-1$	$O(n^{1+1/k})$	O(1)	unweighted
	new	$(1+\epsilon)(2k-1)$	$O(n^{1+1/k})$	O(1)	weighted
	new	$2k-1$	$O(kn^{1+1/k})$	O(1)	weighted
		near-opt	**near-opt**	**opt**	

Outline of main algorithm

Degree reduction

Neighborhood computation

Parallel spanner simulation

Outline of main algorithm

Degree reduction

Neighborhood computation

Parallel spanner simulation

Degree Reduction

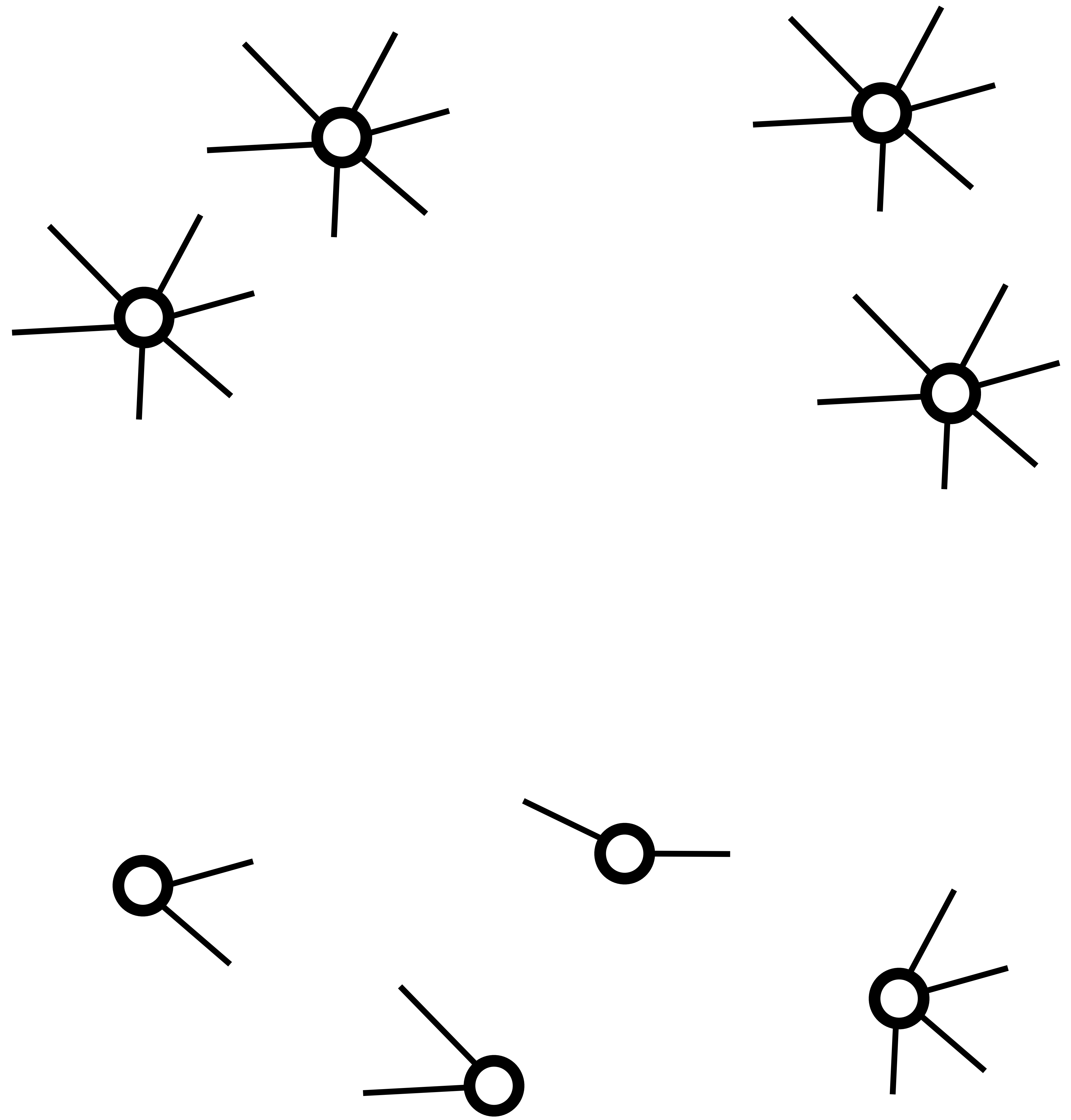
<u>Goal:</u> reduce **maximum degree** to $n^{20/k}$

1. Partition the vertex set $V = V_{\text{hi}} \cup V_{\text{lo}}$

$$V_{\text{hi}} = \{v \mid \deg(v) > n^{20/k}\}$$
$$V_{\text{lo}} = \{v \mid \deg(v) \leq n^{20/k}\}$$

2. Find a hitting set of size $n^{1-10/k}$ which dominates V_{hi} [DFKL'21]

3. Find a $0.2k$-stretch spanner [DFKL'21] on a contracted graph



Degree Reduction

Goal: reduce **maximum degree** to $n^{20/k}$

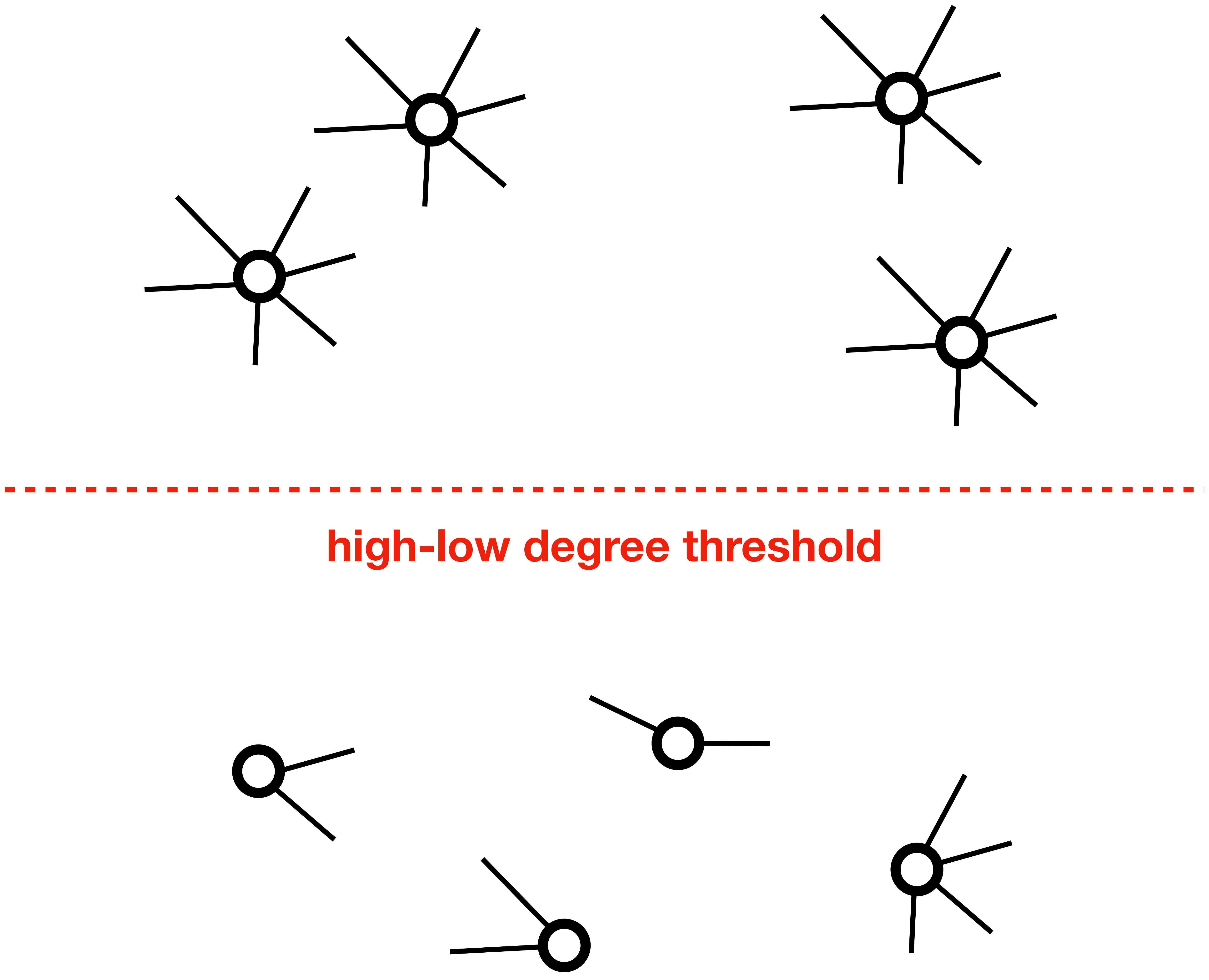
1. Partition the vertex set $V = V_{\text{hi}} \cup V_{\text{lo}}$

   $V_{\text{hi}} = \{v \mid \deg(v) > n^{20/k}\}$

   $V_{\text{lo}} = \{v \mid \deg(v) \leq n^{20/k}\}$

2. Find a hitting set of size $n^{1-10/k}$ which dominates V_{hi} [DFKL'21]

3. Find a $0.2k$-stretch spanner [DFKL'21] on a contracted graph



Degree Reduction

<u>Goal:</u> reduce **maximum degree** to $n^{20/k}$

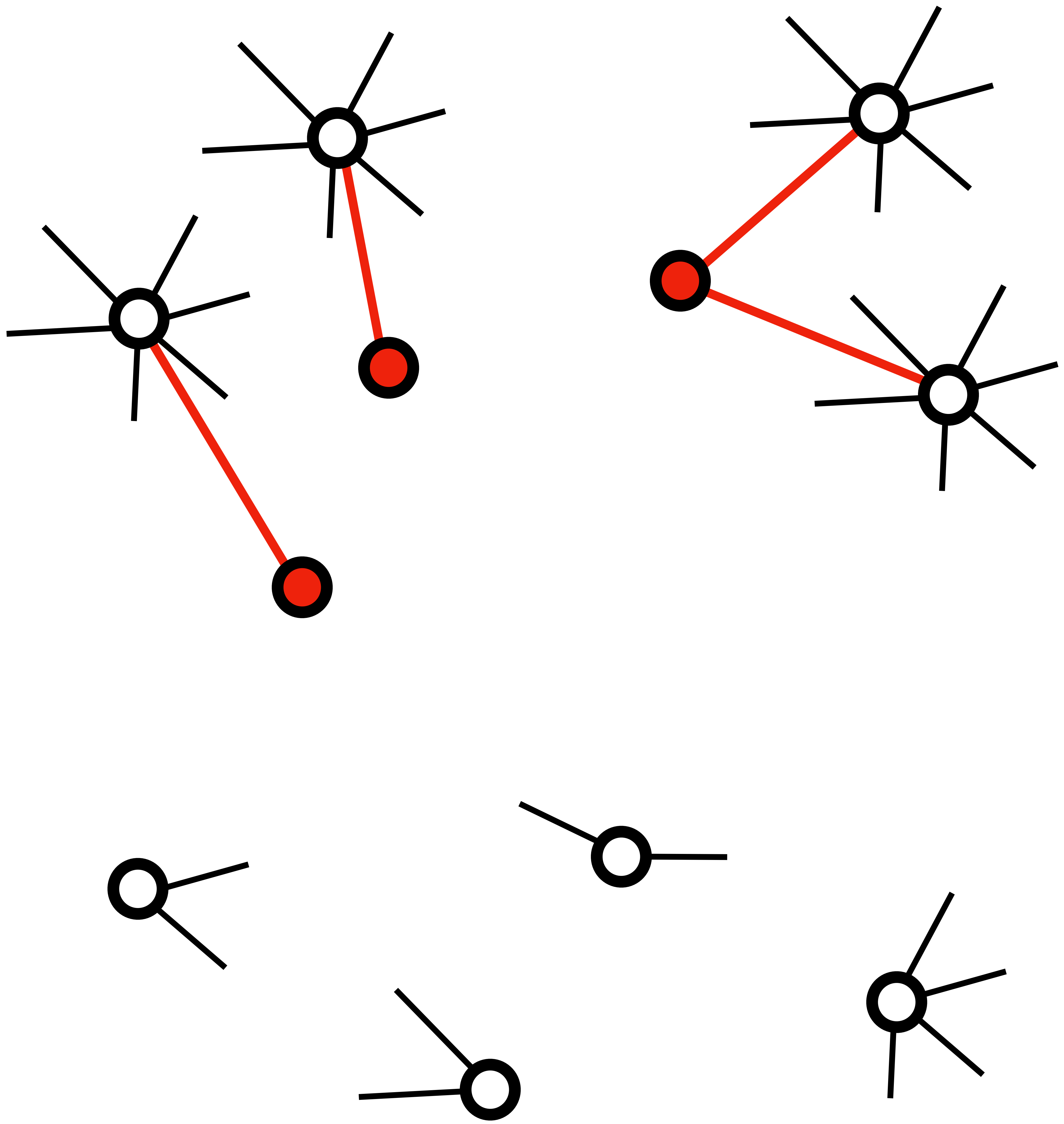
1. Partition the vertex set $V = V_{\text{hi}} \cup V_{\text{lo}}$

   $V_{\text{hi}} = \{v \mid \deg(v) > n^{20/k}\}$

   $V_{\text{lo}} = \{v \mid \deg(v) \leq n^{20/k}\}$

2. Find a hitting set of size $n^{1-10/k}$ which dominates V_{hi} [DFKL'21]

3. Find a $0.2k$-stretch spanner [DFKL'21] on a contracted graph



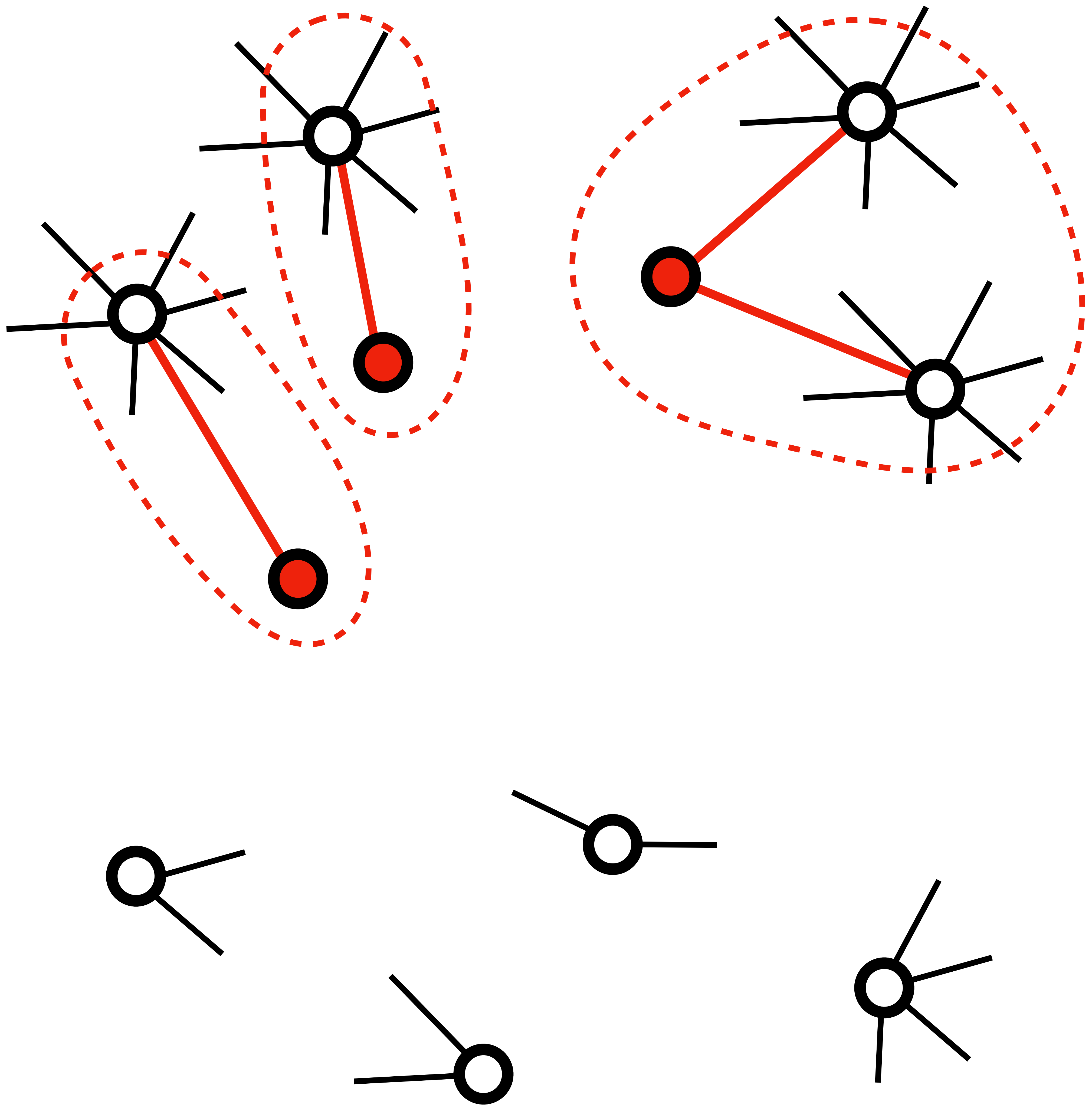
Degree Reduction

<u>Goal:</u> reduce **maximum degree** to $n^{20/k}$

1. Partition the vertex set $V = V_{\text{hi}} \cup V_{\text{lo}}$
   $$V_{\text{hi}} = \{v \mid \deg(v) > n^{20/k}\}$$
   $$V_{\text{lo}} = \{v \mid \deg(v) \leq n^{20/k}\}$$

2. Find a hitting set of size $n^{1-10/k}$ which dominates V_{hi} [DFKL'21]

3. Find a $0.2k$-stretch spanner [DFKL'21] on a contracted graph



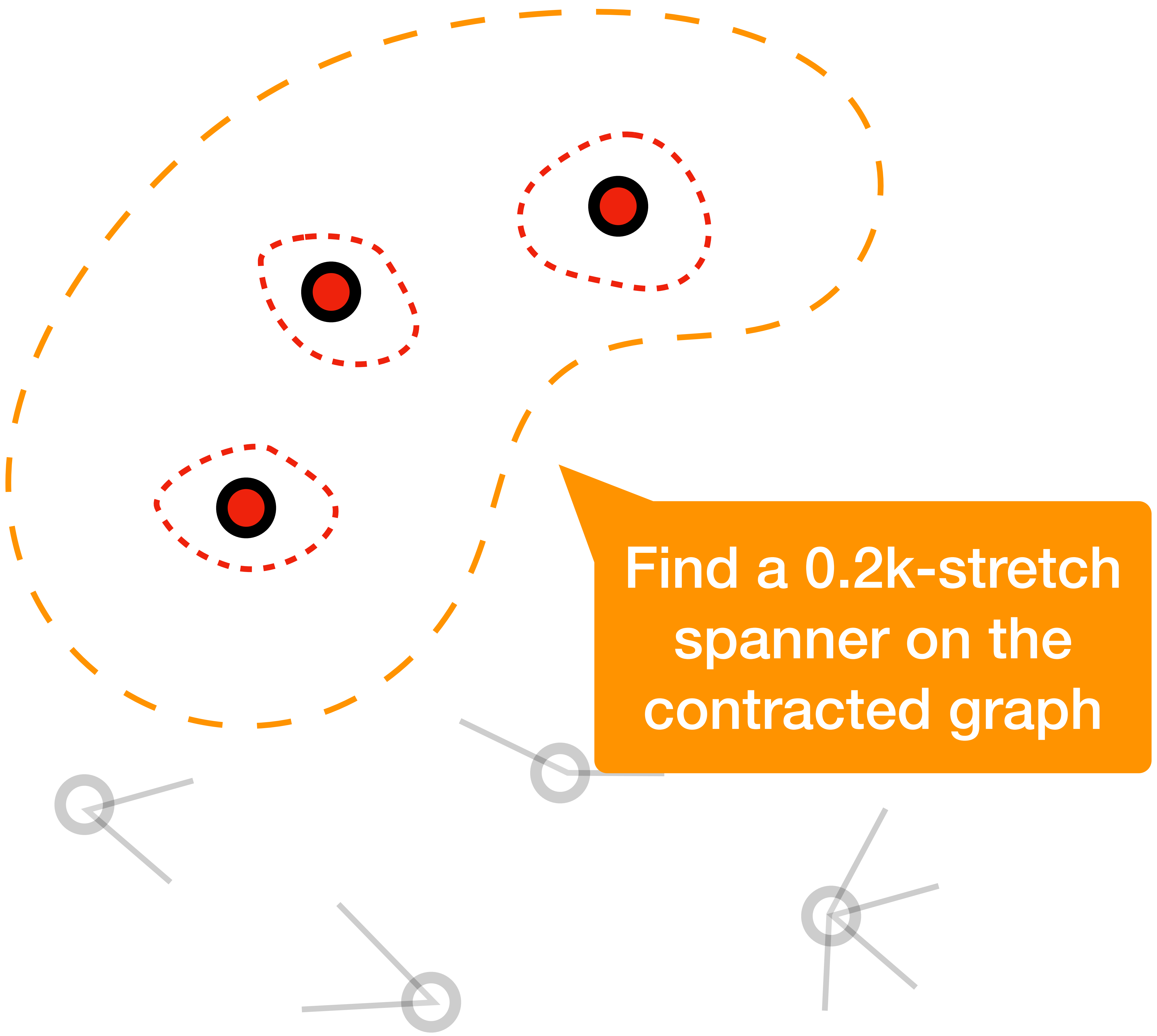
Degree Reduction

Goal: reduce **maximum degree** to $n^{20/k}$

1. Partition the vertex set $V = V_{\text{hi}} \cup V_{\text{lo}}$
   $$V_{\text{hi}} = \{v \mid \deg(v) > n^{20/k}\}$$
   $$V_{\text{lo}} = \{v \mid \deg(v) \leq n^{20/k}\}$$

2. Find a hitting set of size $n^{1-10/k}$ which dominates V_{hi} [DFKL'21]

3. Find a $0.2k$-stretch spanner [DFKL'21] on a contracted graph



Degree Reduction

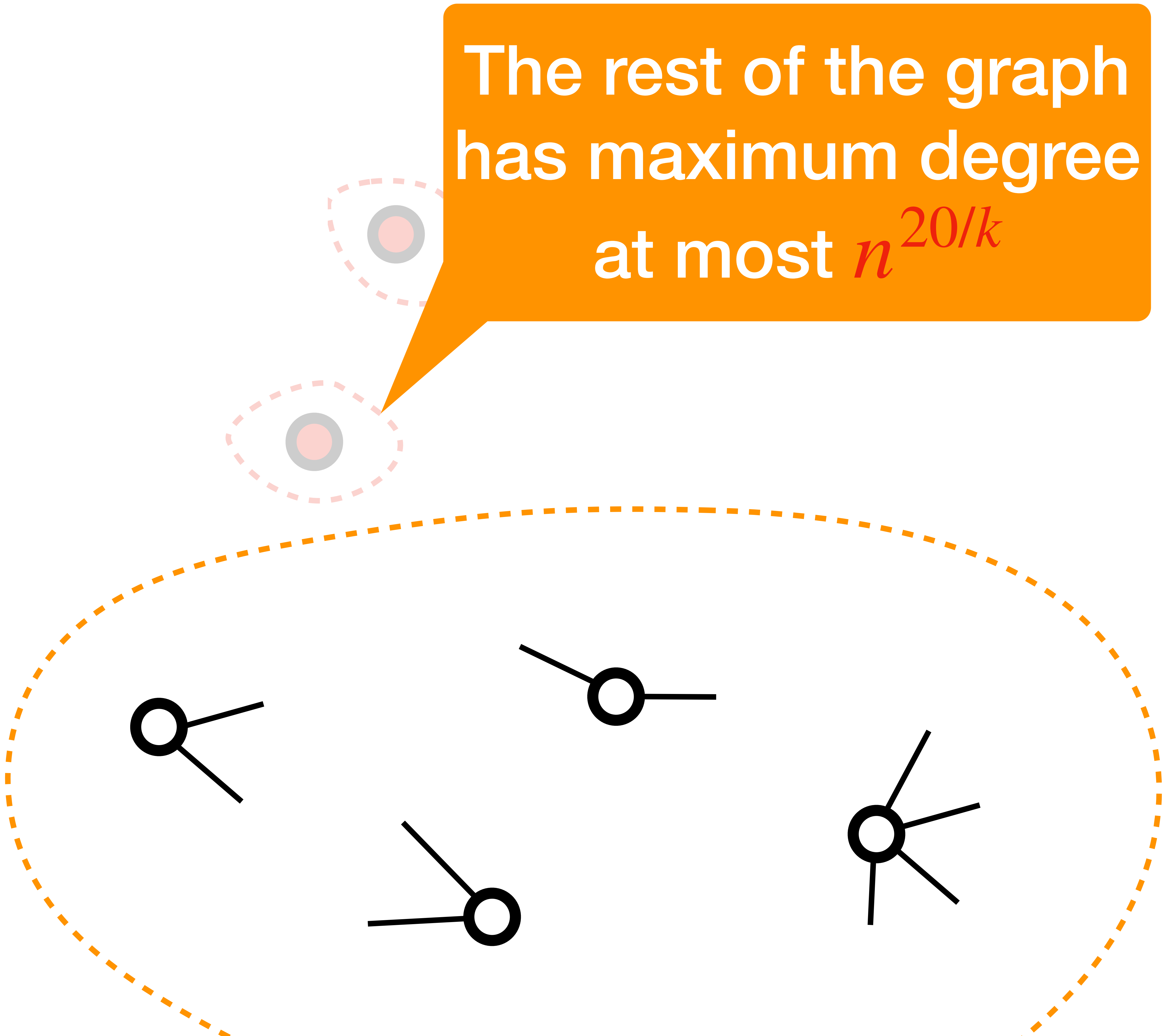
<u>Goal:</u> reduce **maximum degree** to $n^{20/k}$

1. Partition the vertex set $V = V_{\text{hi}} \cup V_{\text{lo}}$

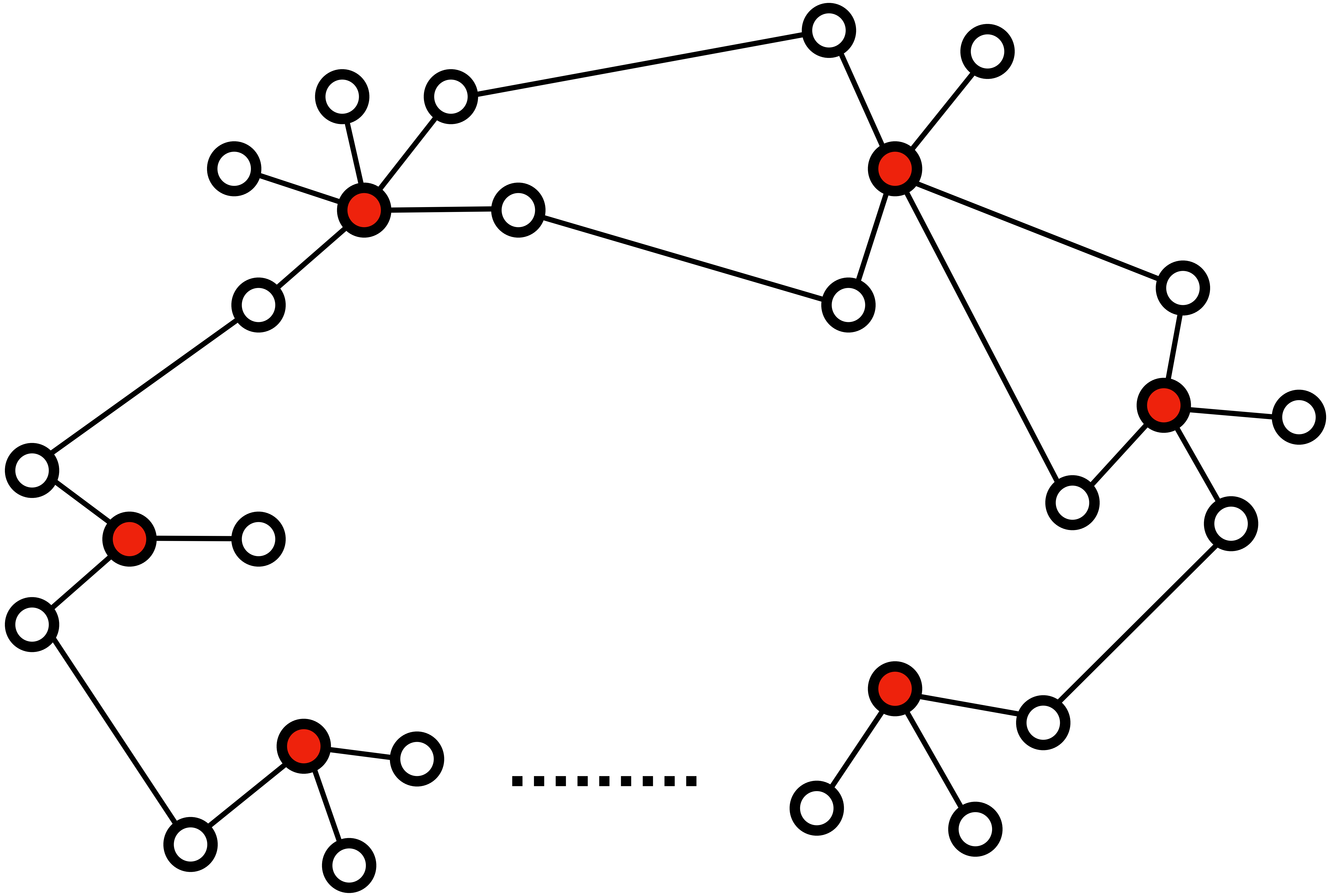
$$V_{\text{hi}} = \{v \mid \deg(v) > n^{20/k}\}$$
$$V_{\text{lo}} = \{v \mid \deg(v) \leq n^{20/k}\}$$

2. Find a hitting set of size $n^{1-10/k}$ which dominates V_{hi} [DFKL'21]

3. Find a $0.2k$-stretch spanner [DFKL'21] on a contracted graph

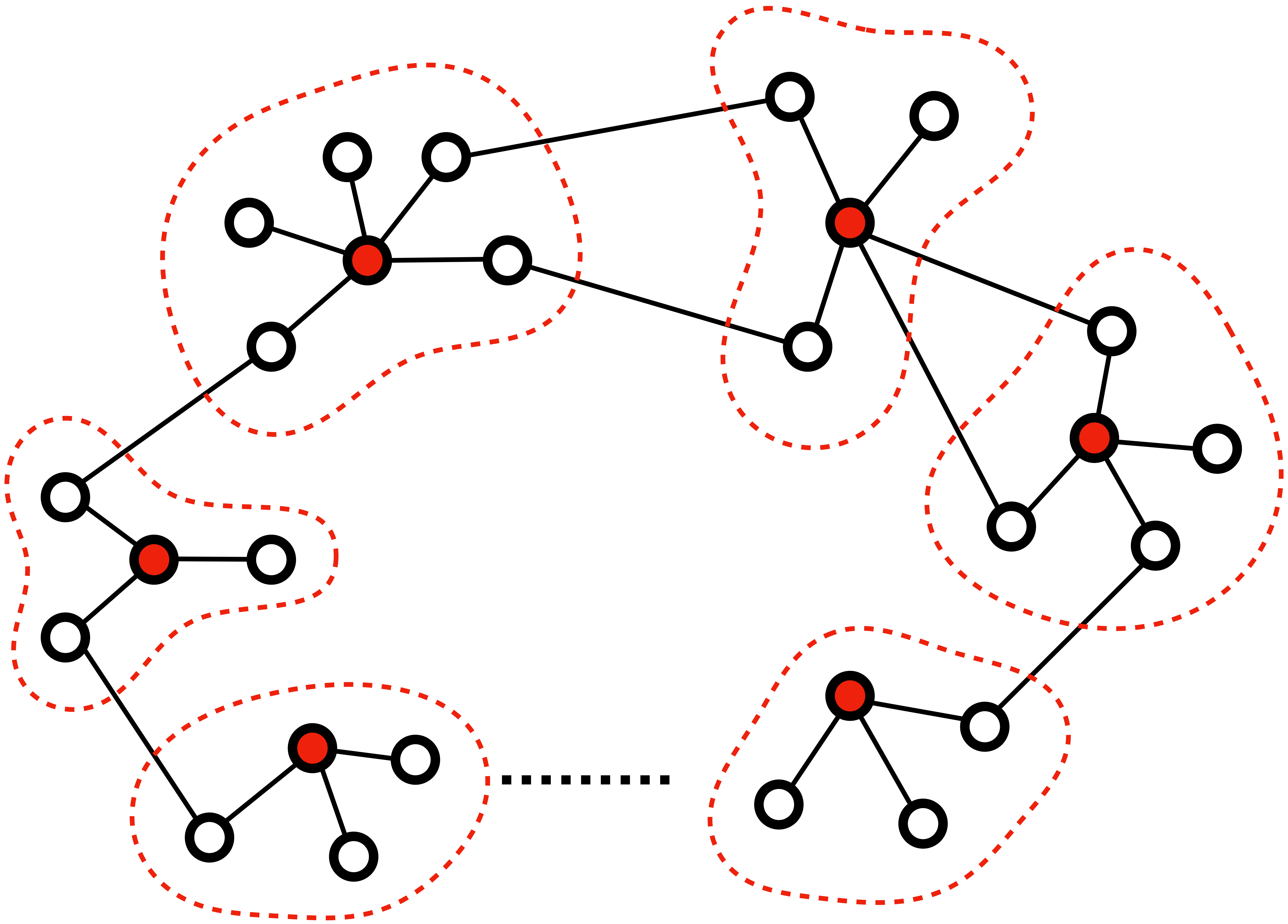


Analysis of size & stretch



Spanner on the **hitting set A**

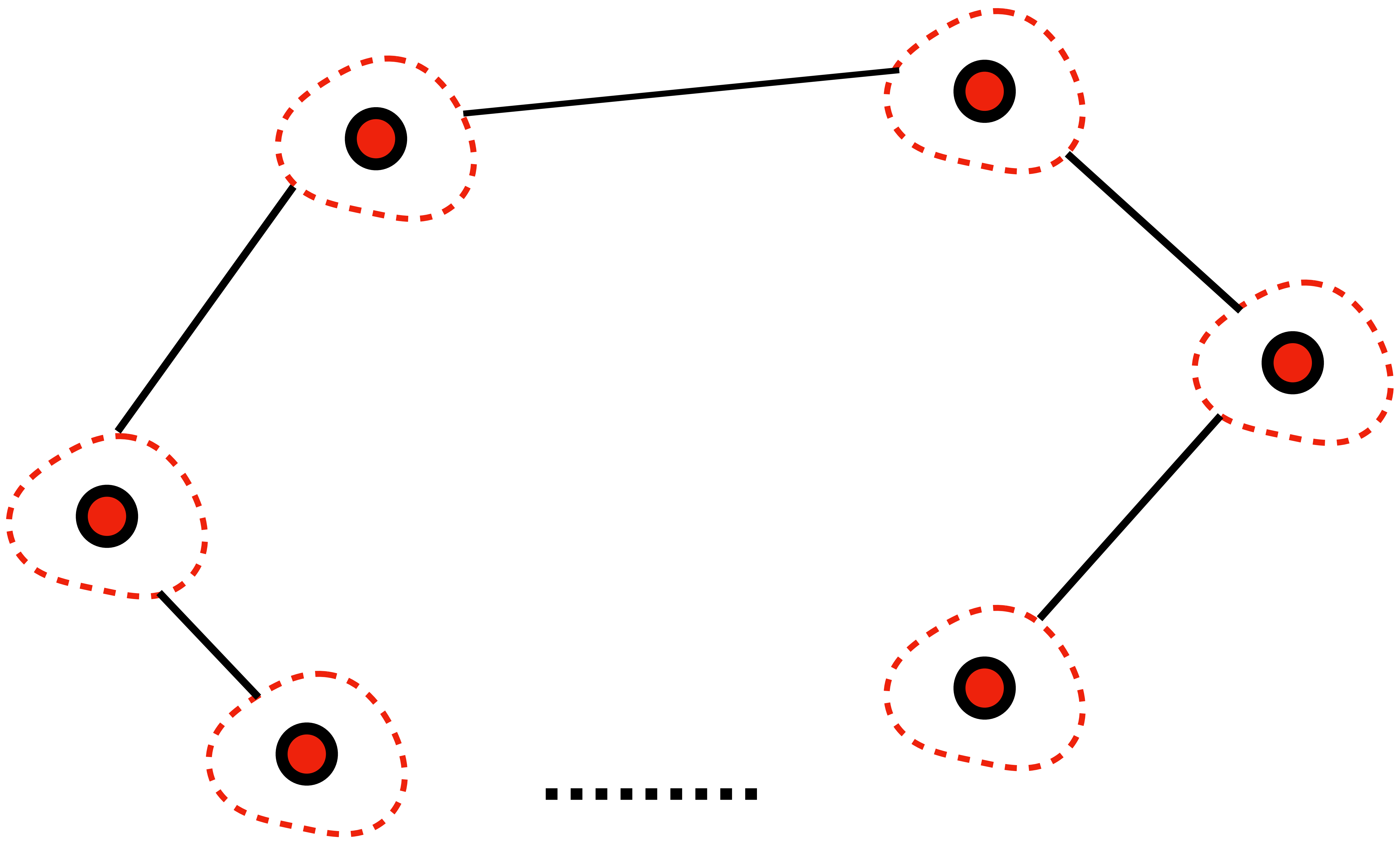
Analysis of size & stretch



Spanner on the **hitting set A**

1. Assign high-deg vertices to neighbor hitting set vertices

Analysis of size & stretch

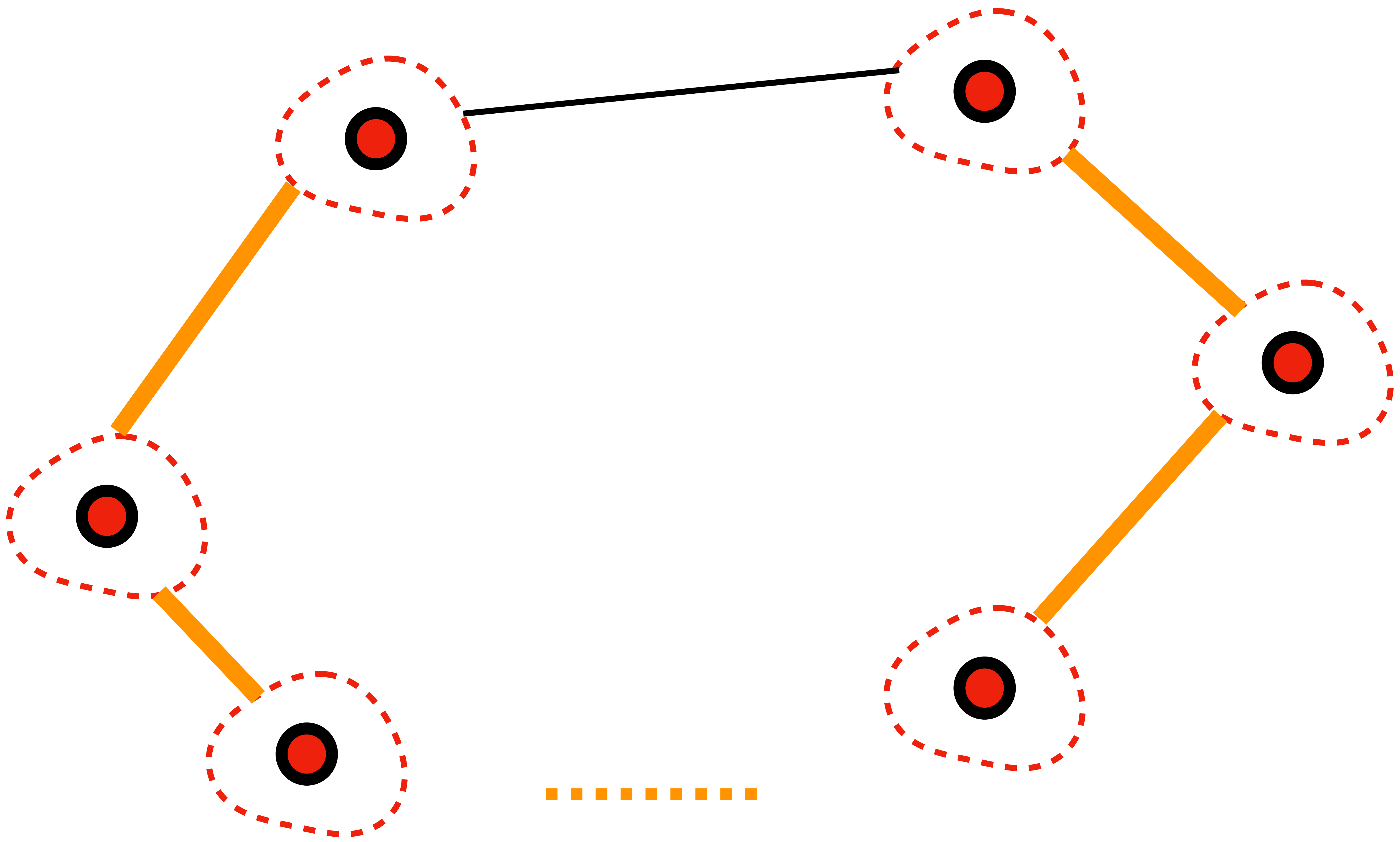


Spanner on the **hitting set A**

1. Assign high-deg vertices to neighbor hitting set vertices

2. Contract stars around hitting set vertices to single nodes

Analysis of size & stretch



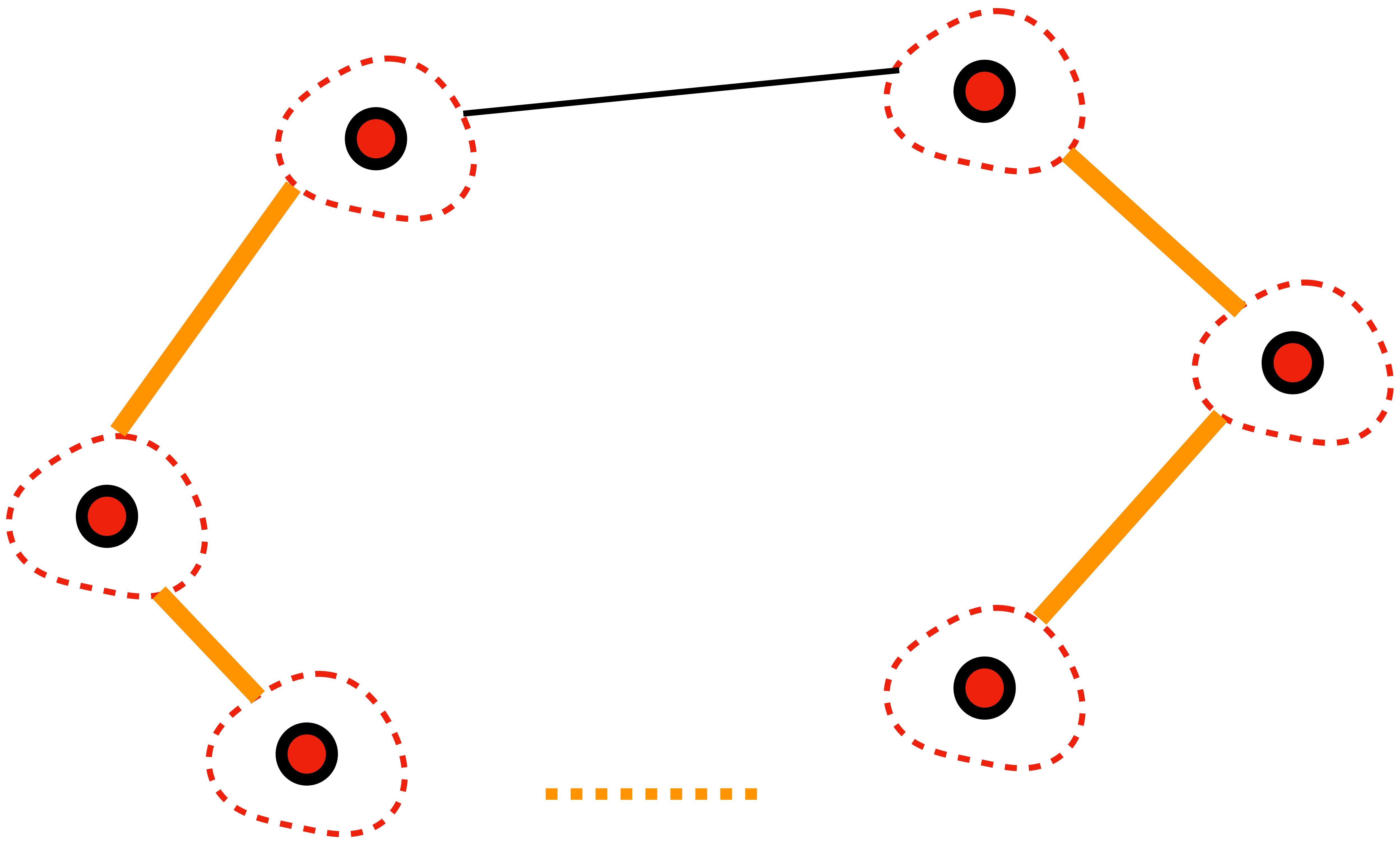
Spanner on the **hitting set A**

1. Assign high-deg vertices to neighbor hitting set vertices

2. Contract stars around hitting set vertices to single nodes

3. Build a **0.2k-stretch spanner** on the contracted graph [DFKL'21]

Analysis of size & stretch



Spanner on the **hitting set A**

1. Assign high-deg vertices to neighbor hitting set vertices

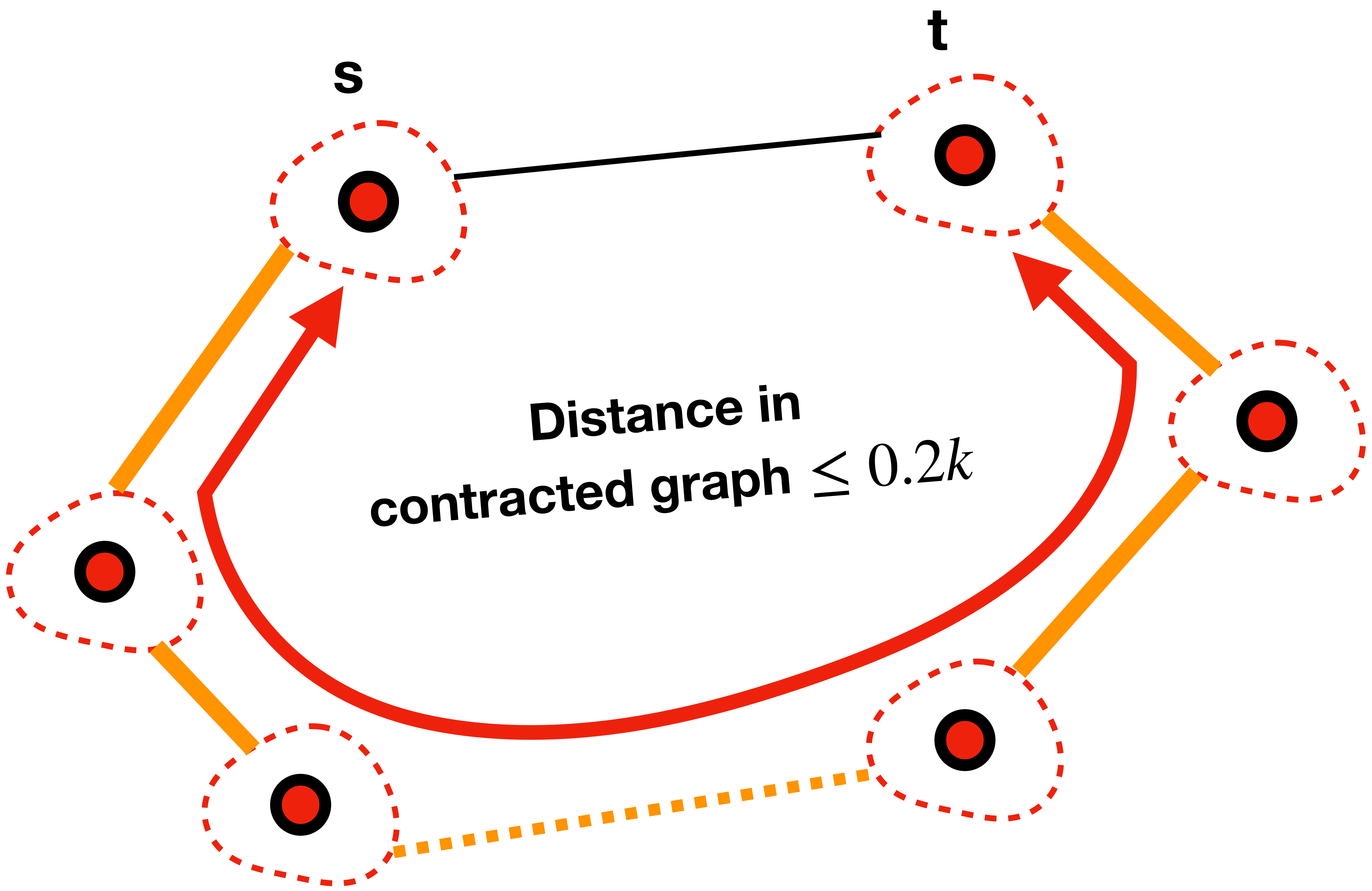
2. Contract stars around hitting set vertices to single nodes

3. Build a **0.2k-stretch spanner** on the contracted graph [DFKL'21]

Size analysis:

$$|A|^{1+O(1/k)} = n^{(1-10/k)\cdot(1+O(1/k))} = O(n)$$

Analysis of size & stretch



Spanner on the **hitting set A**

1. Assign high-deg vertices to neighbor hitting set vertices

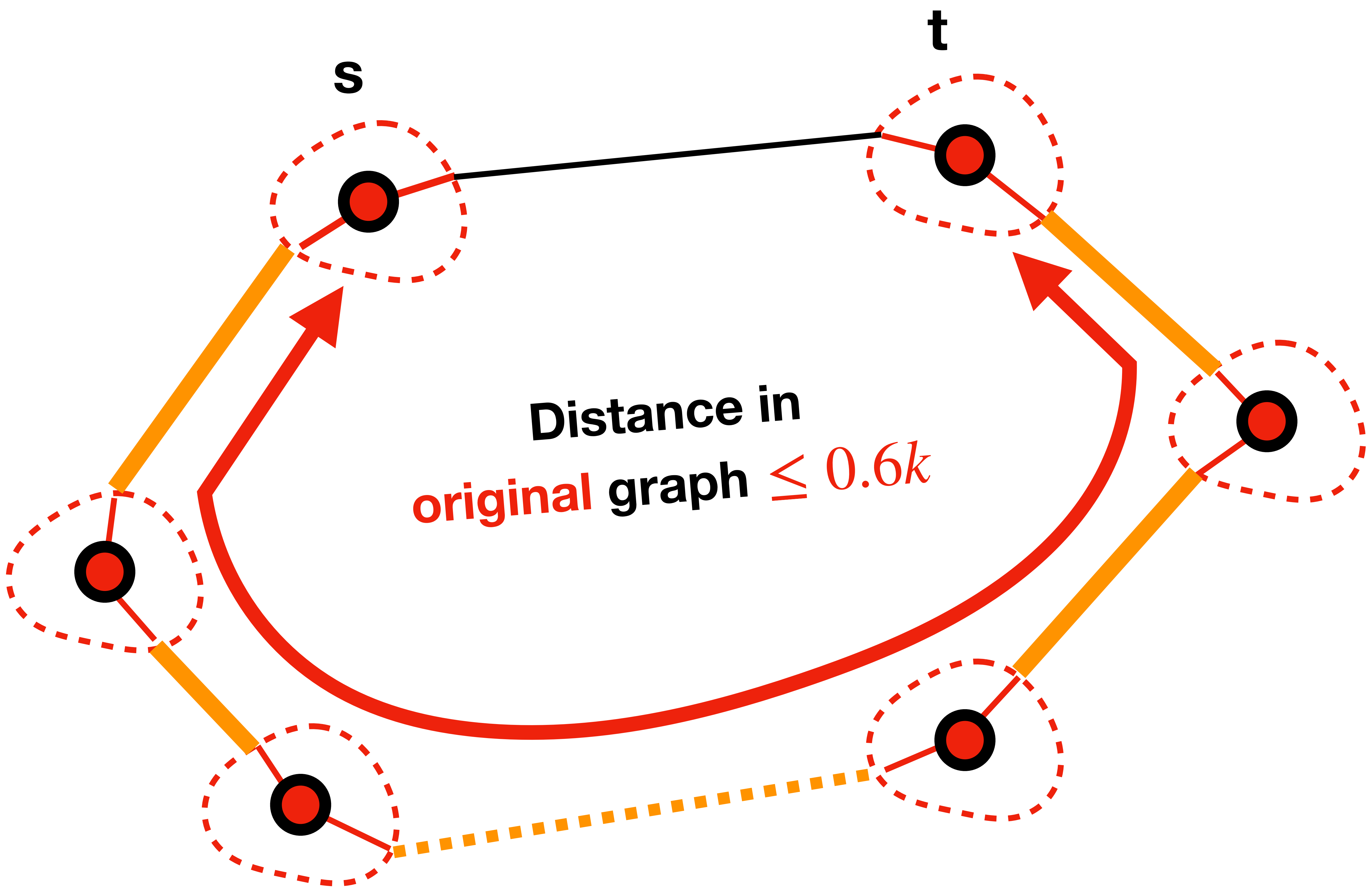
2. Contract stars around hitting set vertices to single nodes

3. Build a **0.2k-stretch spanner** on the contracted graph [DFKL'21]

Stretch analysis:

$$\mathrm{dist}(s, t) \leq 0.2k \cdot 3 = 0.6k$$

Analysis of size & stretch



Spanner on the **hitting set A**

1. Assign high-deg vertices to neighbor hitting set vertices
2. Contract stars around hitting set vertices to single nodes
3. Build a **0.2k-stretch spanner** on the contracted graph [DFKL'21]

Stretch analysis:

$\mathrm{dist}(s, t) \leq 0.2k \cdot 3 = 0.6k$

Outline of main algorithm

Degree reduction

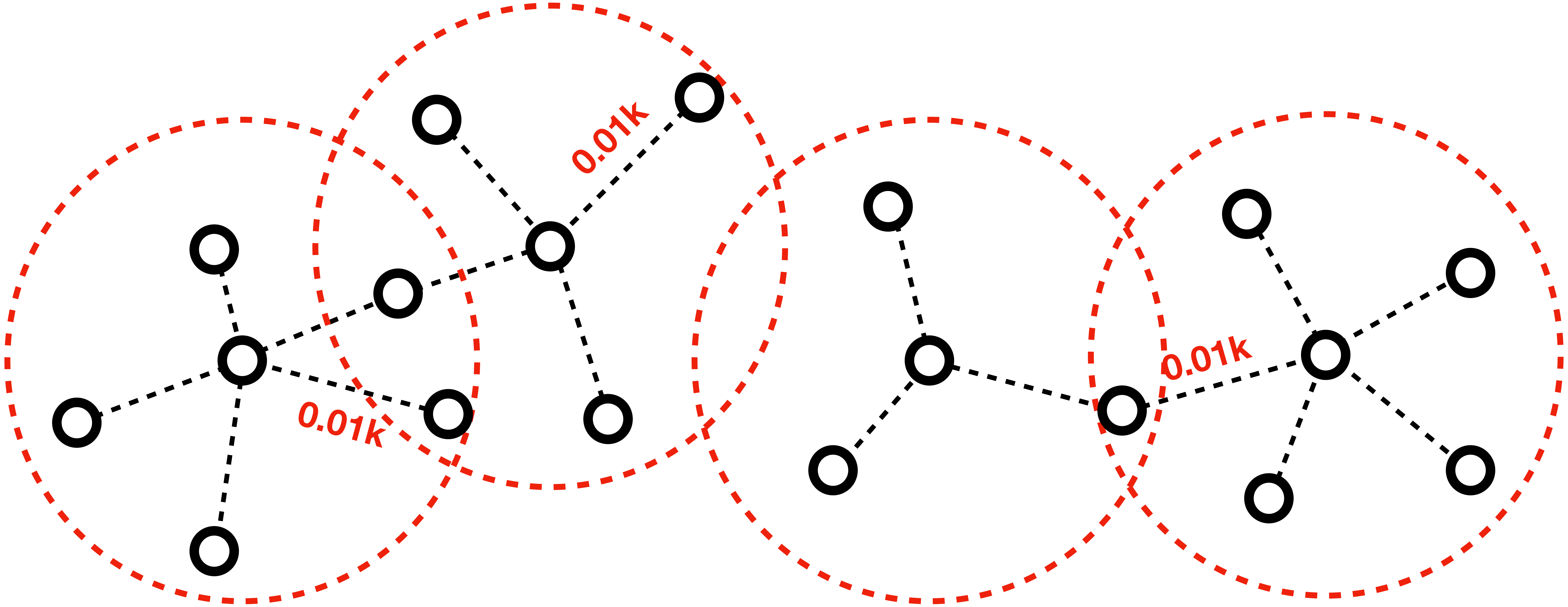
Neighborhood computation

Parallel spanner simulation

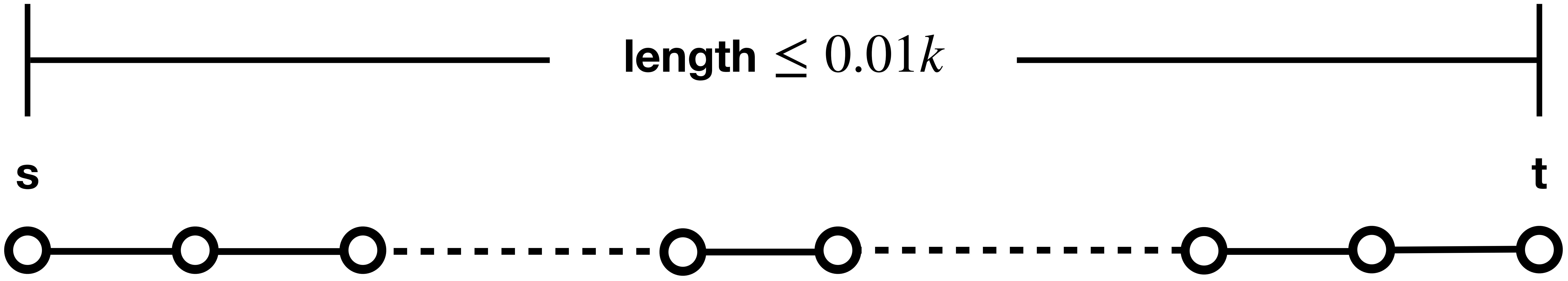
Neighborhood Computation

Input: a graph with max-deg $\leq n^{20/k}$

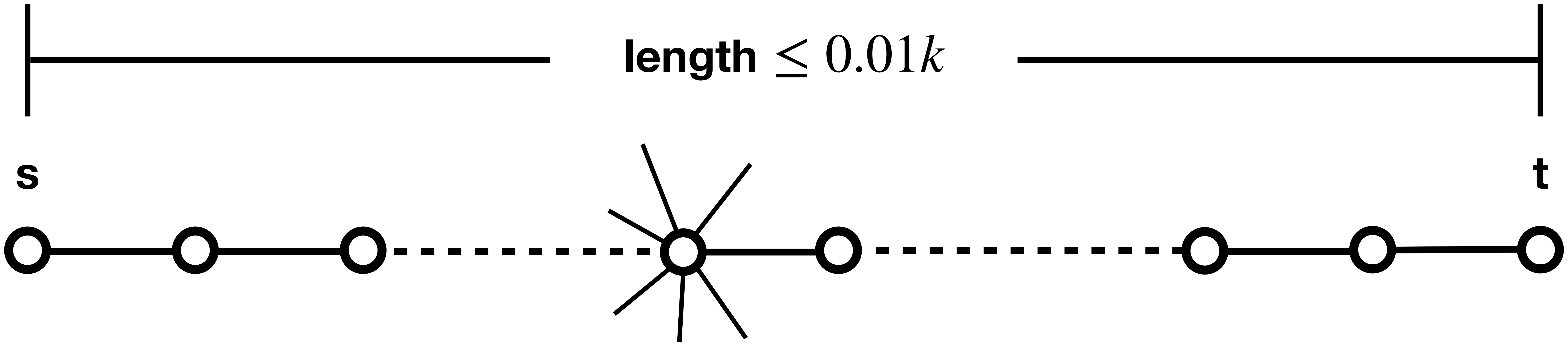
Output: compute the $0.01k$-neighborhood of each vertex



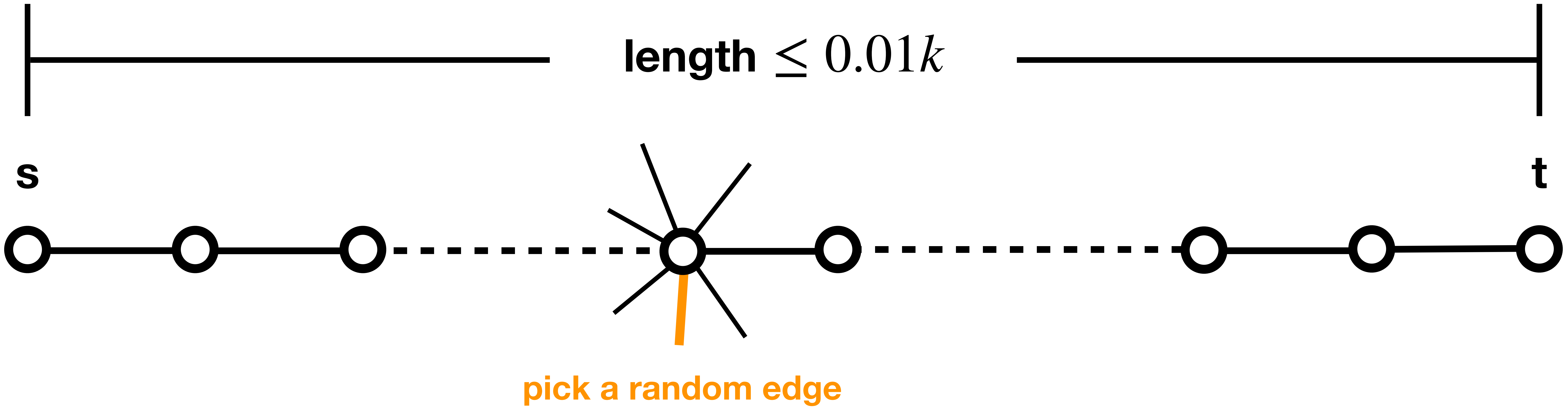
Neighborhood Computation



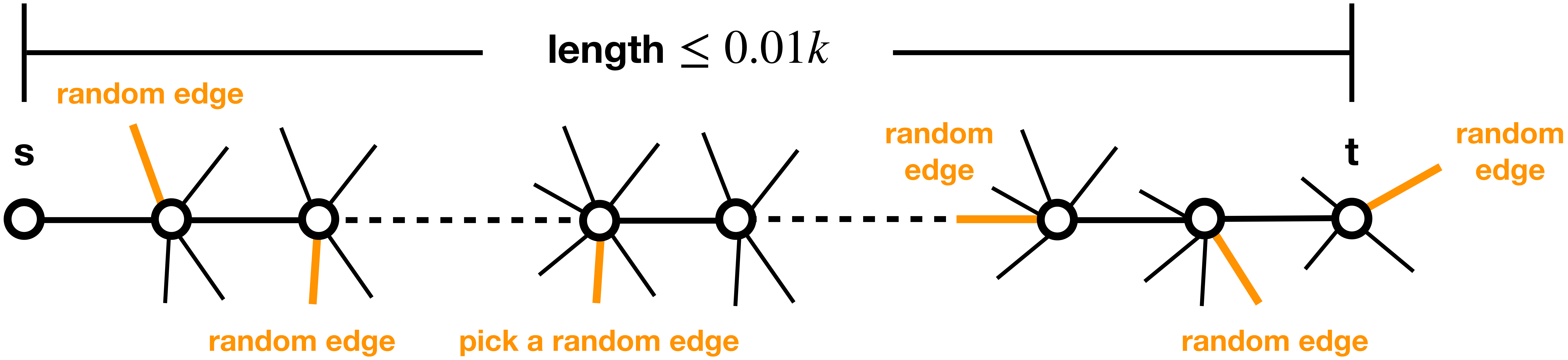
Neighborhood Computation



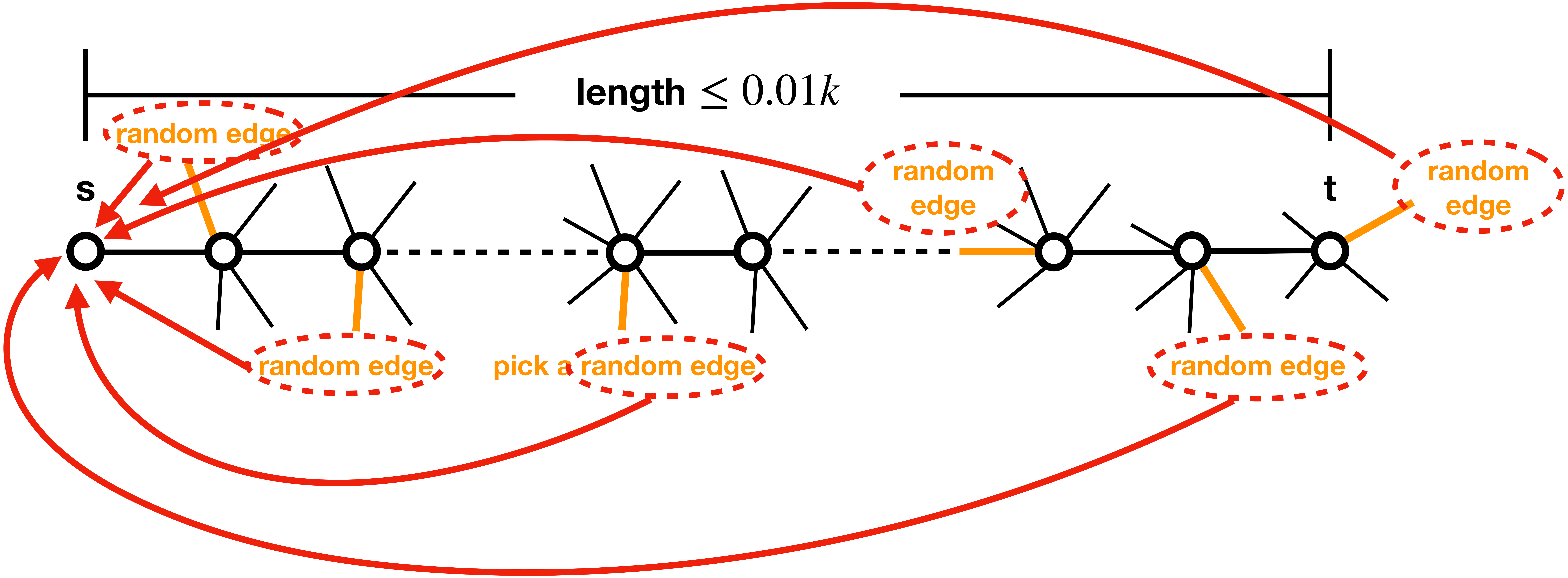
Neighborhood Computation



Neighborhood Computation

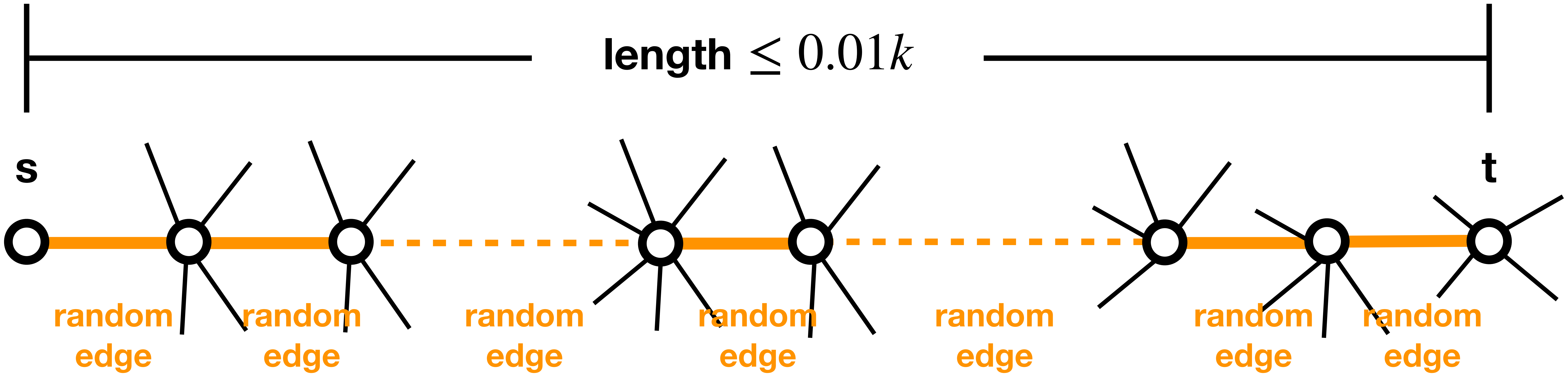


Neighborhood Computation



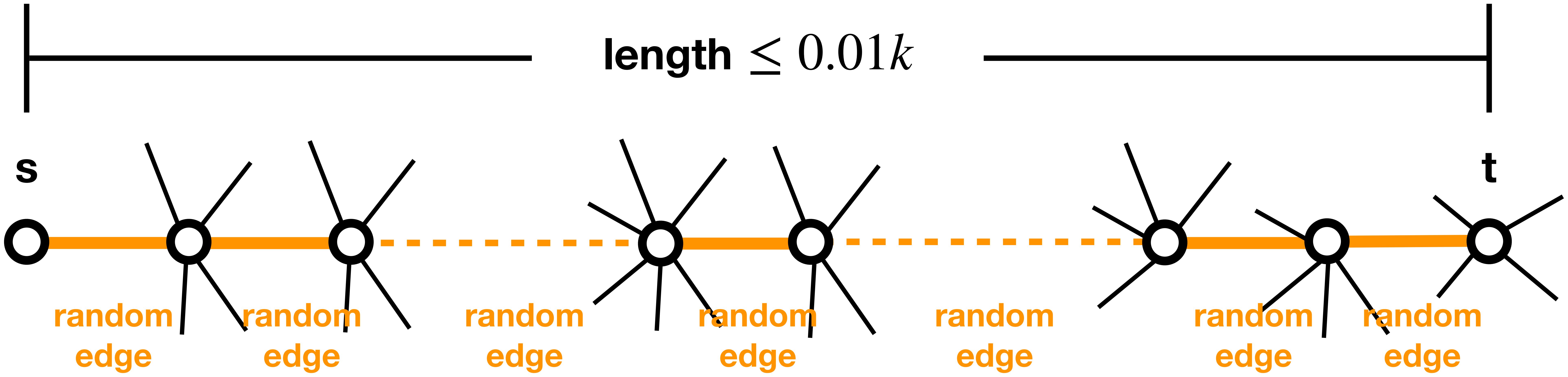
Send all random edges to source vertex s in 1 round

Neighborhood Computation



What is the **probability** that the **s-t path is picked**?

Neighborhood Computation

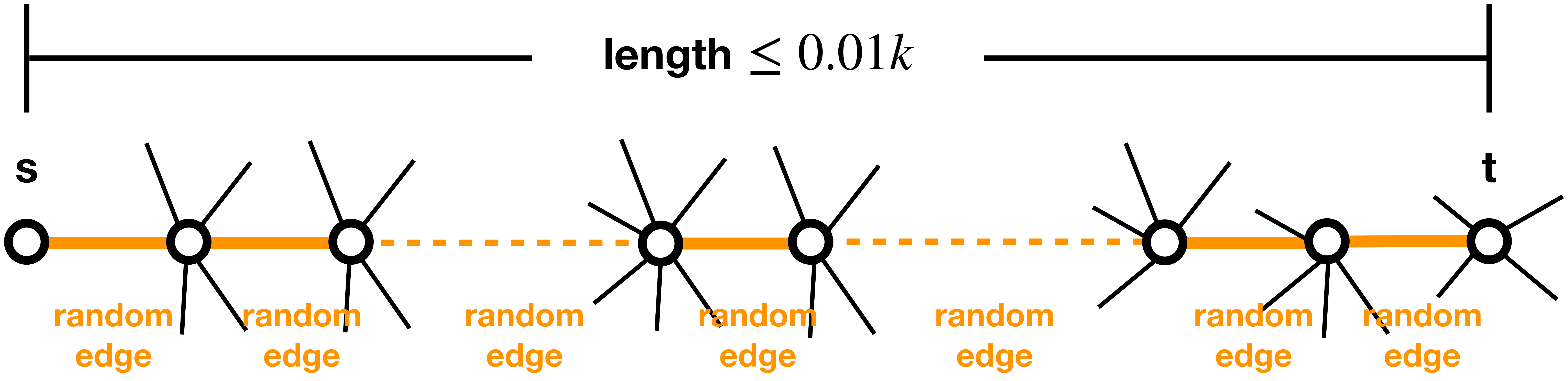


What is the **probability** that the **s-t path is picked**?

- $\Pr[\text{st-path is picked}] \geq (1/\deg)^{0.01k} \geq n^{-20/k \cdot 0.01k} = n^{-0.2}$

(distributed) random walk

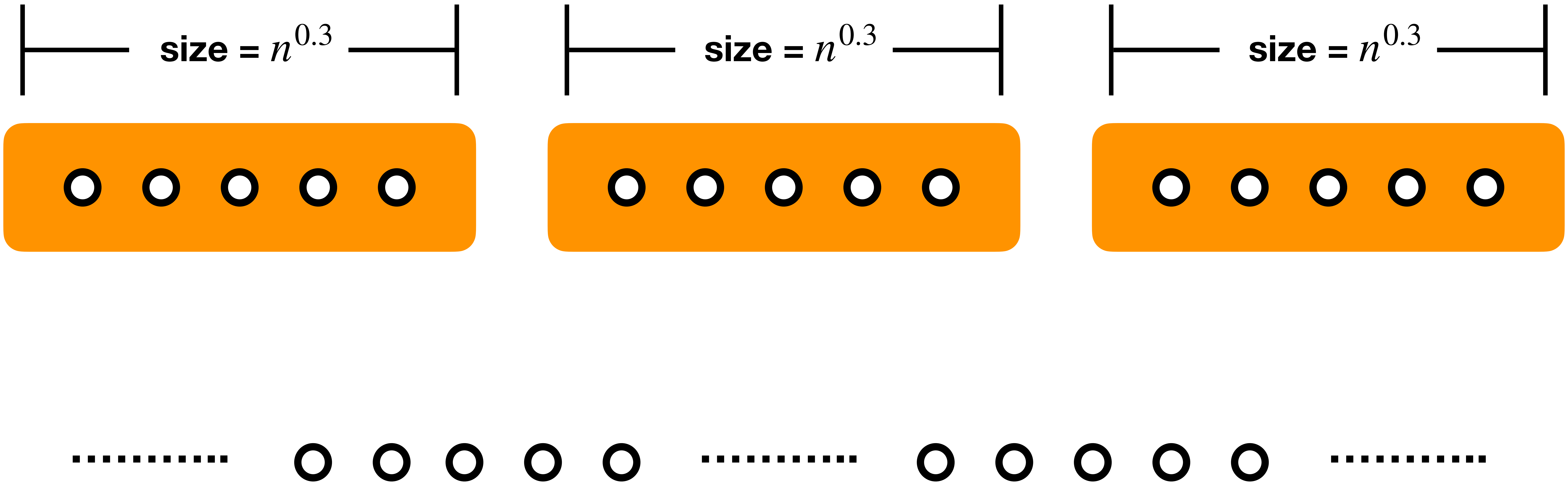
Neighborhood Computation



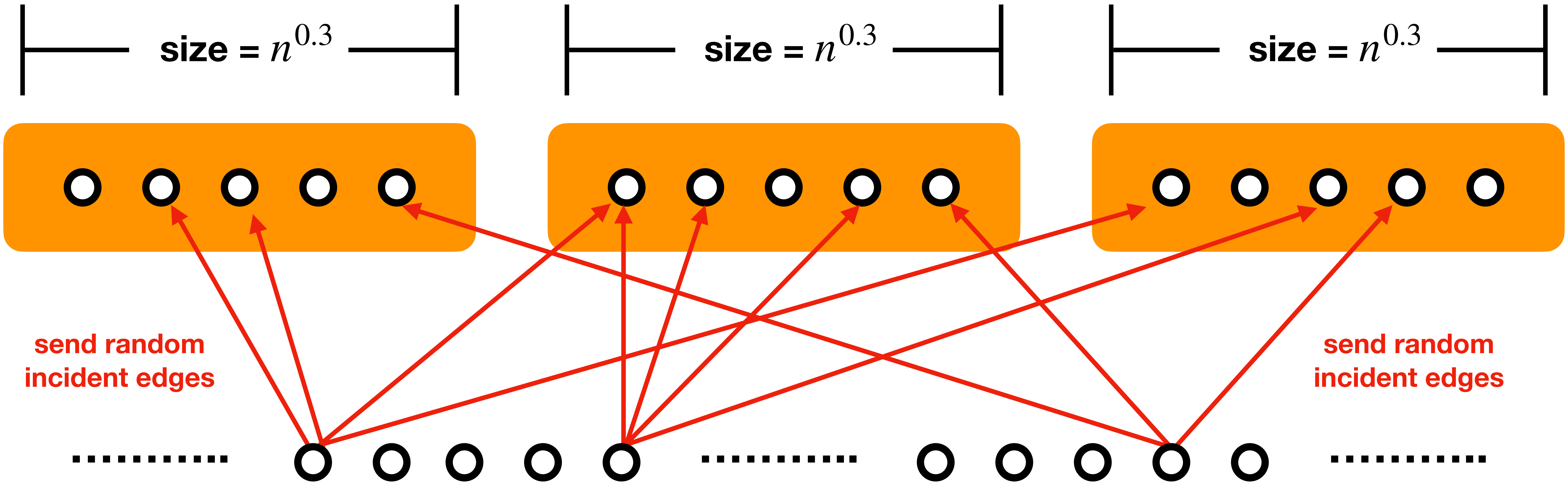
What is the **probability** that the **s-t path is picked**?

- $\Pr[\text{st-path is picked}] \geq (1/\deg)^{0.01k} \geq n^{-20/k \cdot 0.01k} = n^{-0.2}$
- How to boost success probability?

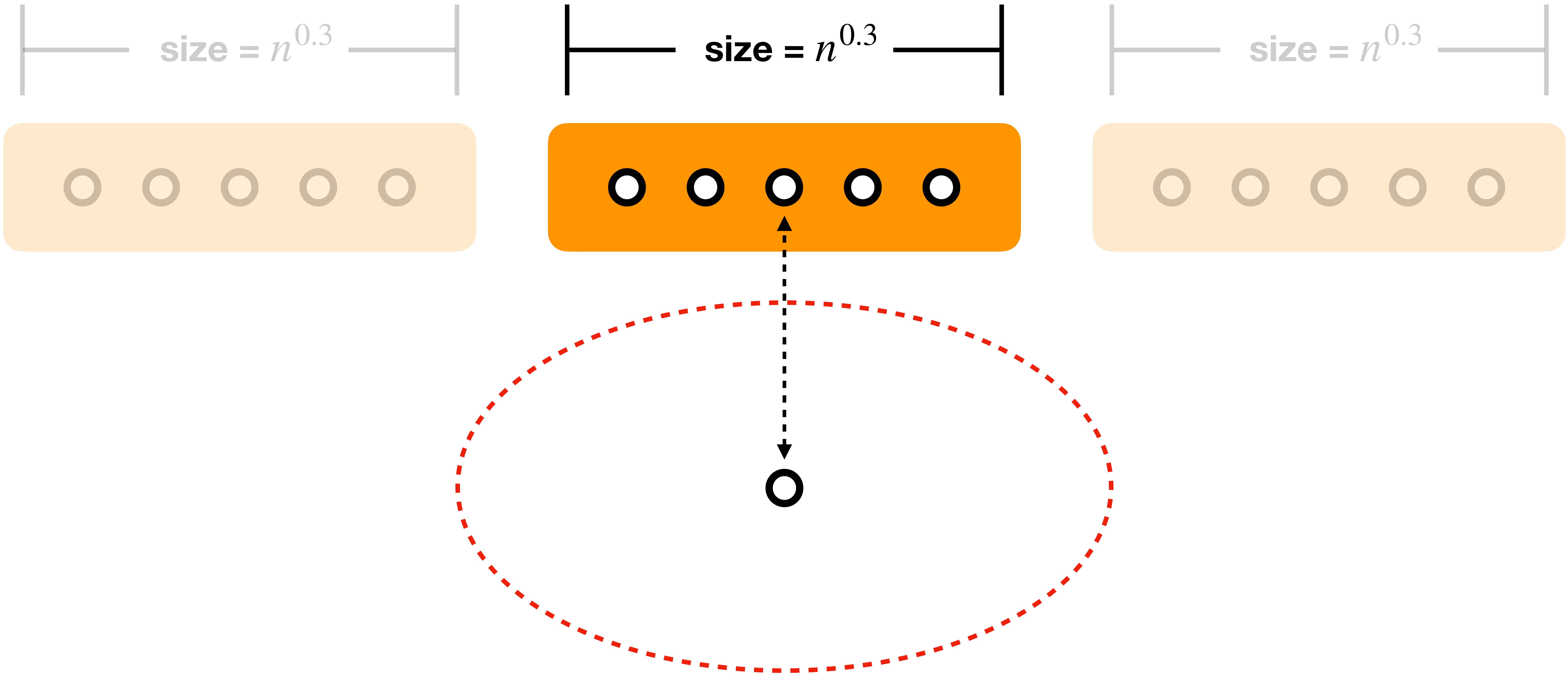
Boosting by replication



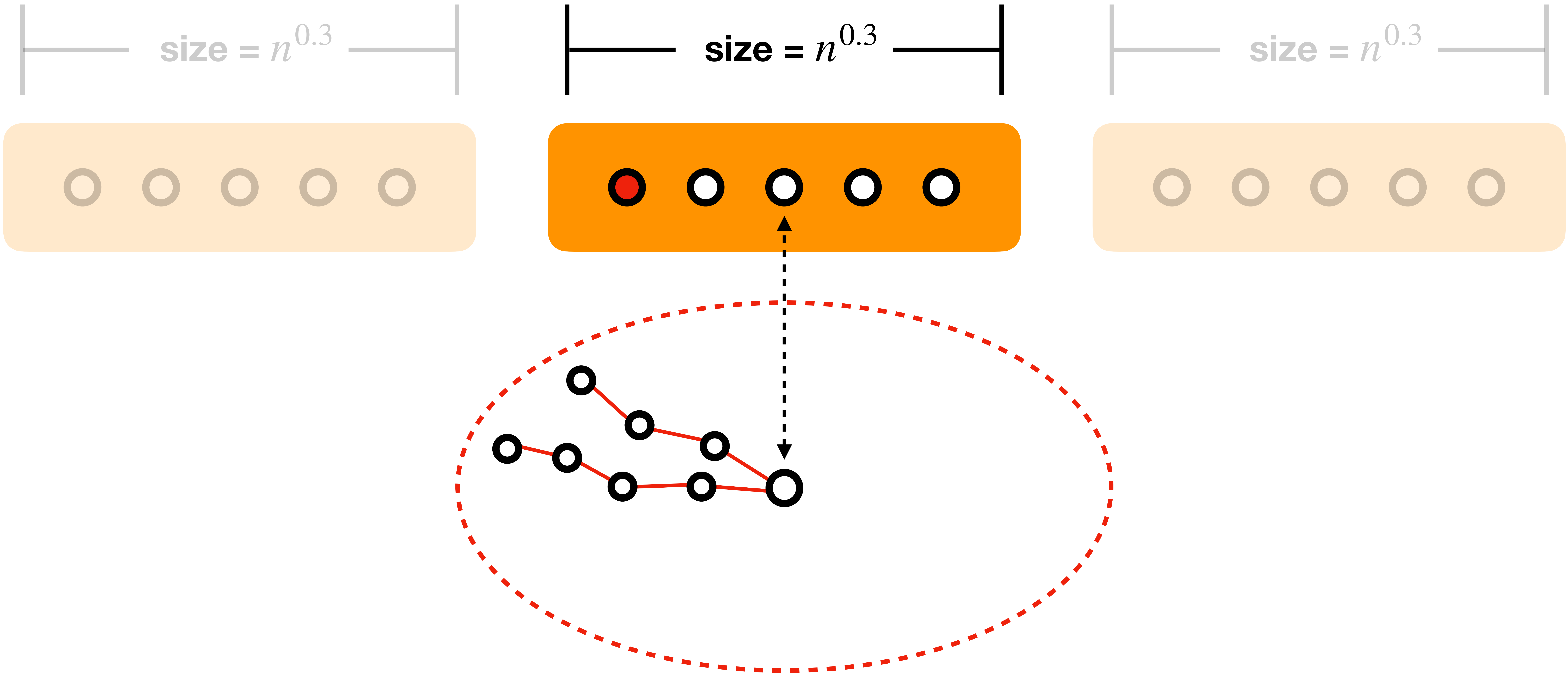
Boosting by replication



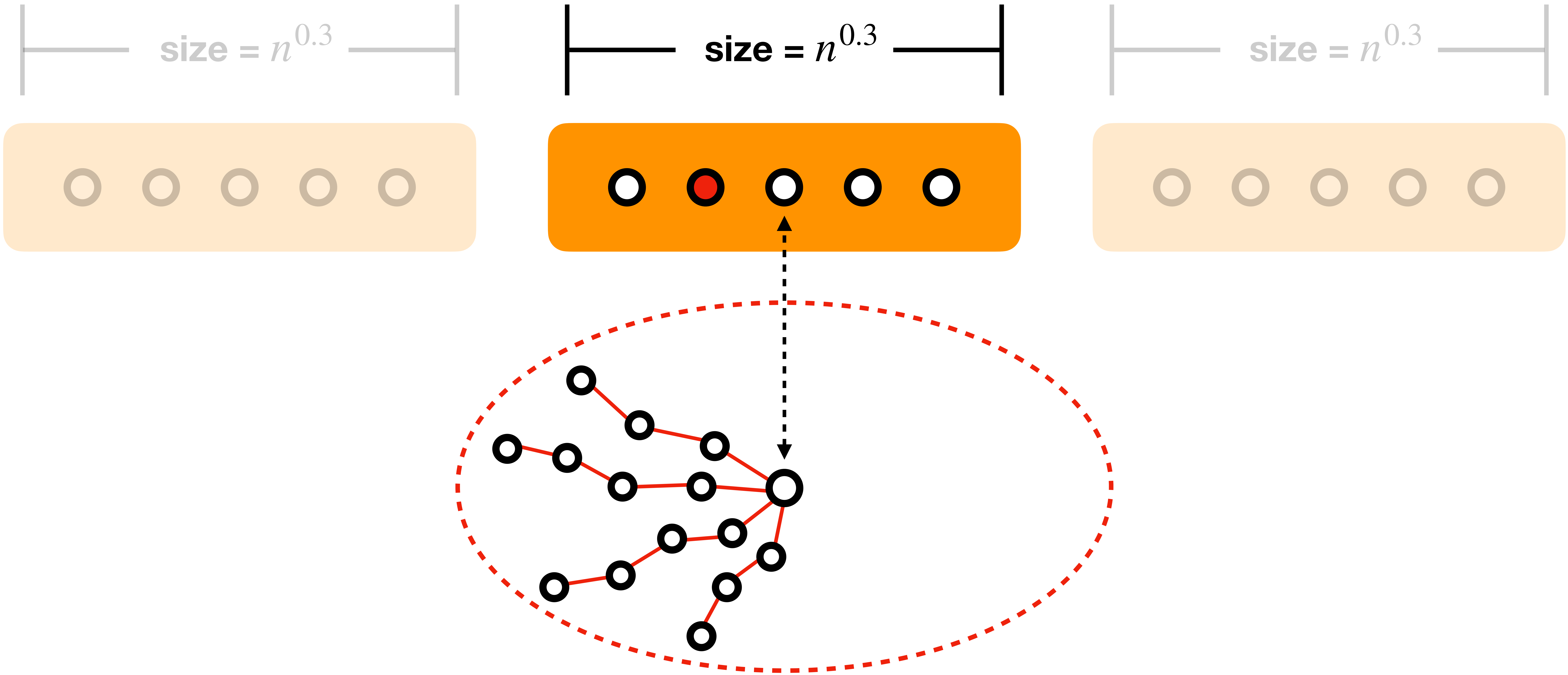
Boosting by replication



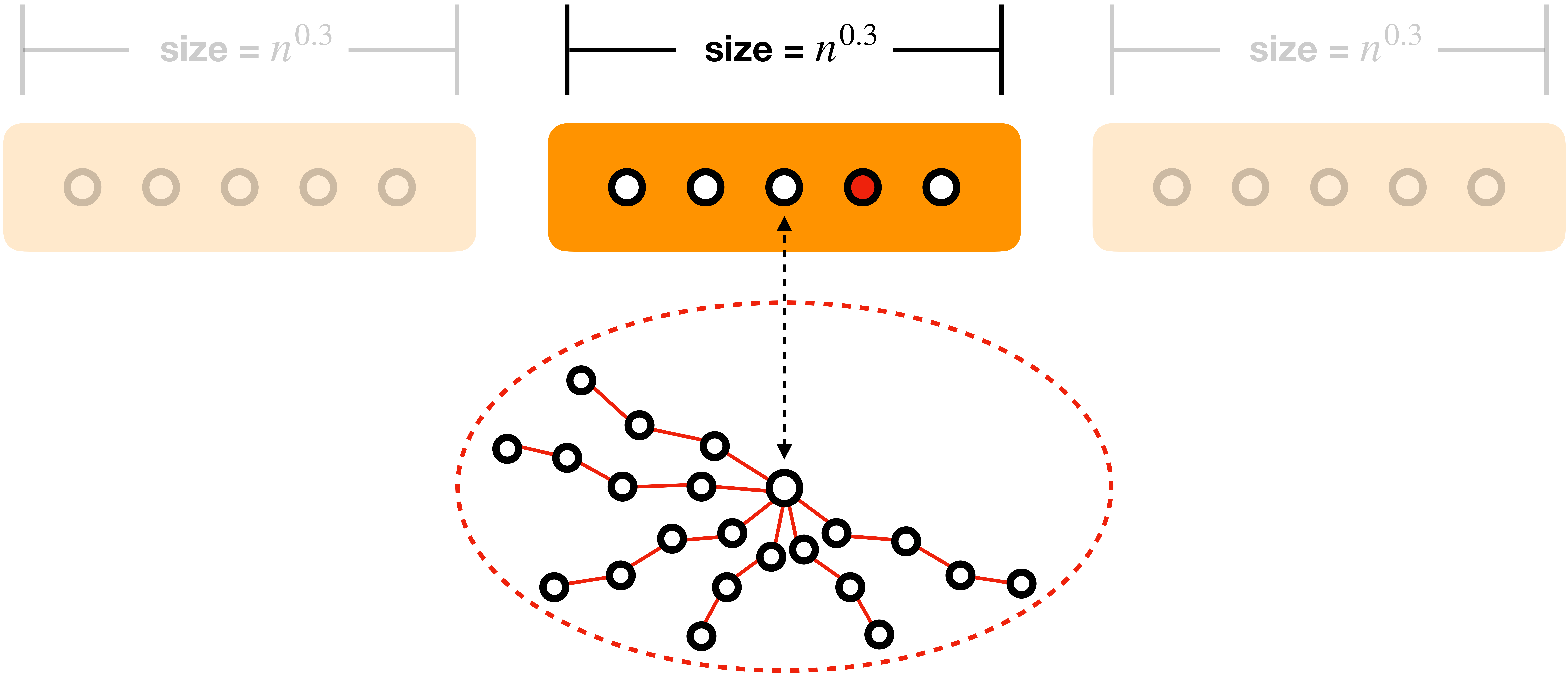
Boosting by replication



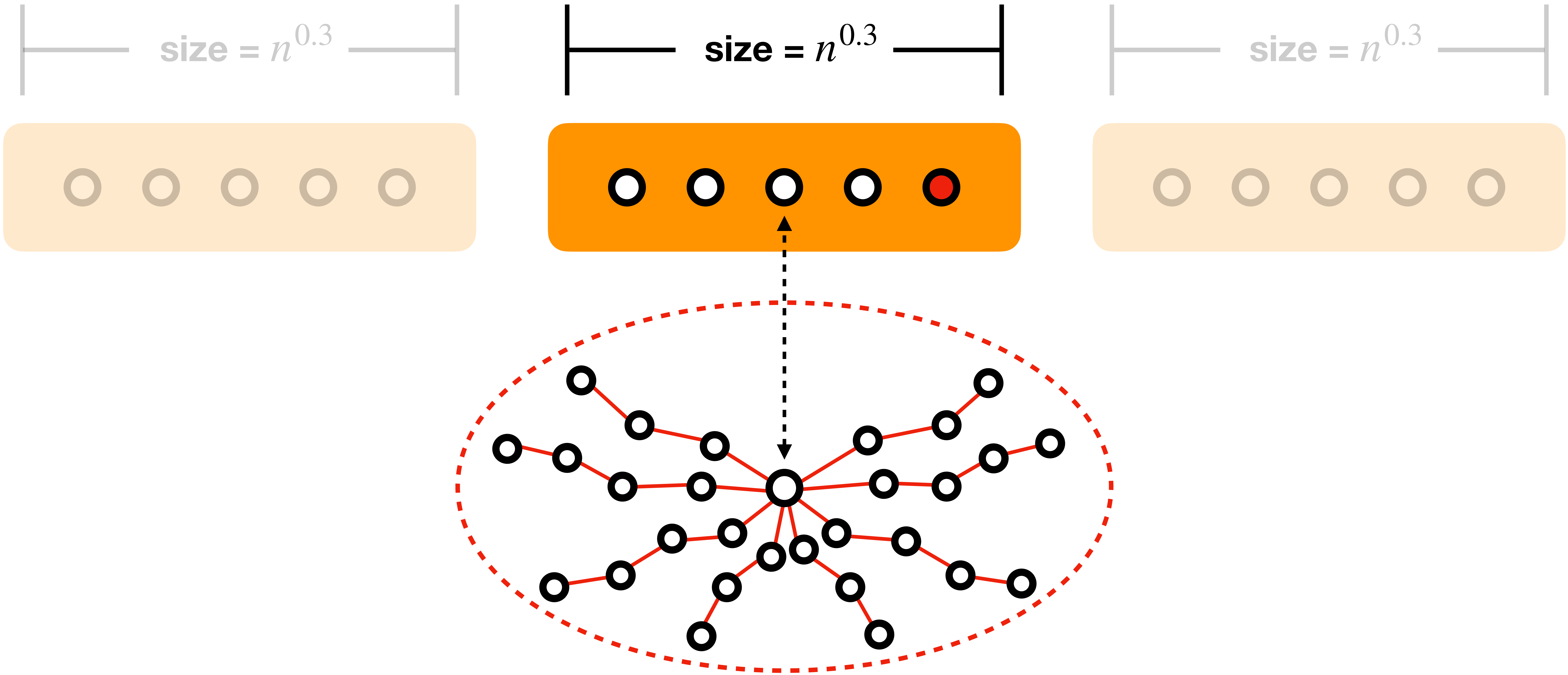
Boosting by replication



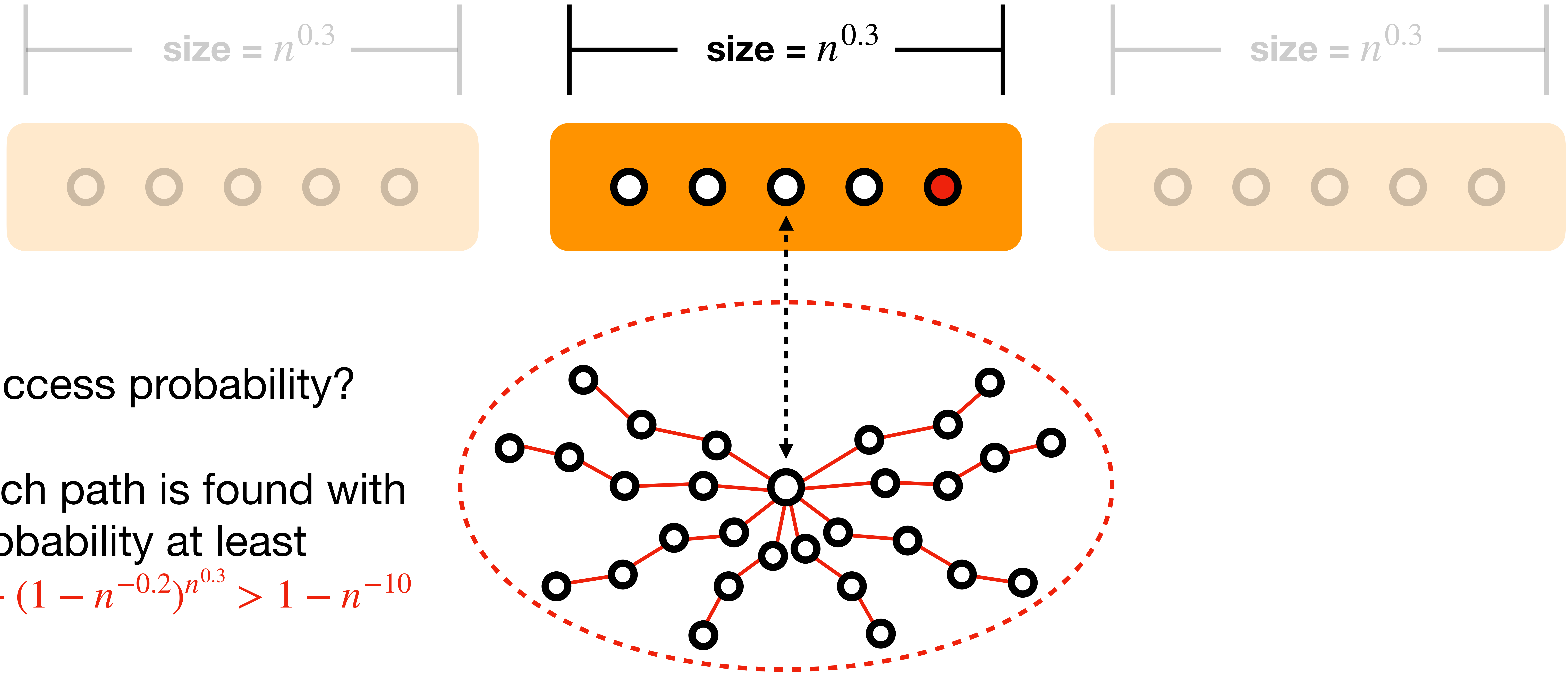
Boosting by replication



Boosting by replication



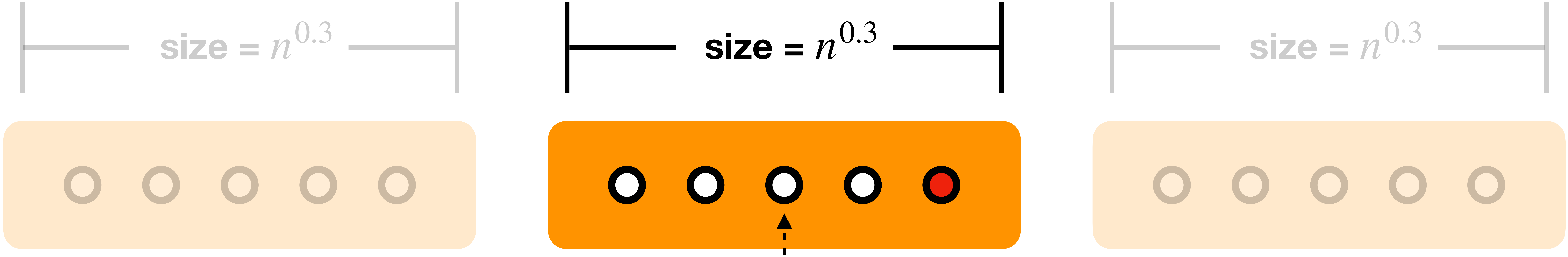
Boosting by replication

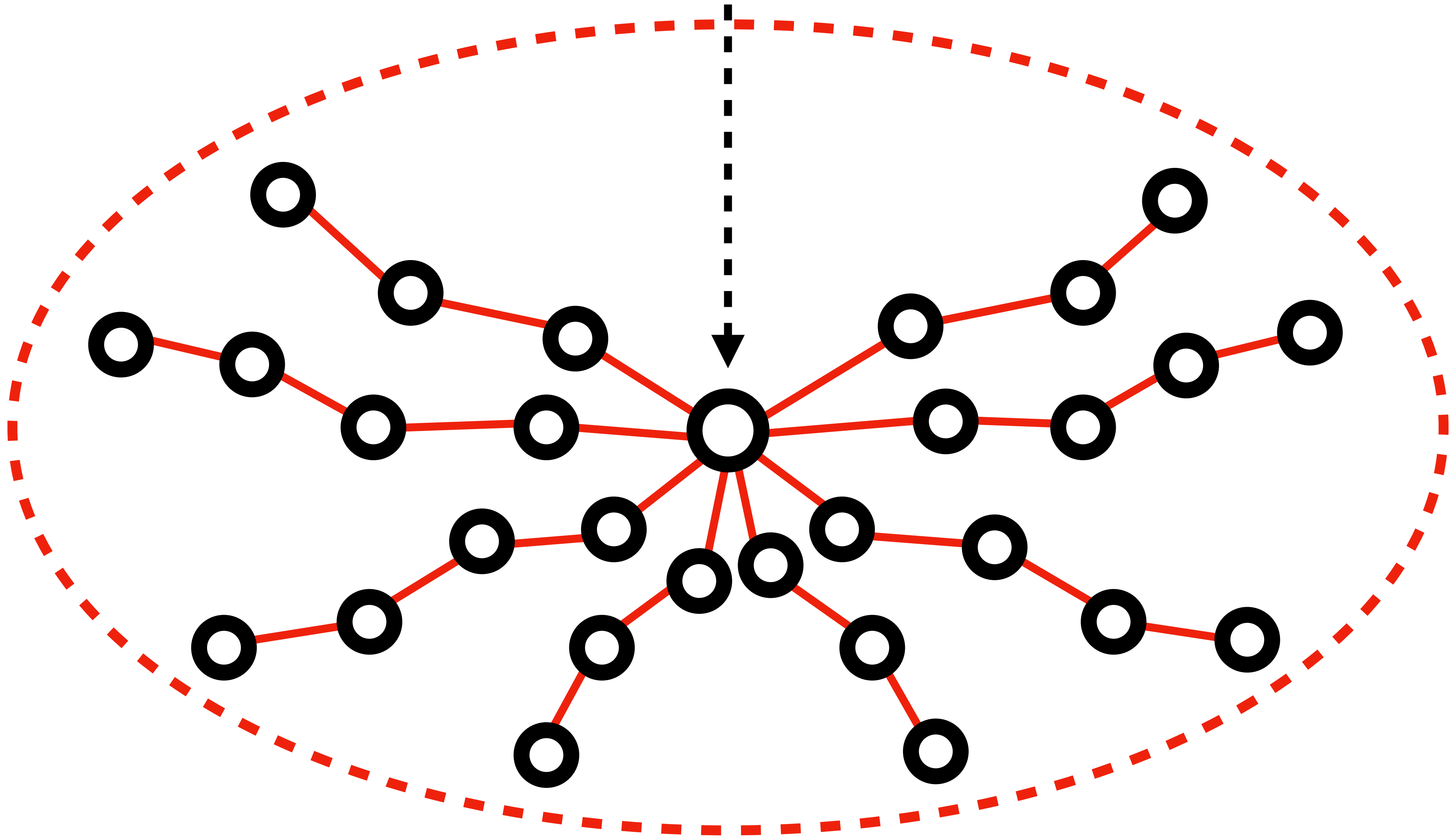


Success probability?

Each path is found with probability at least
$1-(1-n^{-0.2})^{n^{0.3}} > 1-n^{-10}$

Boosting by replication





Success probability?

Each path is found with probability at least $1-(1-n^{-0.2})^{n^{0.3}} > 1-n^{-10}$

Communication?

- Receive: $O(n)$
- Send: $n^{0.3} \cdot (\deg)^{0.01k} \leq n^{0.5}$

Outline of main algorithm

Degree reduction

Neighborhood computation

Parallel spanner simulation

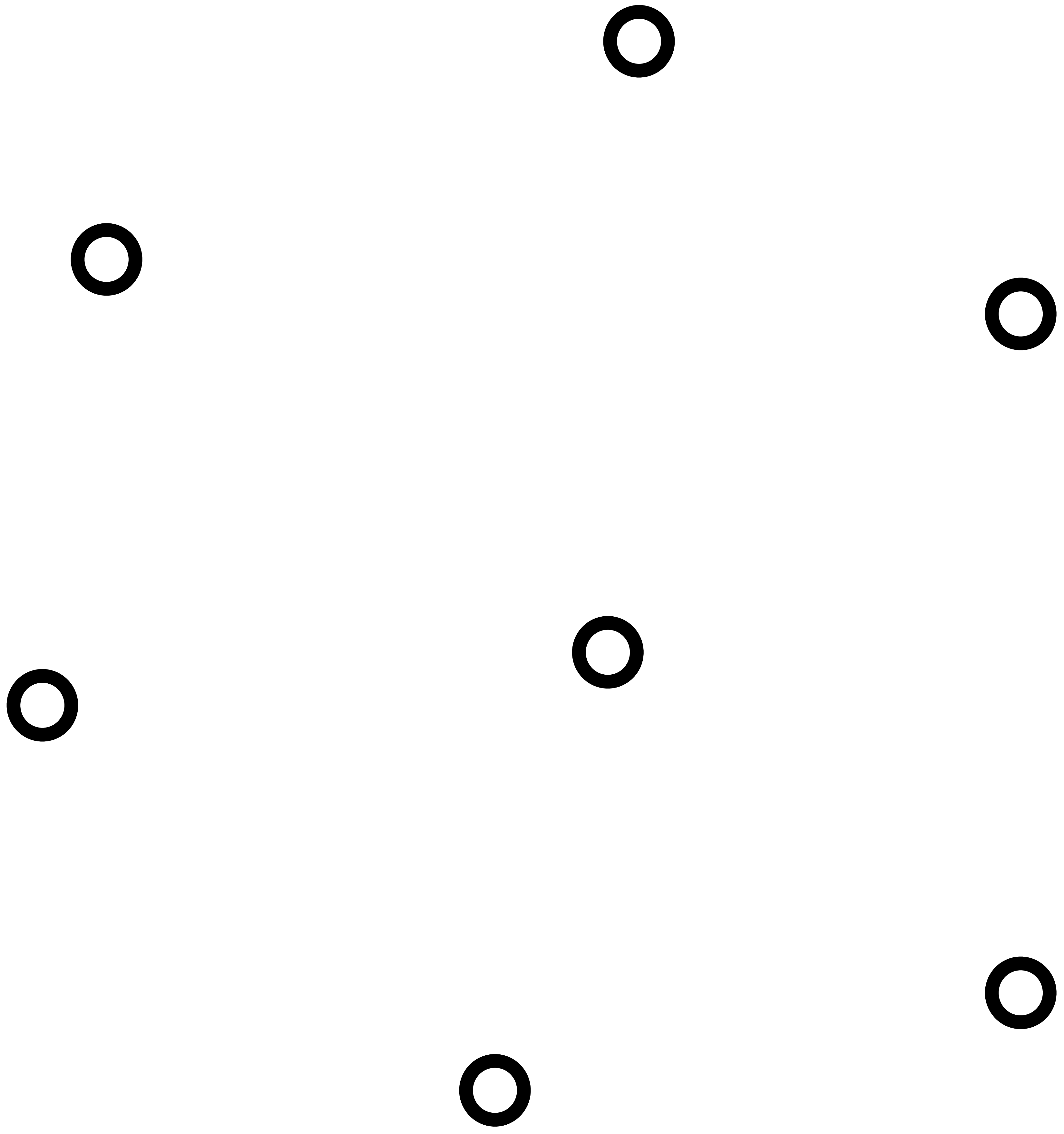
Parallel Spanner Simulation

A PRAM spanner algorithm
[Miller, Peng, Vladu, Xu, 2015]

1. Each vertex *v* takes a random value
   $r_v \sim \exp[\ln(10n)/k]$

2. Define $\text{shift}_u(v) = \text{dist}(u, v) - r_v$
   and $\text{shift}_u = \min_{v \in V}\{\text{shift}_u(v)\}$

3. Add (u, w) to spanner, if
   $\text{shift}_u(v) \leq \text{shift}_u + 1$



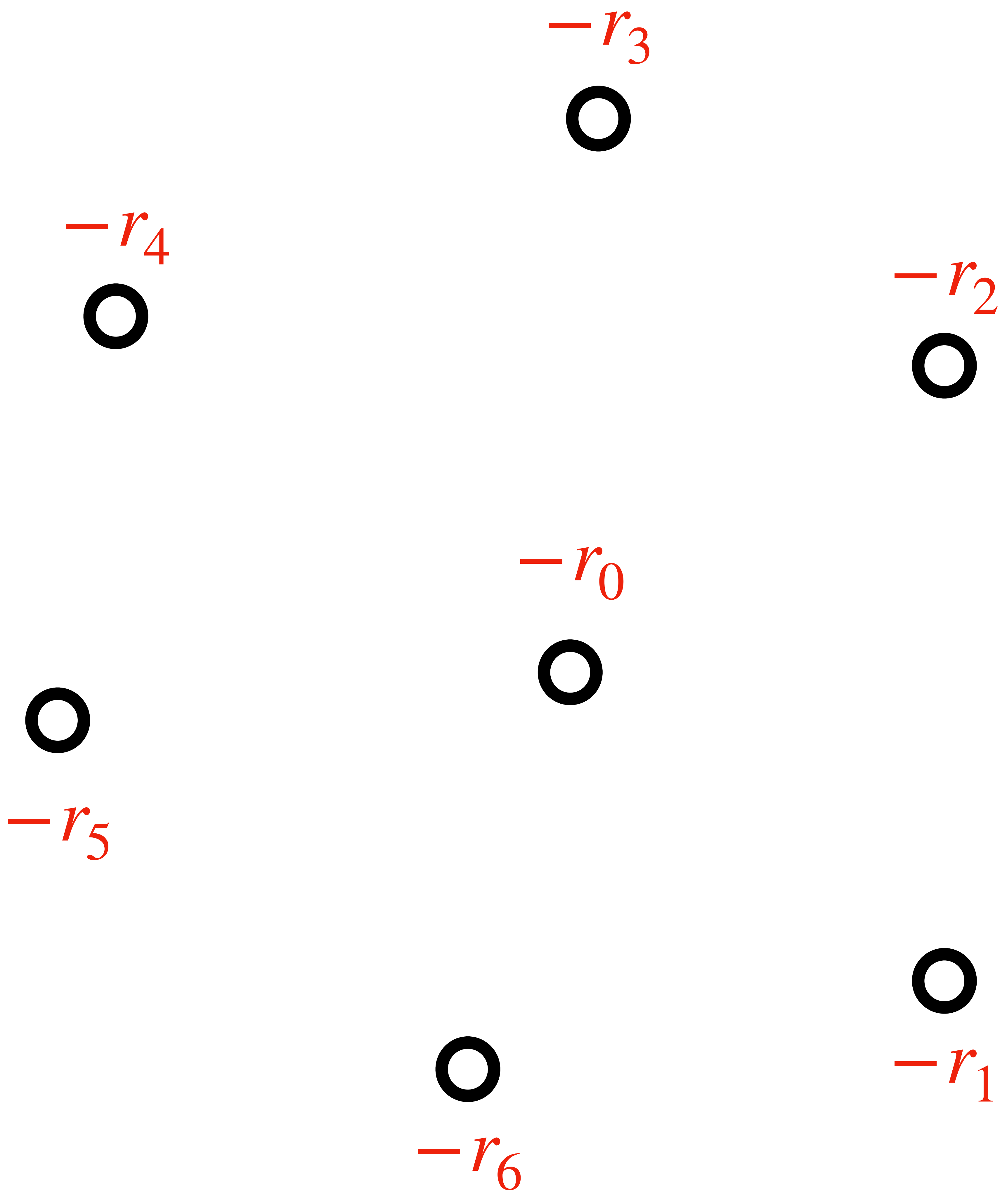
Parallel Spanner Simulation

A PRAM spanner algorithm
[Miller, Peng, Vladu, Xu, 2015]

1. **Each vertex *v* takes a random value**
   $r_v \sim \exp[\ln(10n)/k]$

2. Define $\mathrm{shift}_u(v) = \mathrm{dist}(u, v) - r_v$
   and $\mathrm{shift}_u = \min_{v \in V}\{\mathrm{shift}_u(v)\}$

3. Add (u, w) to spanner, if
   $\mathrm{shift}_u(v) \leq \mathrm{shift}_u + 1$



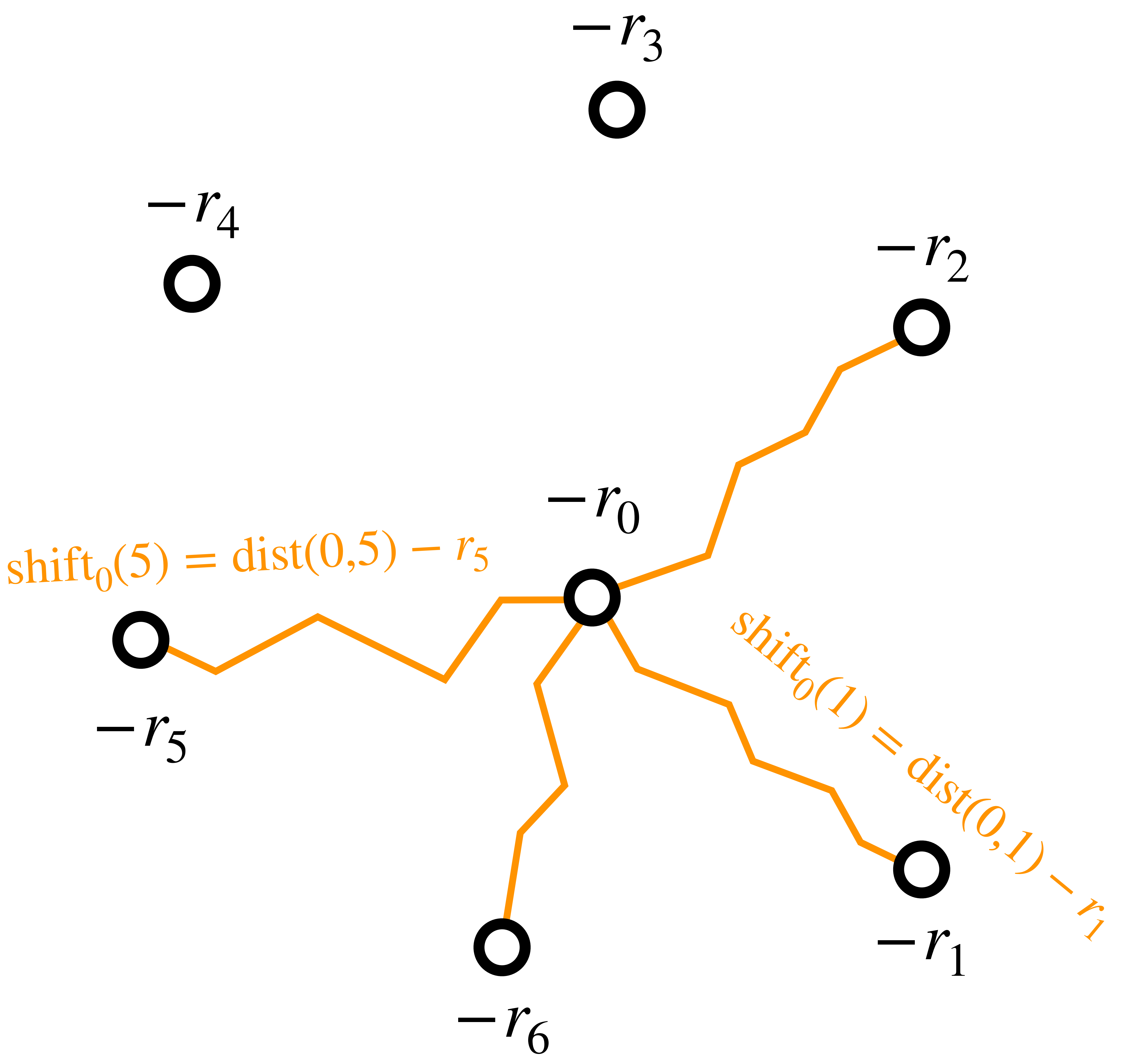
Parallel Spanner Simulation

A PRAM spanner algorithm
[Miller, Peng, Vladu, Xu, 2015]

1. Each vertex *v* takes a random value $r_v \sim \exp[\ln(10n)/k]$

2. **Define** $\text{shift}_u(v) = \text{dist}(u, v) - r_v$ **and** $\text{shift}_u = \min_{v \in V} \{\text{shift}_u(v)\}$

3. Add (u, w) to spanner, if $\text{shift}_u(v) \leq \text{shift}_u + 1$



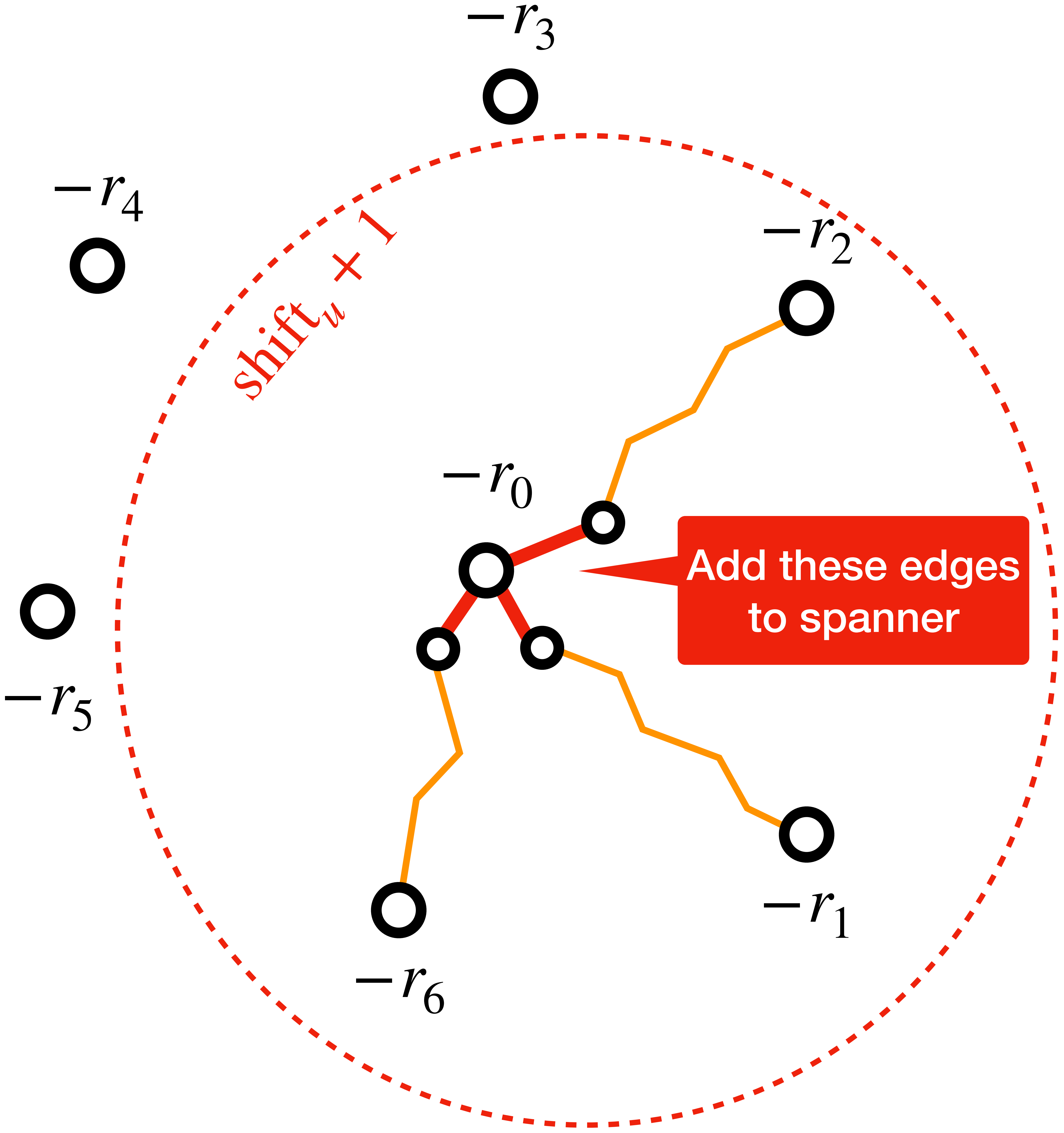
Parallel Spanner Simulation

A PRAM spanner algorithm
[Miller, Peng, Vladu, Xu, 2015]

1. Each vertex v takes a random value $r_v \sim \exp[\ln(10n)/k]$

2. Define $\text{shift}_u(v) = \text{dist}(u, v) - r_v$ and $\text{shift}_u = \min_{v \in V}\{\text{shift}_u(v)\}$

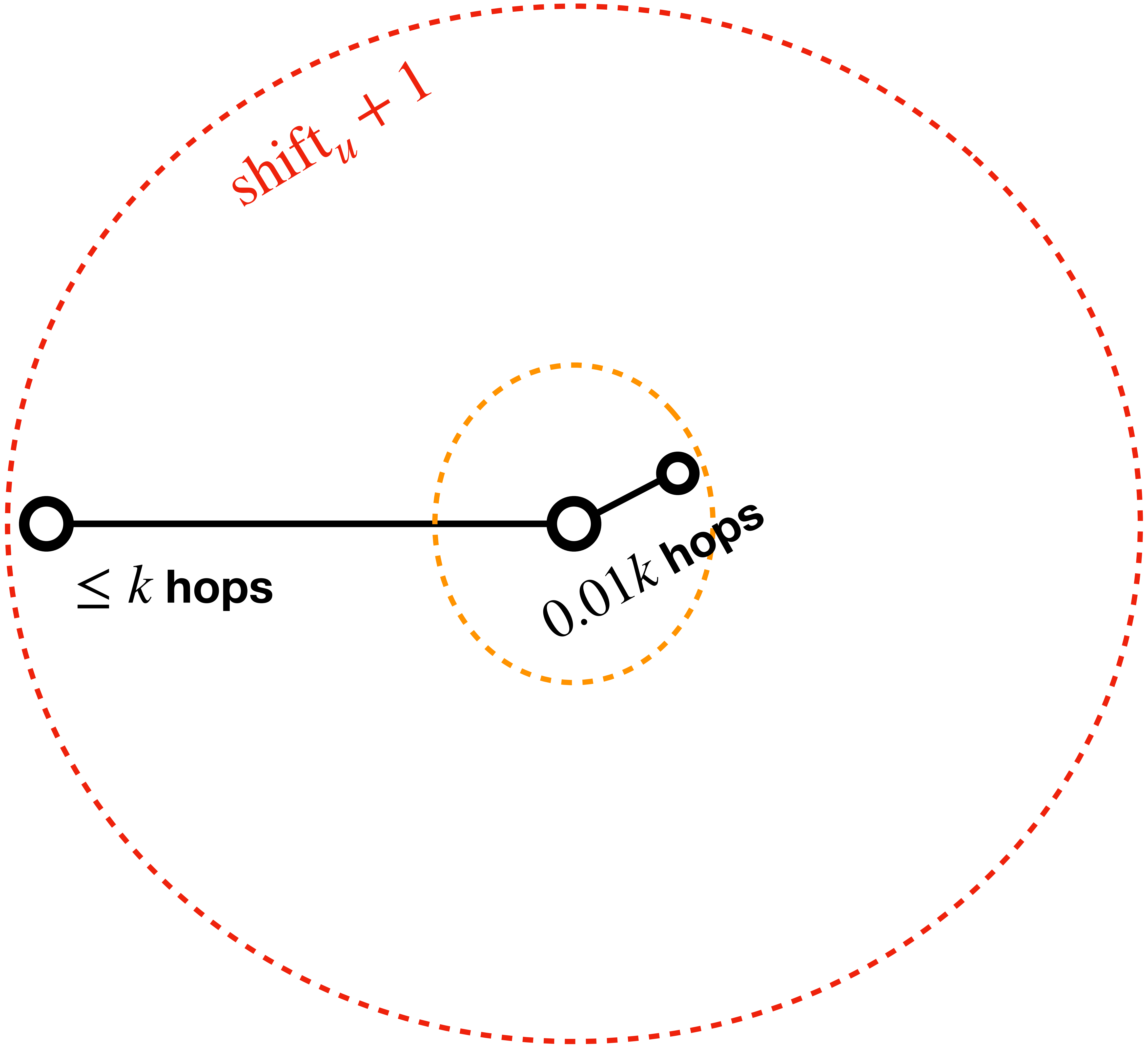
3. **Add (u, w) to spanner, if $\text{shift}_u(v) \leq \text{shift}_u + 1$**



Parallel Spanner Simulation

Simulating the parallel spanner

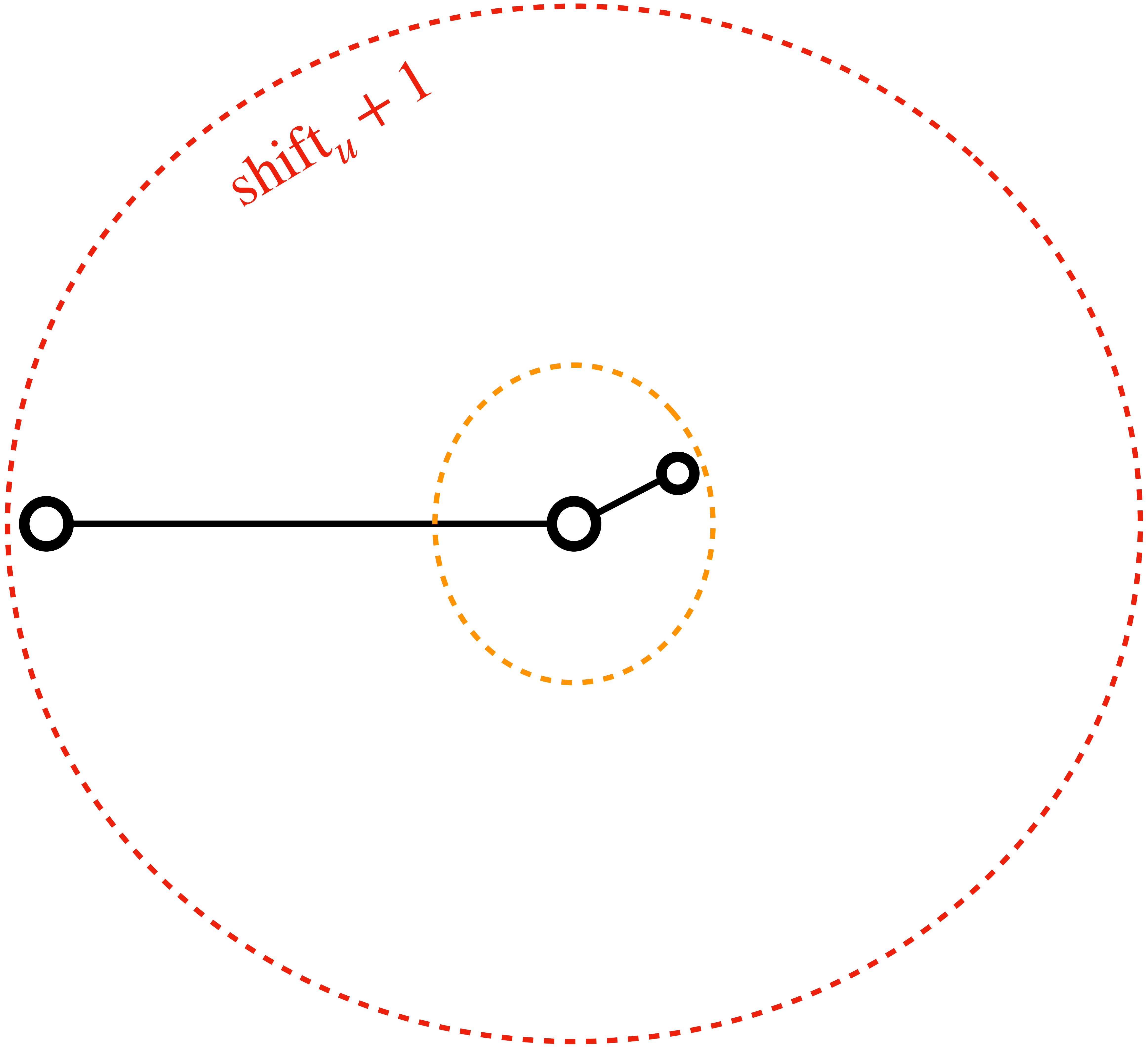
- The radius of $(\mathrm{shift}_u + 1)$**-area** is at most **k hops** [MPVX'15]
- Our neighborhood subroutine only computes **0.01k** ball



Parallel Spanner Simulation

Simulating the parallel spanner

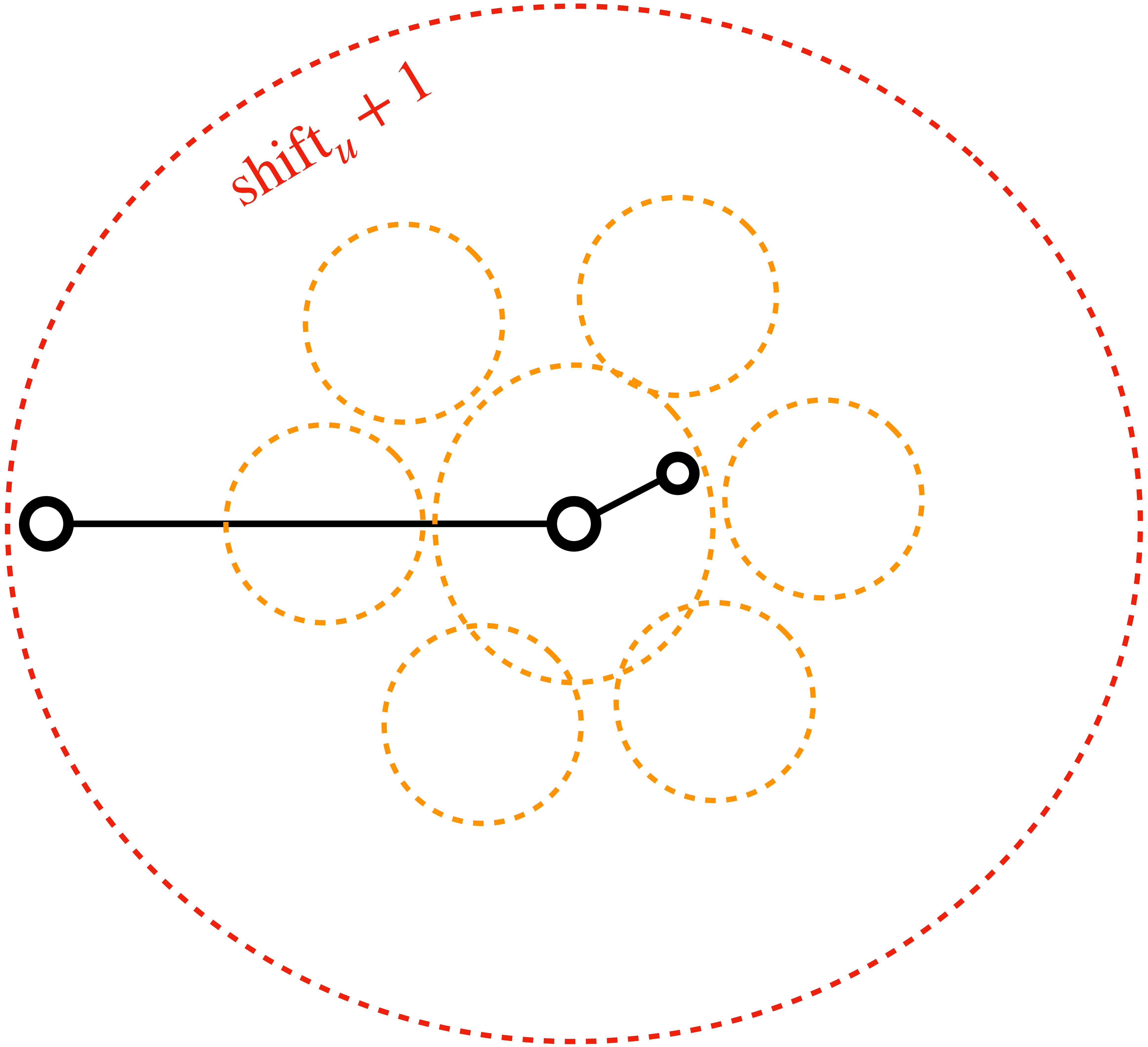
- The radius of $(\mathrm{shift}_u + 1)$**-area** is at most **k hops** [MPVX'15]
- Our neighborhood subroutine only computes **0.01k** ball
- **100-round** of communication can collect all relevant values of $\mathrm{shift}_u(v)$ within the $(\mathrm{shift}_u + 1)$**-area**



Parallel Spanner Simulation

Simulating the parallel spanner

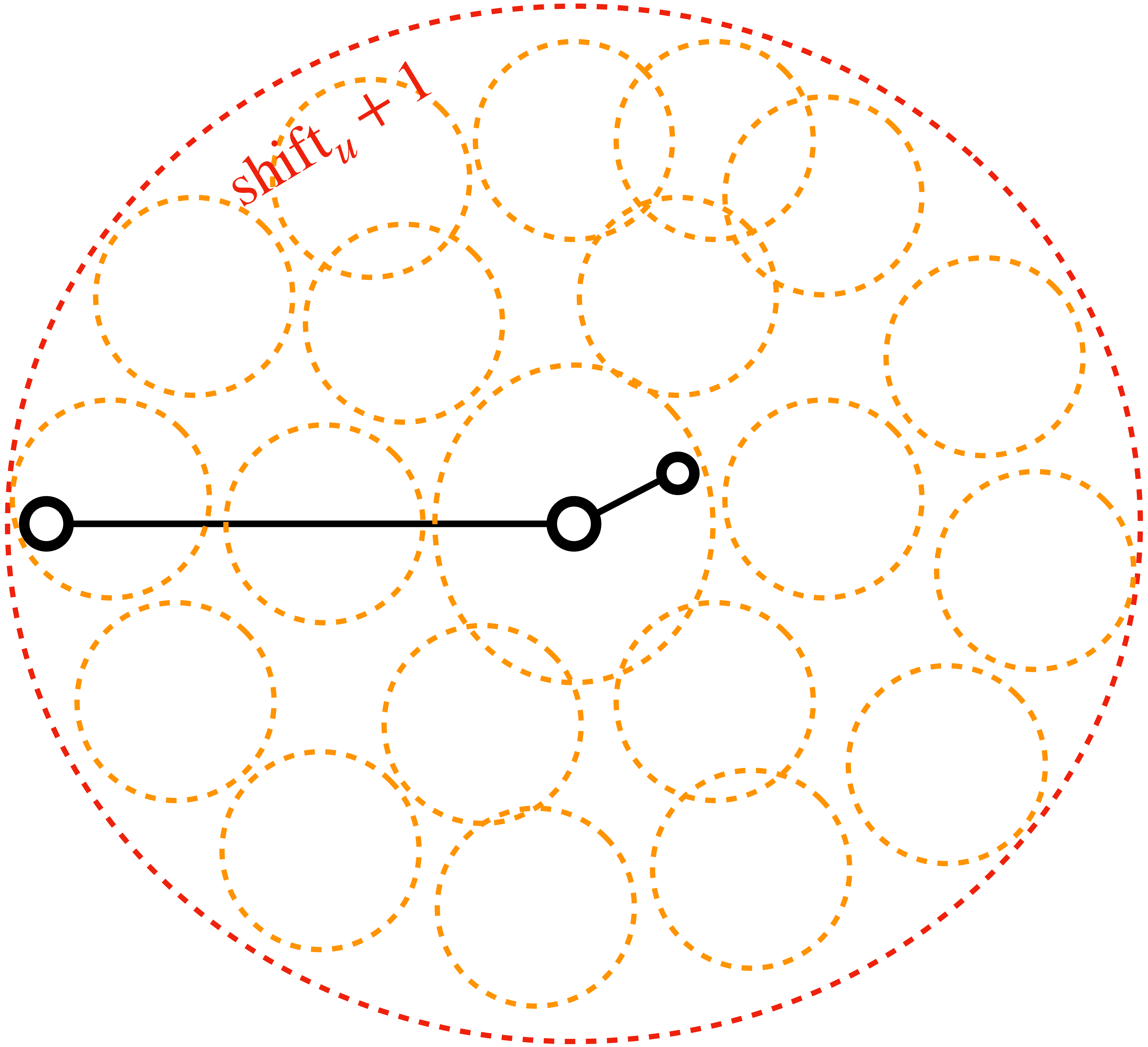
- The radius of $(\mathrm{shift}_u + 1)$**-area** is at most **k hops** [MPVX'15]
- Our neighborhood subroutine only computes **0.01k** ball
- **100-round** of communication can collect all relevant values of $\mathrm{shift}_u(v)$ within the $(\mathrm{shift}_u + 1)$**-area**



Parallel Spanner Simulation

Simulating the parallel spanner

- The radius of $(\text{shift}_u + 1)$**-area** is at most **k hops** [MPVX'15]
- Our neighborhood subroutine only computes **0.01k** ball
- **100-round** of communication can collect all relevant values of $\text{shift}_u(v)$ within the $(\text{shift}_u + 1)$**-area**



Further questions

Optimal spanners in weighted graphs in congested clique?
In our work, either $(1+\epsilon)(2k-1)$ stretch or $kn^{1+1/k}$ edges

1. Greedy spanner, highly sequential

2. $O(kn^{2+1/k})$ sequential runtime [Roditty & Zwick, 2004]