Oblivious Routing on Geometric Networks

Costas Busch, **Malik Magdon-Ismail** and Jing Xi

{buschc,magdon,xij2}@cs.rpi.edu

July 20, 2005.

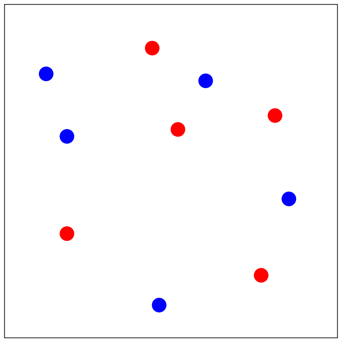
Rensselaer

Outline

- Oblivious Routing: Background and Our Contribution
- The Algorithm: Oblivious Routing with Single Intermediate Node
- Good Geometric (Metric) Embeddings; Examples
- Routing Result; Examples
- Discussion

Routing

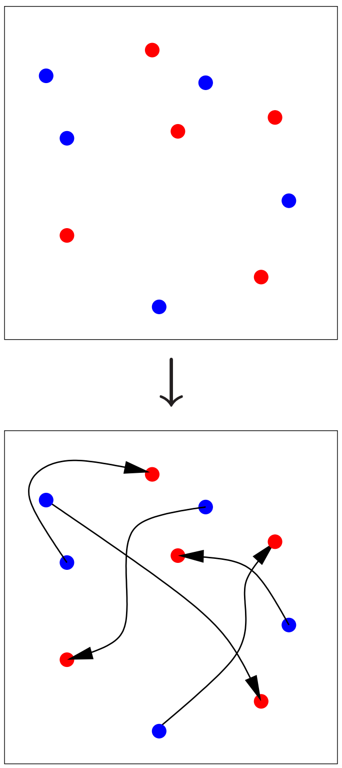
Routing: coustruct “good” paths given sources and destinations.



- Communication Networks – eg. Internet.
- Ad-hoc Networks – eg. sensor networks.
- Parallel Architectures – eg. Mesh.
- . . .

Routing

Routing: coustruct "good" paths given sources and destinations.



- Communication Networks – eg. Internet.
- Ad-hoc Networks – eg. sensor networks.
- Parallel Architectures – eg. Mesh.
- . . .

Wish List:

simple, scalable, efficient, near-optimal, general

Oblivious Routing

A packet's path is specified **independently** of other packets' paths.

– Suffices to specify algorithm for any single packet to select its path.
– Every packet uses this algorithm independent of other packets.

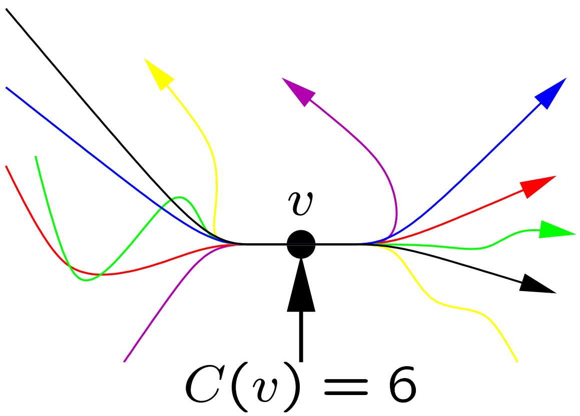
distributed, hence scalable;
applies to dynamic (online) setting with streaming packets;

(A packet π is a source-destination pair (s, t))

Optimality of Paths

Congestion

$C(v)$: # paths using node v

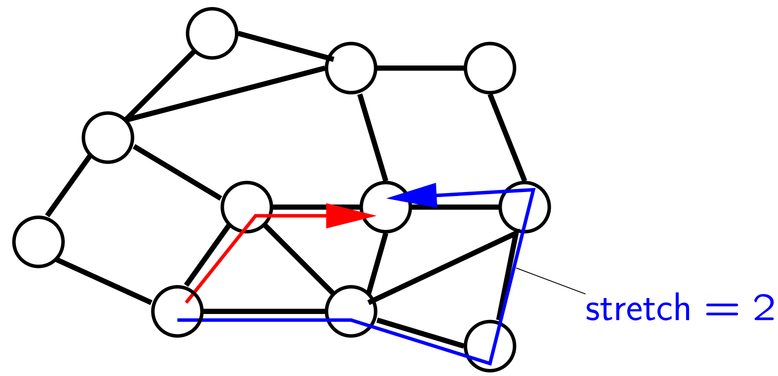


Congestion (C): $\max_e C(e)$

C^*: optimal congestion

Stretch

$\text{stretch}(\pi) : \dfrac{D(\pi)}{d(\pi)}$ $\leftarrow$ packet's path length, $\leftarrow$ shortest path length



stretch: $\max_\pi \text{stretch}(\pi)$

Optimal: $C = O(C^*)$; $\text{stretch} = O(1)$

(Similarily can define w.r.t. edge congestion C_{edge}.)

Srinivasan *et al.* [STOC97]: Near-optimal; *offline; non-oblivious.*

Related Work – Opt. C

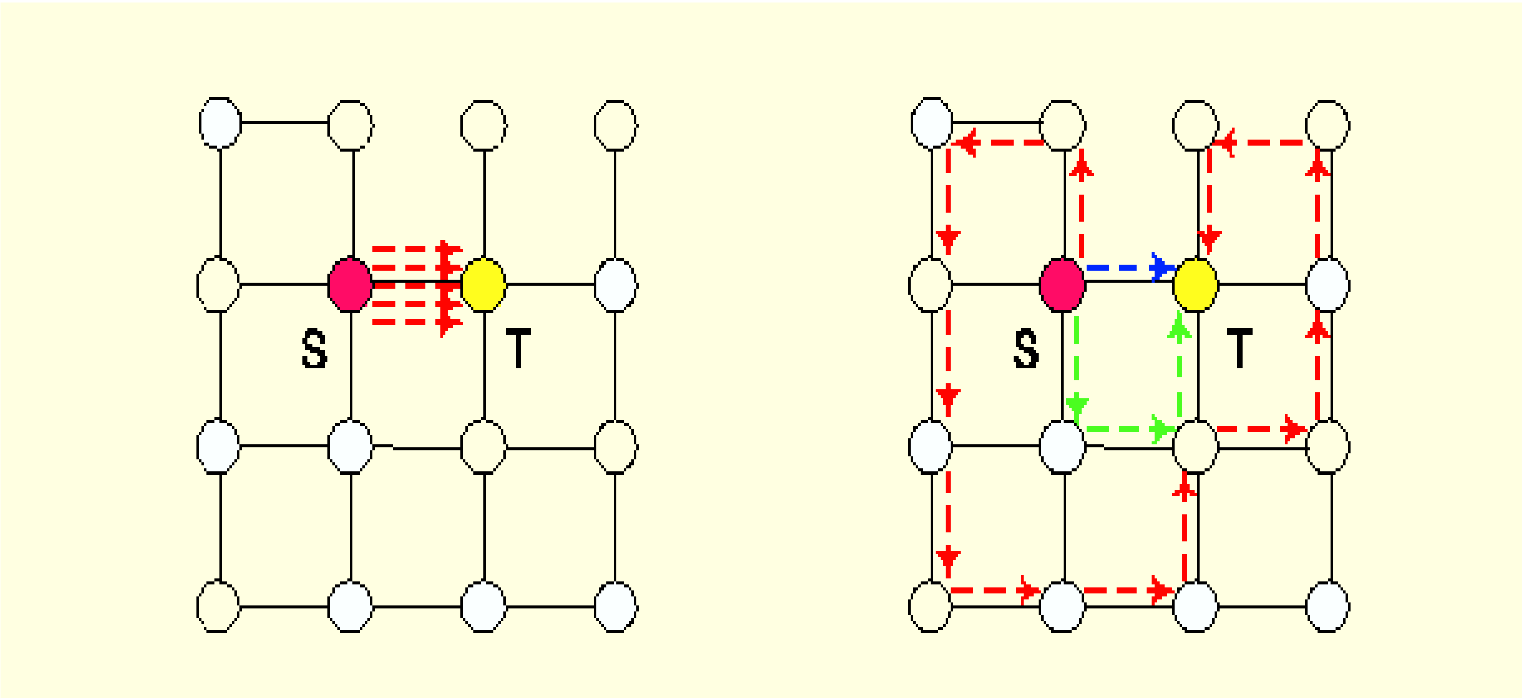
d-dim Mesh: $C = O(C^* \cdot d \cdot \log n)$
(**Maggs et al. [FOCS97]**)
(Also gave a lower bound $\Omega(C^* \cdot \frac{1}{d} \cdot \log n)$)

Arbitrary: $C = O(C^* \log^3 n)$
Räcke [FOCS02], (non-constructive)
Azar et al. [STOC03]
Harrelson et al. [SPAA03]
Bienkowski et al. [SPAA03]
(Polynomial-time, constructive)
Bansal et al. [SPAA03], (On-line version)

Oblivious algorithms with near-optimal C; **Unbounded** stretch

simple, **scalable**, **efficient**, near-optimal, **general**

Congestion–Stretch Trade Off



Good stretch, Bad C   Bad stretch, Good C

We want **Good Stretch**, **Good C**

Related Work – Opt. C,stretch

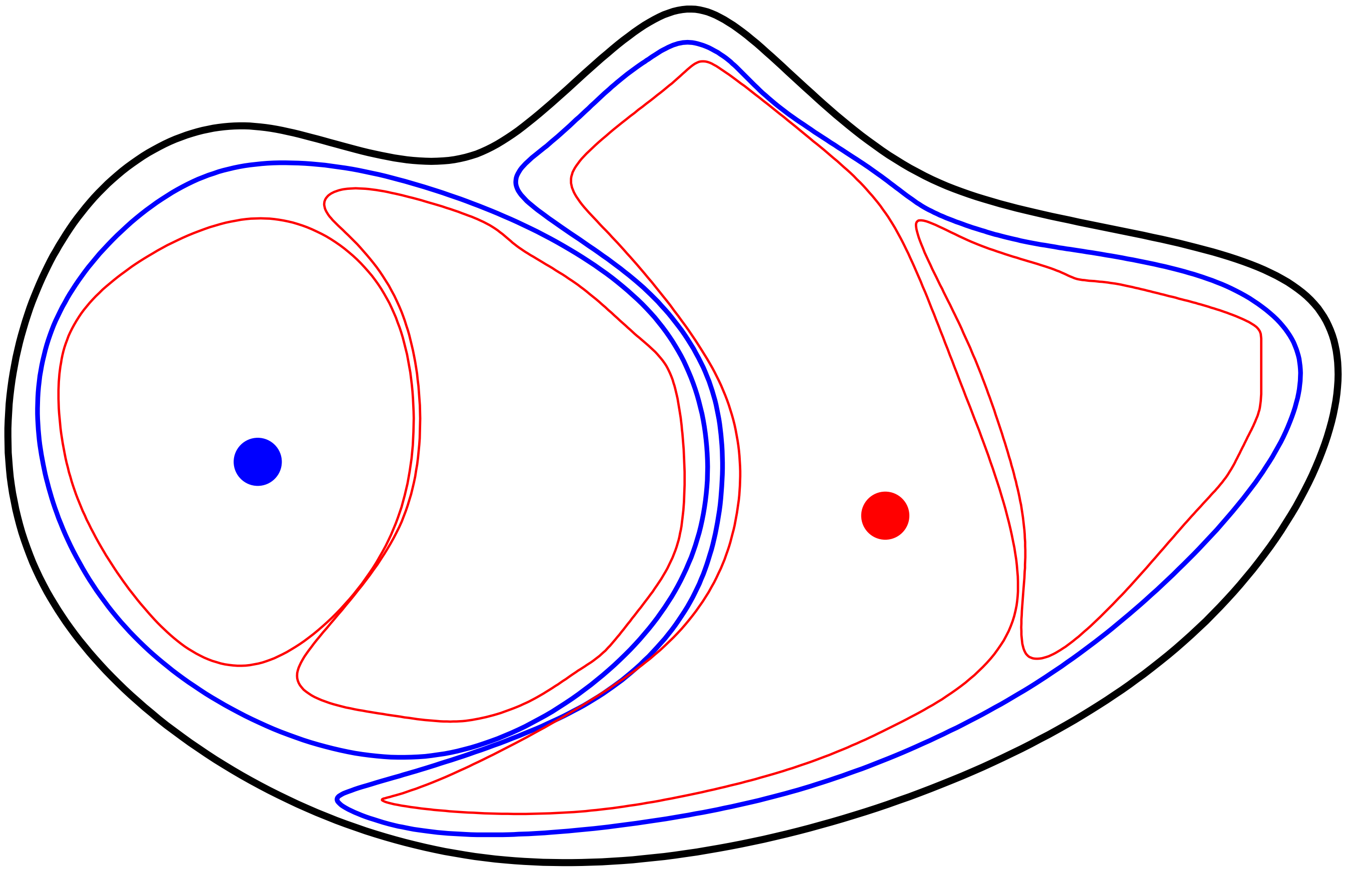
***d*-dim Mesh:** $C = O(C^* \cdot d^2 \cdot \log n)$; stretch $= O(d^2)$;
(**Busch et al. [IPDPS05]**)
(Also lower bound $\Omega(\log d(\pi))$ random bits per packet)
(**Scheidler (class notes)** indep. considered $d = 2$)

Arbitrary: Not Possible (constructive).

simple, **scalable, efficient, near-optimal,** general

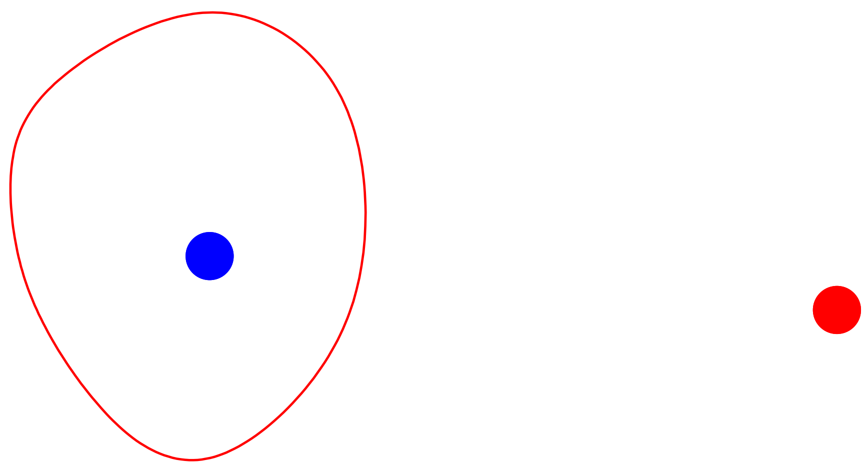
Hierarchical Decompositions

Existing algorithms use **hierarchical network decompositions**



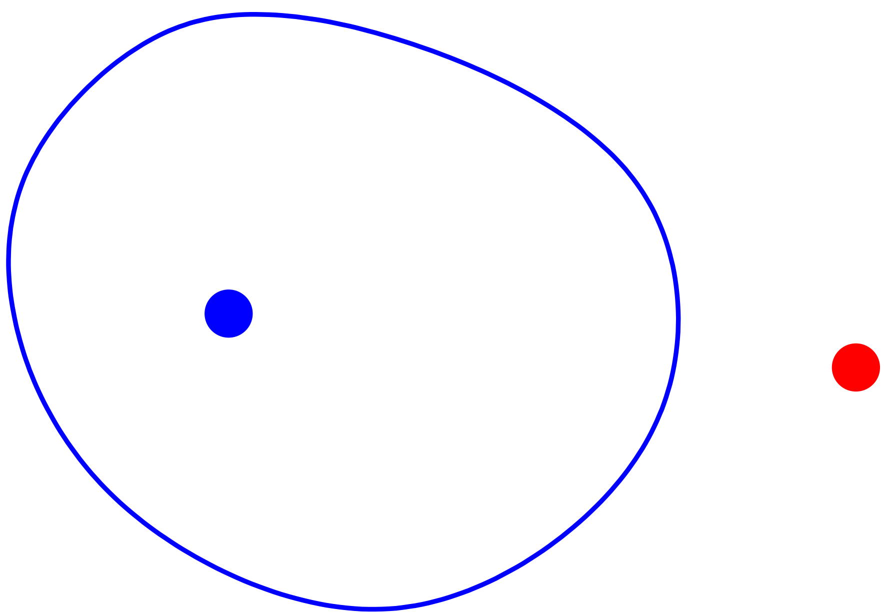
Hierarchical Decompositions

Existing algorithms use **hierarchical network decompositions**



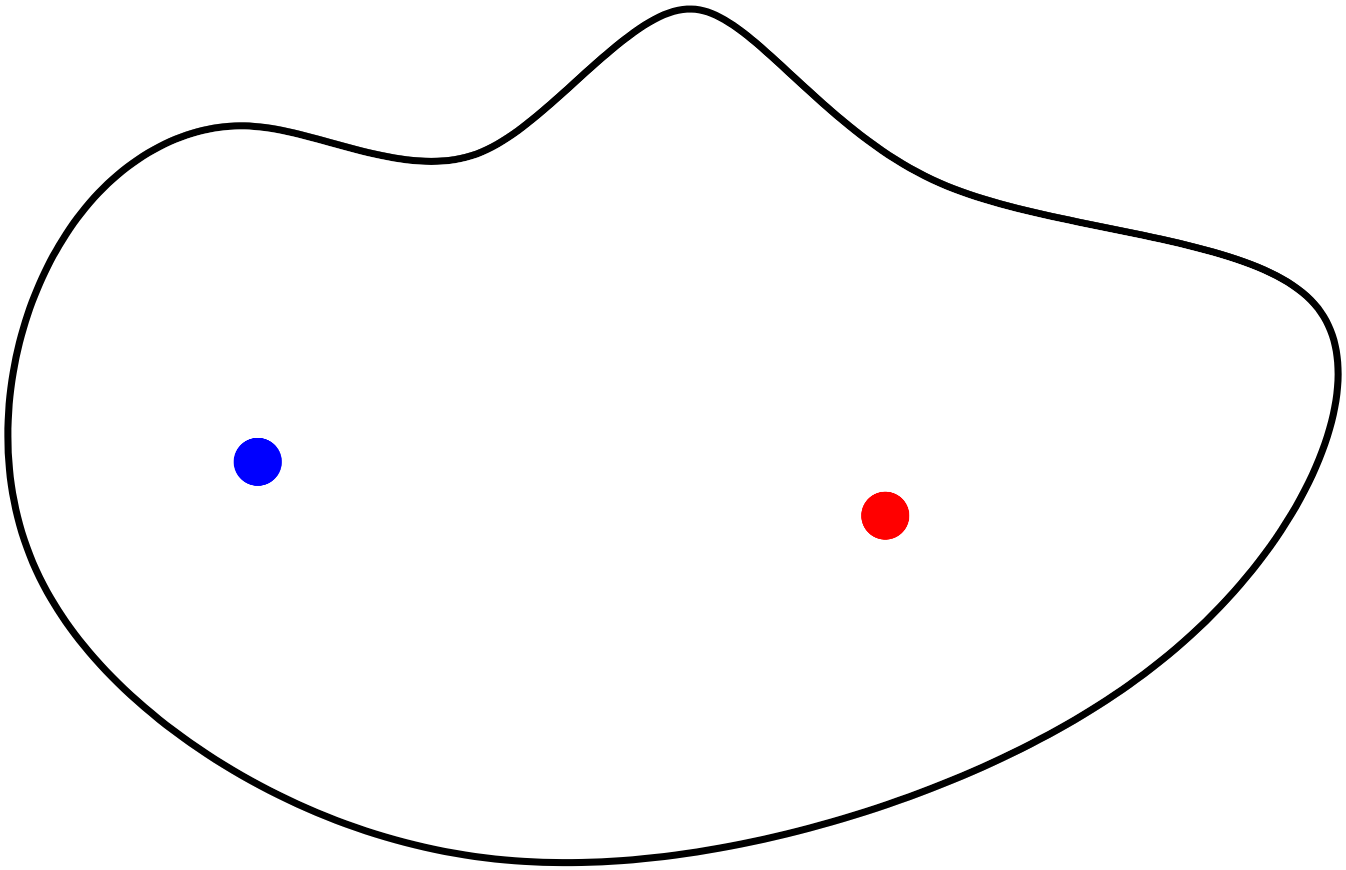
Hierarchical Decompositions

Existing algorithms use **hierarchical network decompositions**



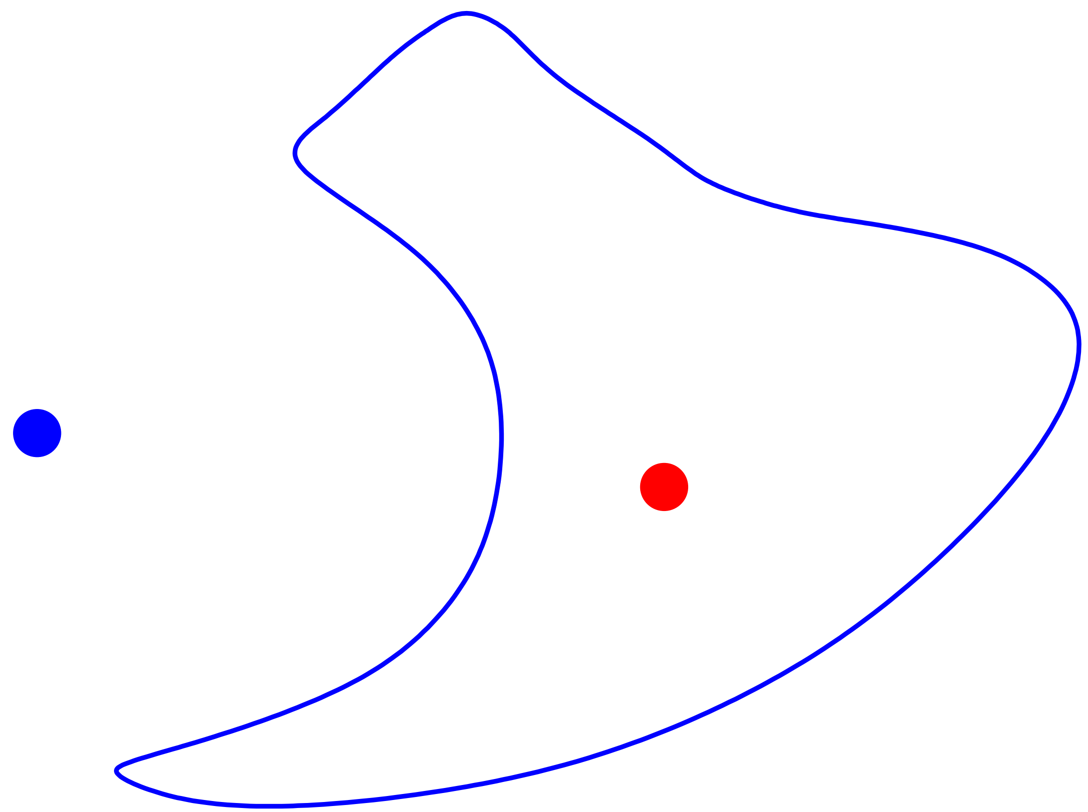
Hierarchical Decompositions

Existing algorithms use **hierarchical network decompositions**



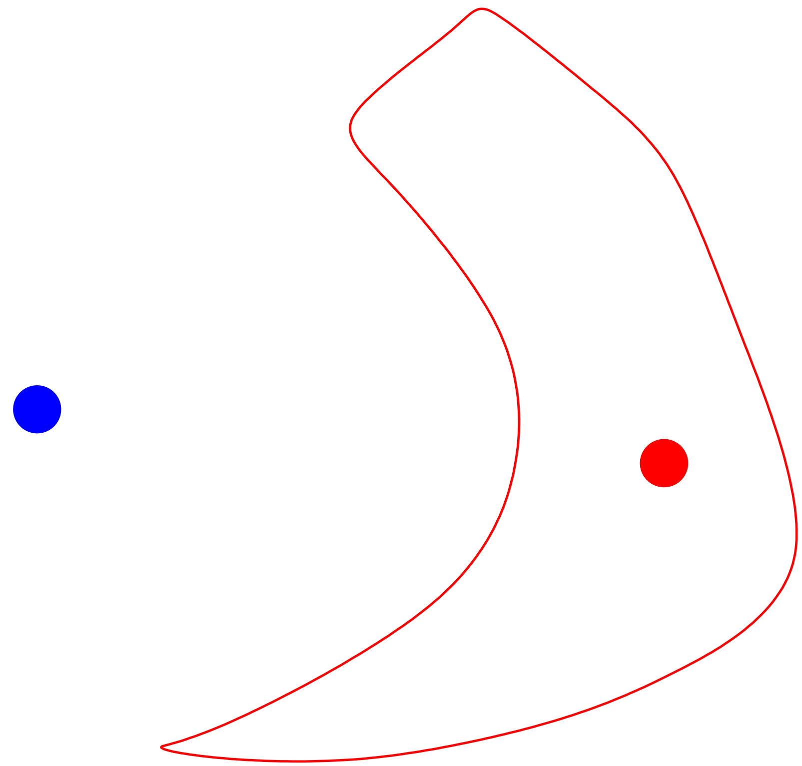
Hierarchical Decompositions

Existing algorithms use **hierarchical network decompositions**



Hierarchical Decompositions

Existing algorithms use **hierarchical network decompositions**



The Gap

simple, scalable, efficient, near-optimal, general

$\downarrow$

simple, scalable, efficient, near-optimal, general

$\downarrow$?

simple, scalable, efficient, near-optimal, general

(Not Possible)

Our Contribution

simple, scalable, efficient, near-optimal, general

↓

simple, scalable, efficient, near-optimal, general

↓

simple, scalable, efficient, near-optimal, general

Link Routing to Finite Metric Embedding

Our Contribution

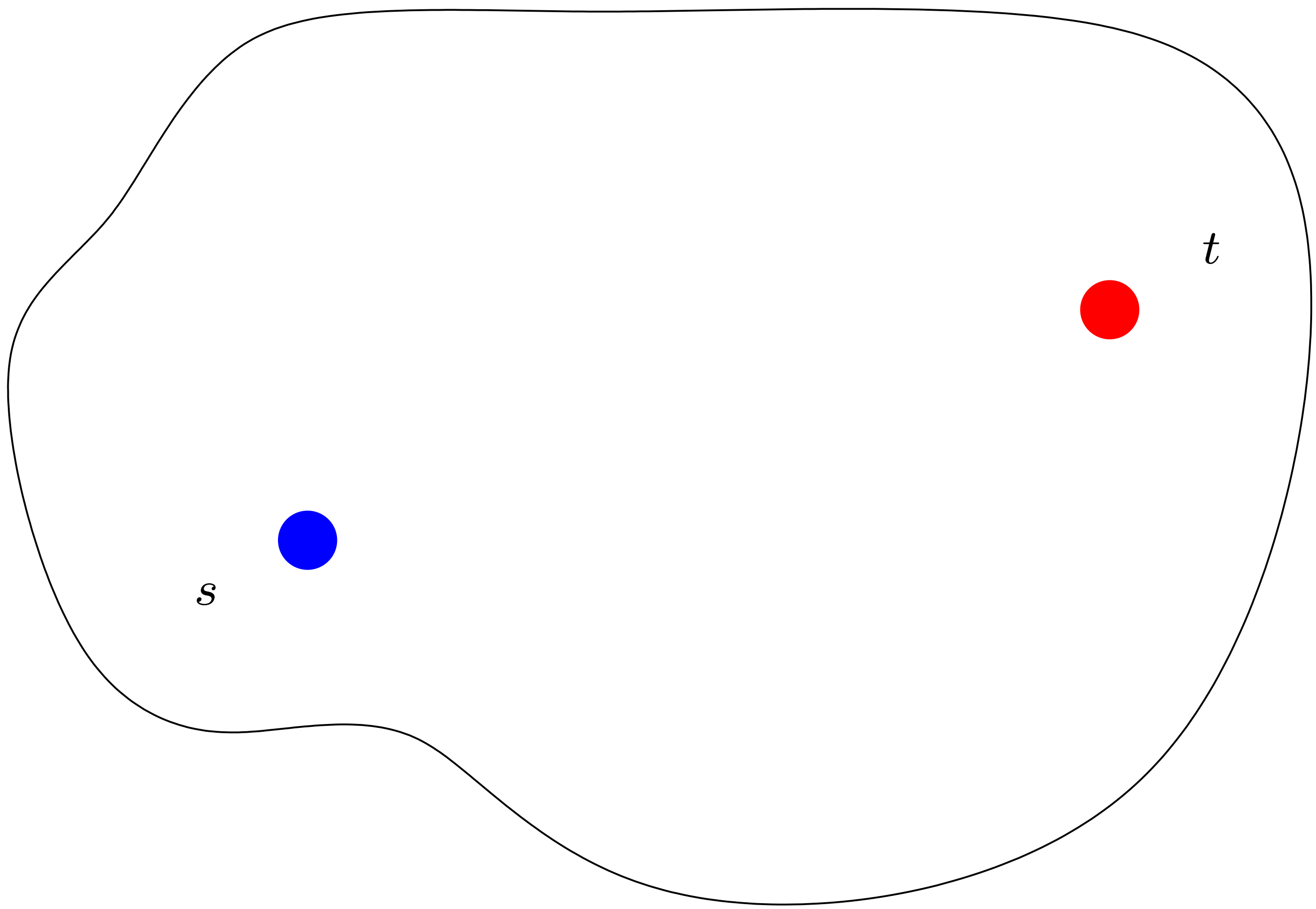
Simple: Single intermediate point algorithm, not based on hierarchical decomposition.

Near-Optimal: For networks that have low distortion embeddings, eg. Mesh, sensor networks.

General: Result holds for all networks.

General Idea: Diffusive Routing

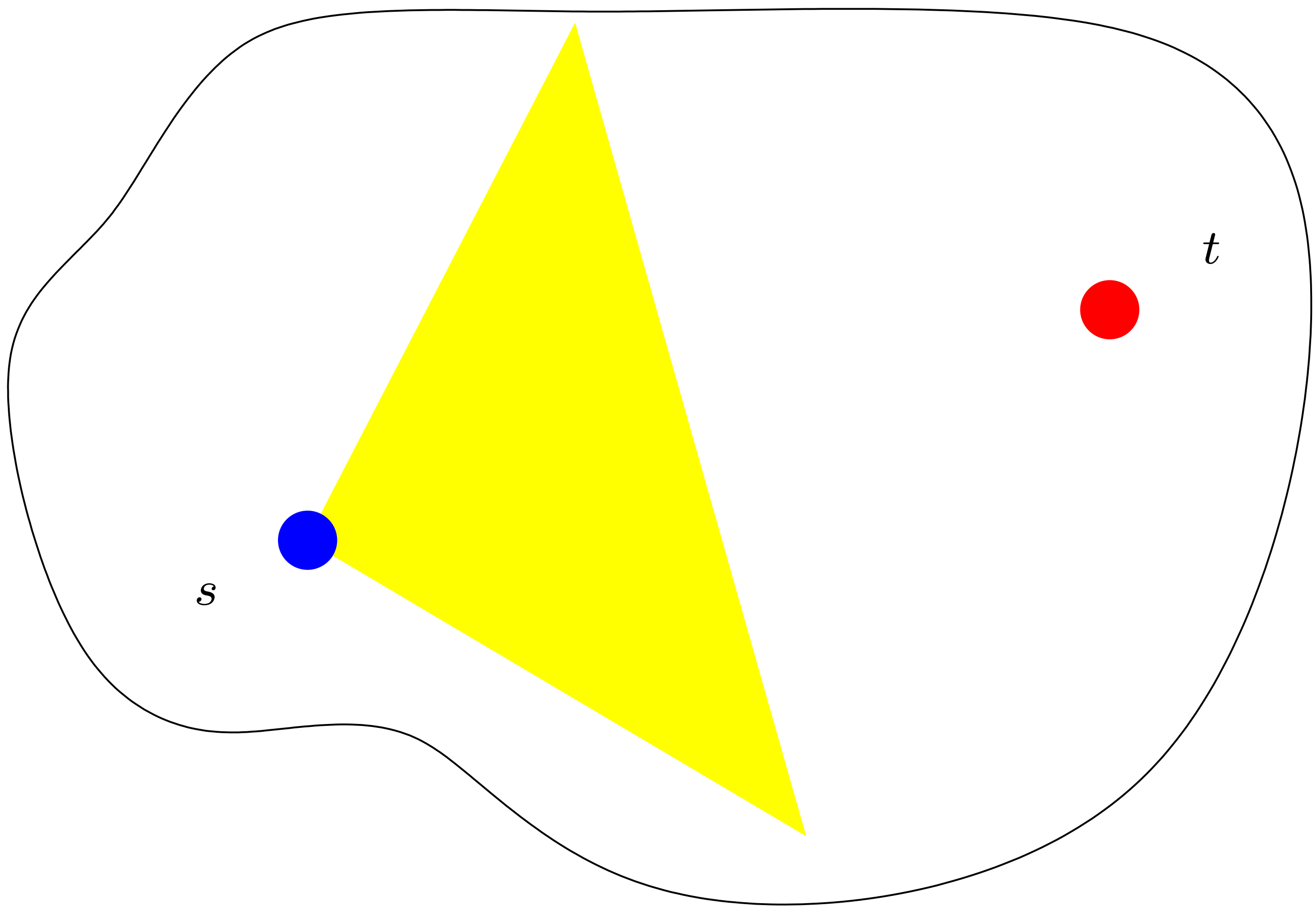
Imagine the network in space [Network Embedding]



source: s; destination: t.

General Idea: Diffusive Routing

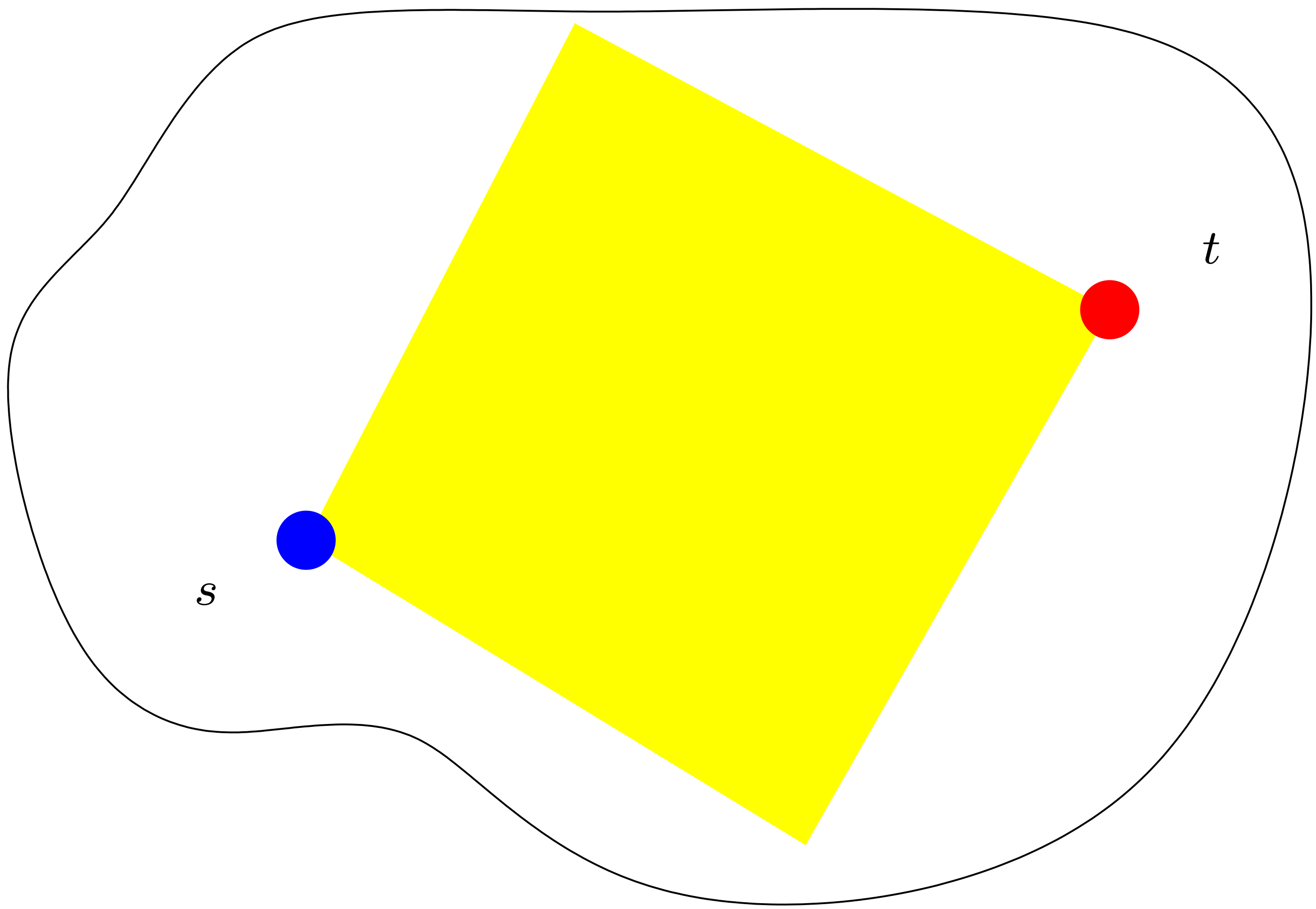
Imagine the network in space [Network Embedding]



Packet "diffuses" out from s – congestion spreads.

General Idea: Diffusive Routing

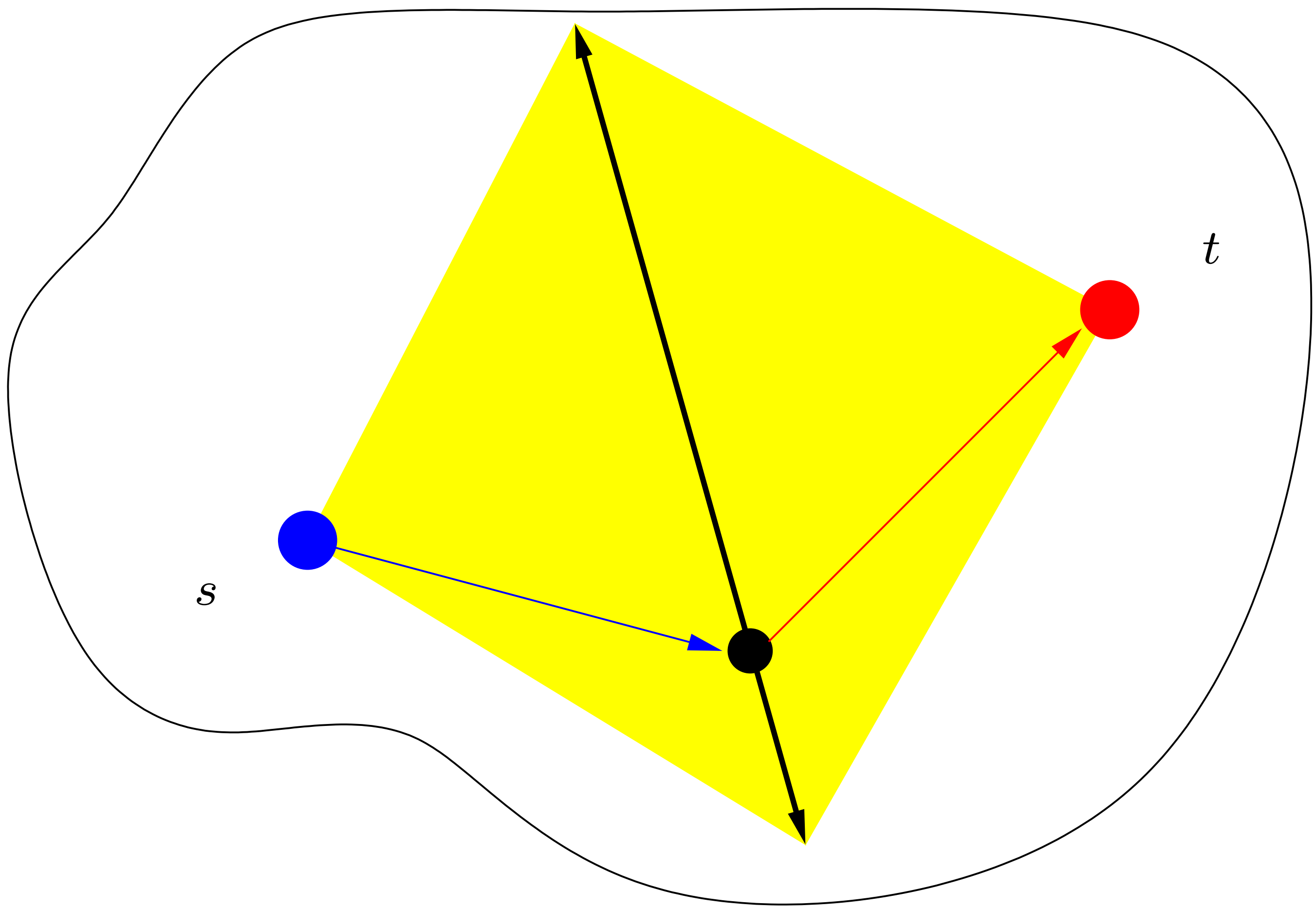
Imagine the network in space [Network Embedding]



Packet “focuses” back to t.

General Idea: Diffusive Routing

Imagine the network in space [Embedded Network]



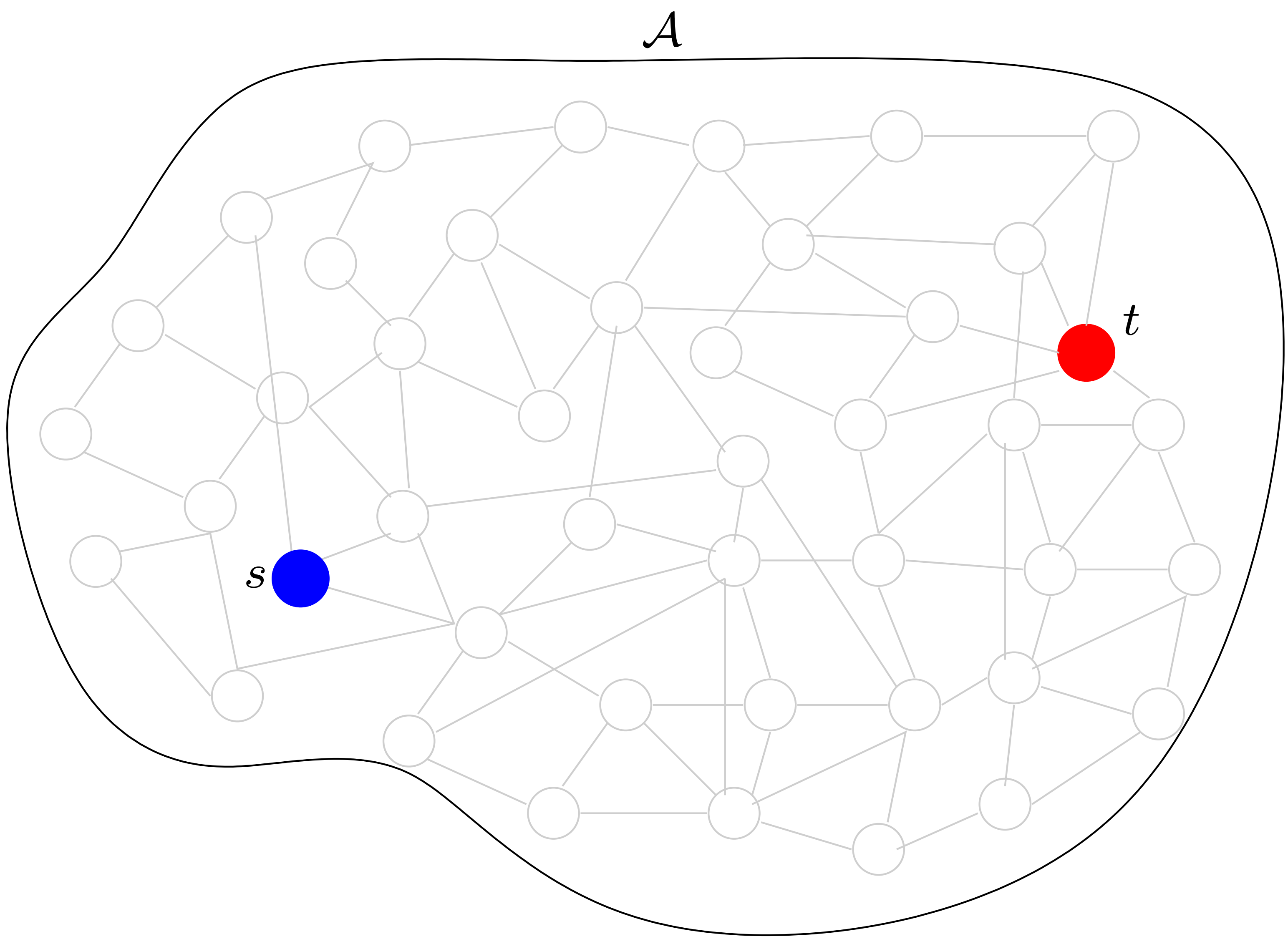
Diffusion by random choice of intermediate node.

Outline

- Oblivious Routing: Background and Our Contribution
- The Algorithm: Oblivious Routing with Single Intermediate Node
- Good Geometric (Metric) Embeddings; Examples
- Routing Result; Examples
- Discussion

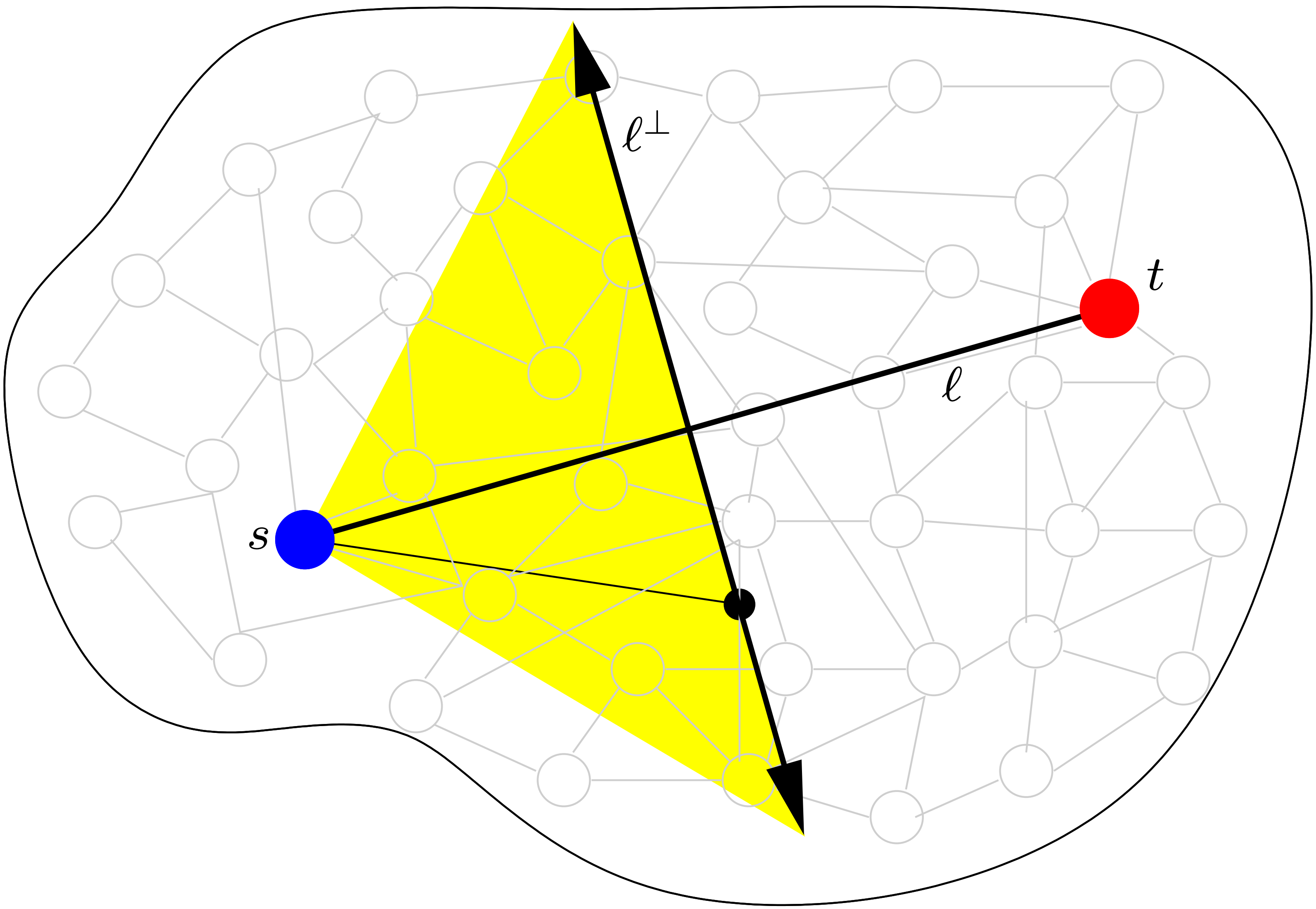
I: Embed the Network

Embedding function $f : v \in V \mapsto \mathbf{x} \in \mathcal{A} \subset \mathbb{R}^2$; $\{v_1, \ldots, v_n\} \mapsto \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$



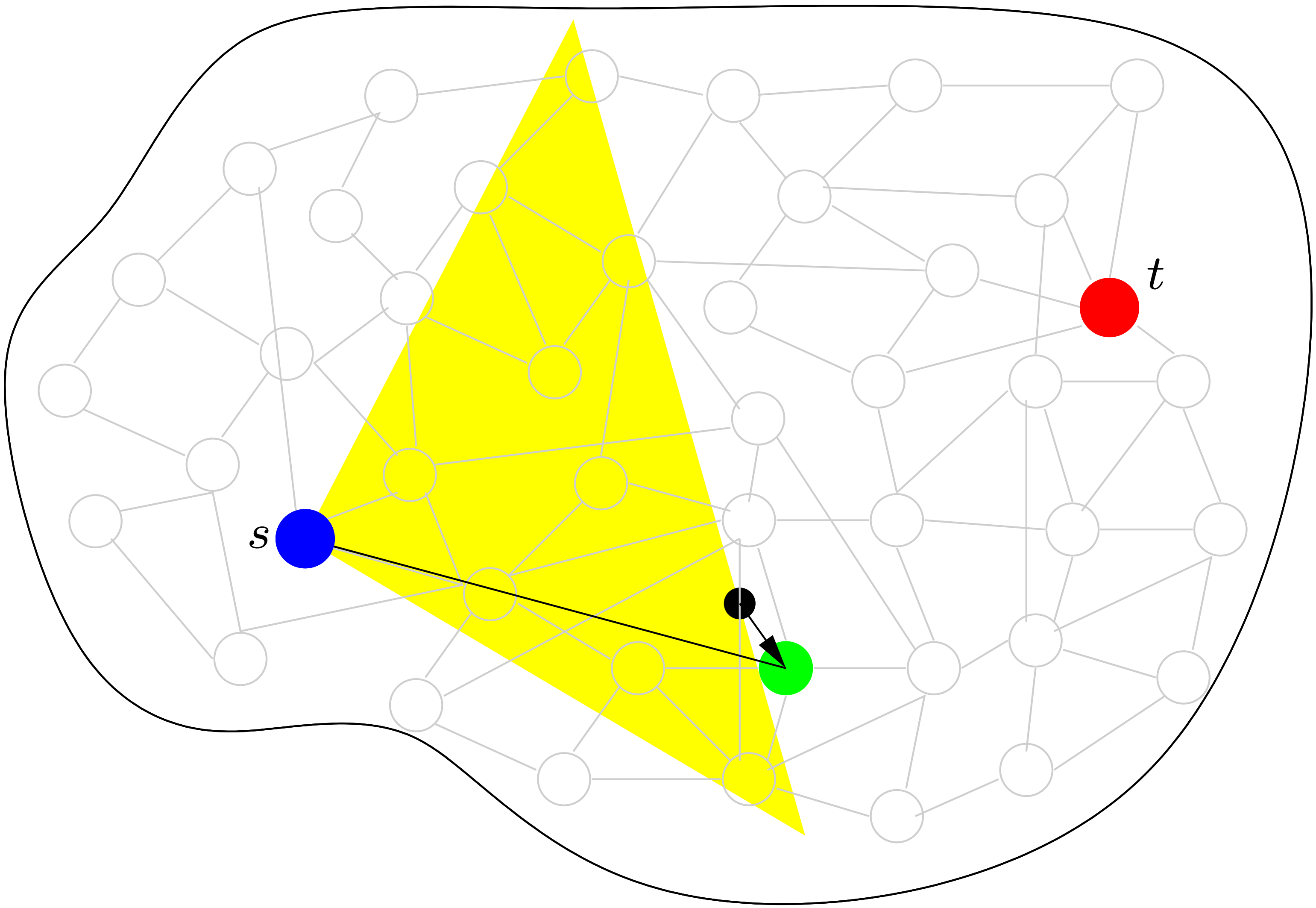
II: Random Intermediate Point

$$|\ell^{\perp}| = |\ell|$$



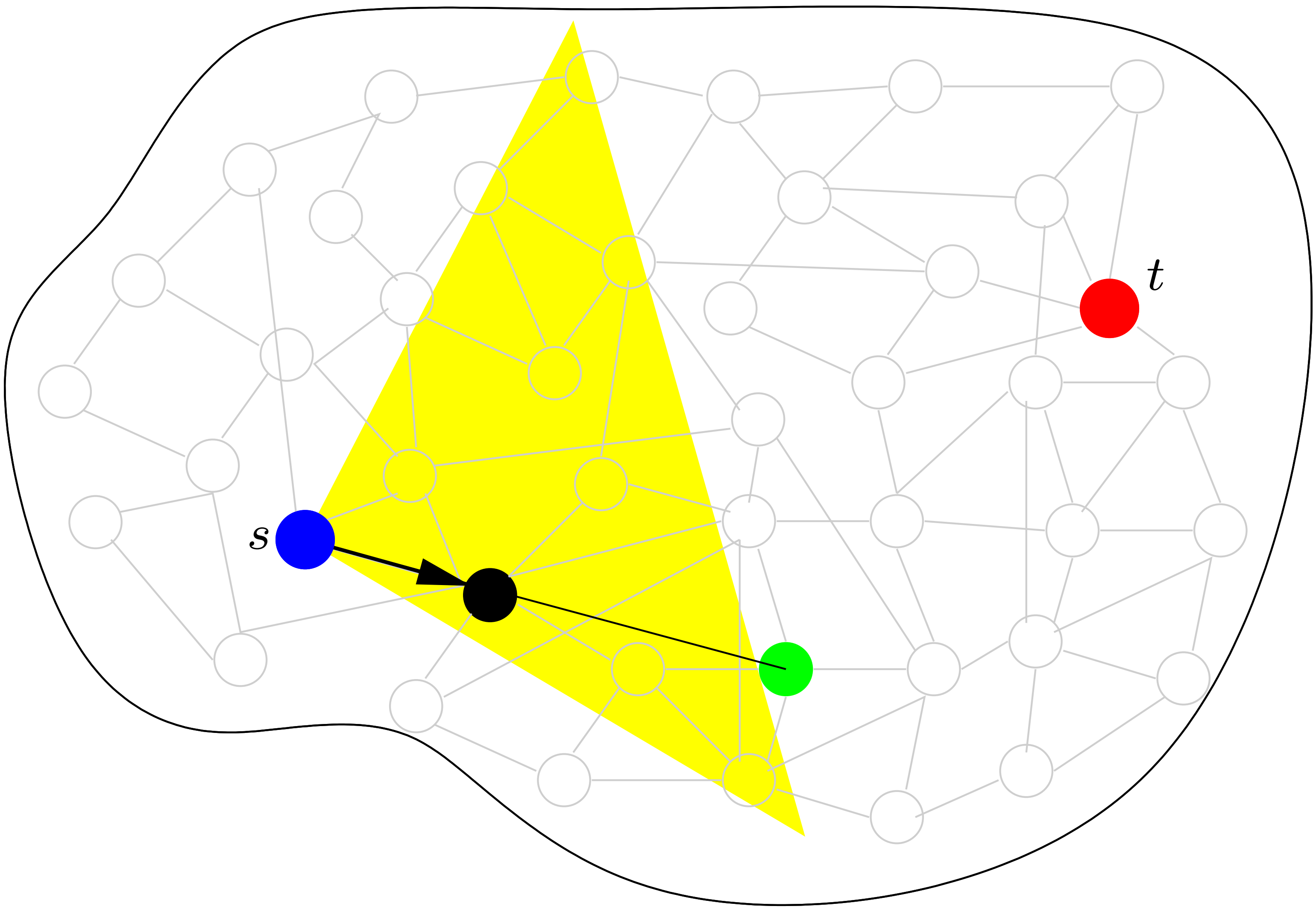
IIa: Random Intermediate Node

Choose a node close to the random intermediate point.



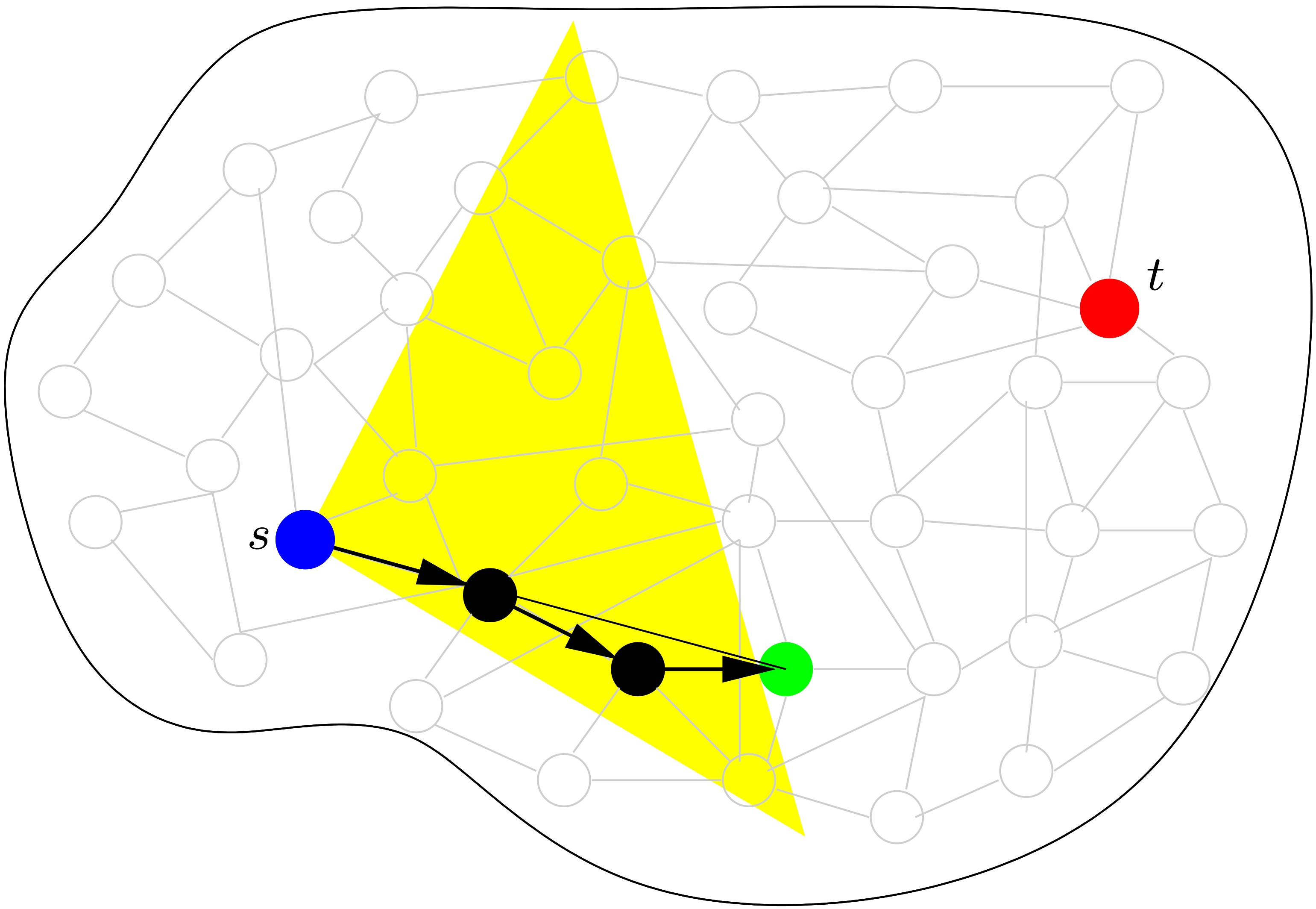
III: Follow the Geodesic

Choose a path as close as possible to the geodesic.



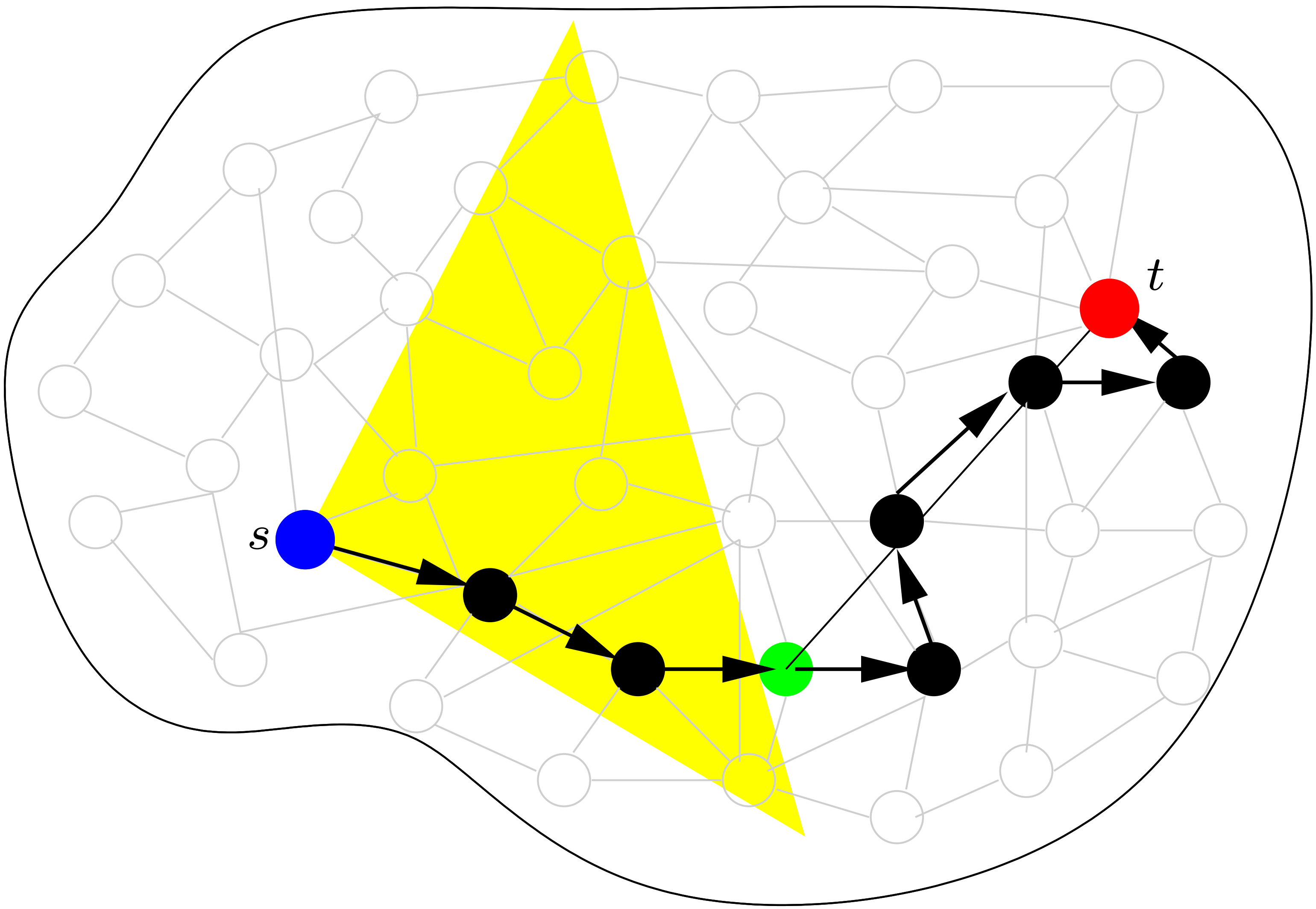
III: Follow the Geodesic

Choose a path as close as possible to the geodesic.



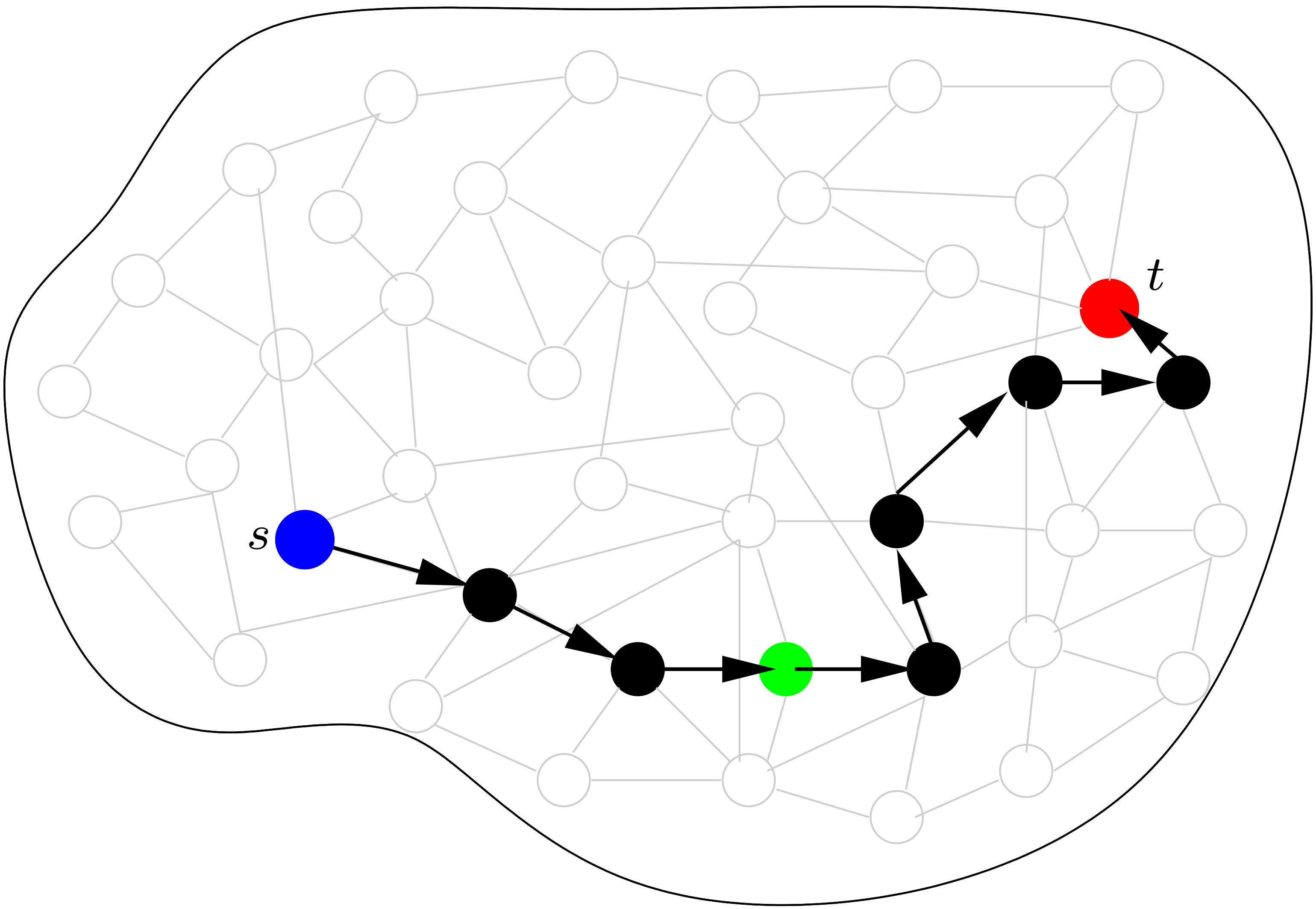
III: Follow the Geodesic

Choose a path as close as possible to the geodesic.



III: Follow the Geodesic

Choose a path as close as possible to the geodesic.



Outline

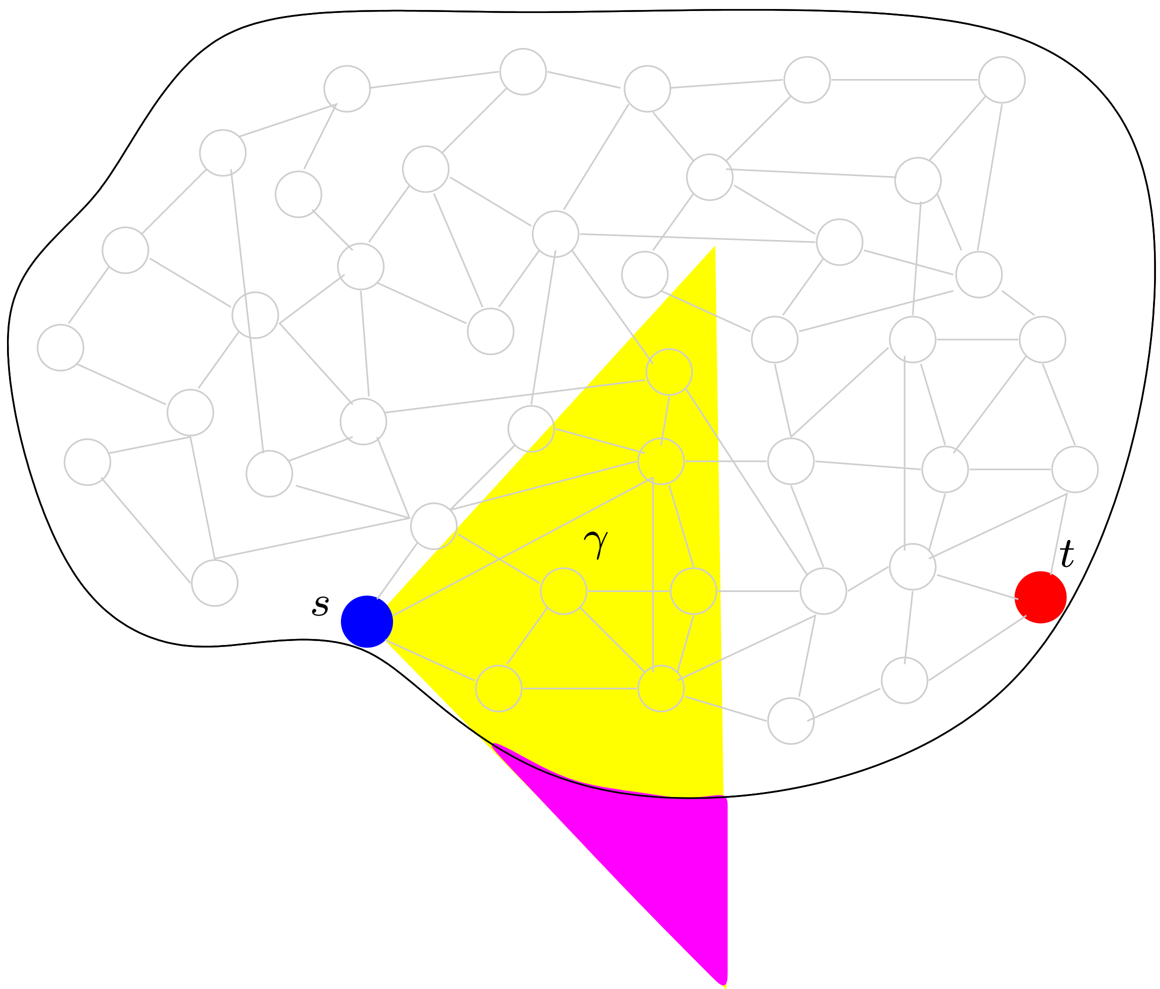
- Oblivious Routing: Background and Our Contribution
- The Algorithm: Oblivious Routing with Single Intermediate Node
- Good Geometric (Metric) Embeddings; Examples
- Routing Result; Examples
- Discussion

What Can Go Wrong?

Entire diffusive area is not within the network.

γ **(pseudo-convexity)**: fraction guaranteed to lie within network.

– Want γ to be large (note, $\gamma \leq \frac{1}{2}$).

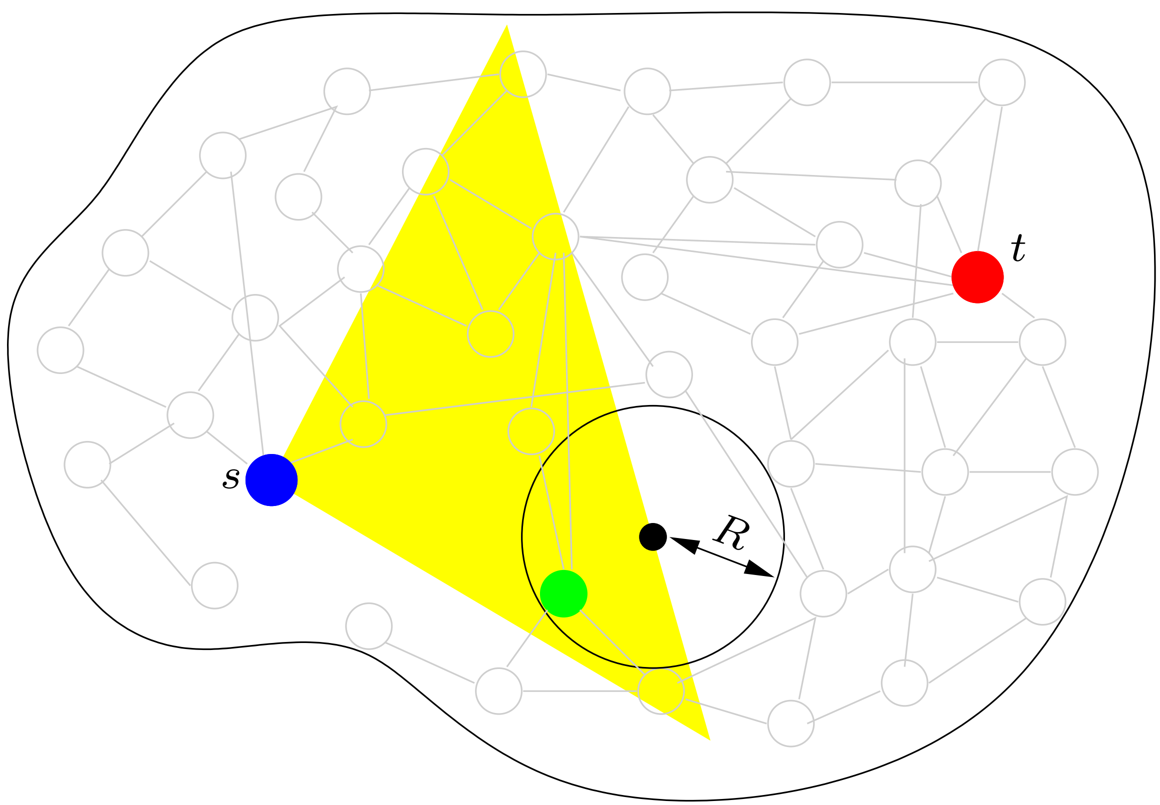


What Can Go Wrong?

There is no node close to the random intermediate node

R **(Coverage Radius):** Maximum distance to an intermediate node.

– Want R to be small.

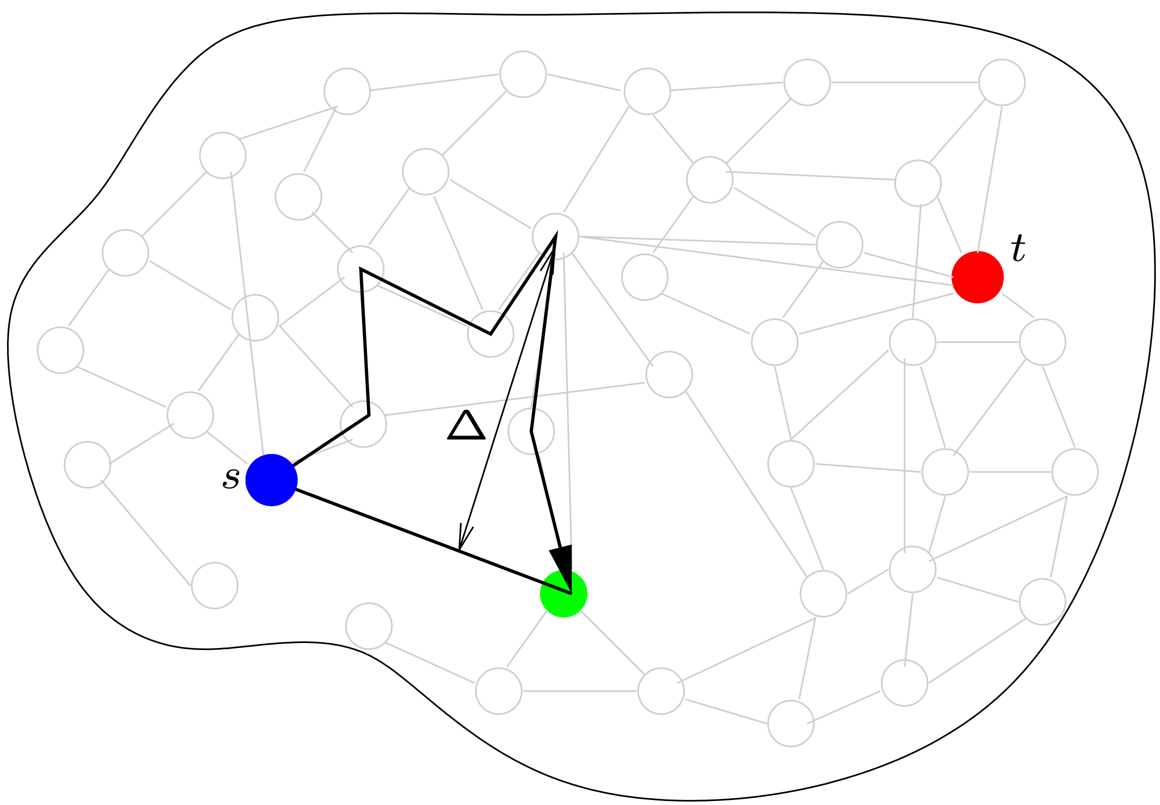


What Can Go Wrong?

No Geodesic following path

Δ (Deviation): Furtherest a geodesic path gets from the geodesic.

– Want Δ to be small.

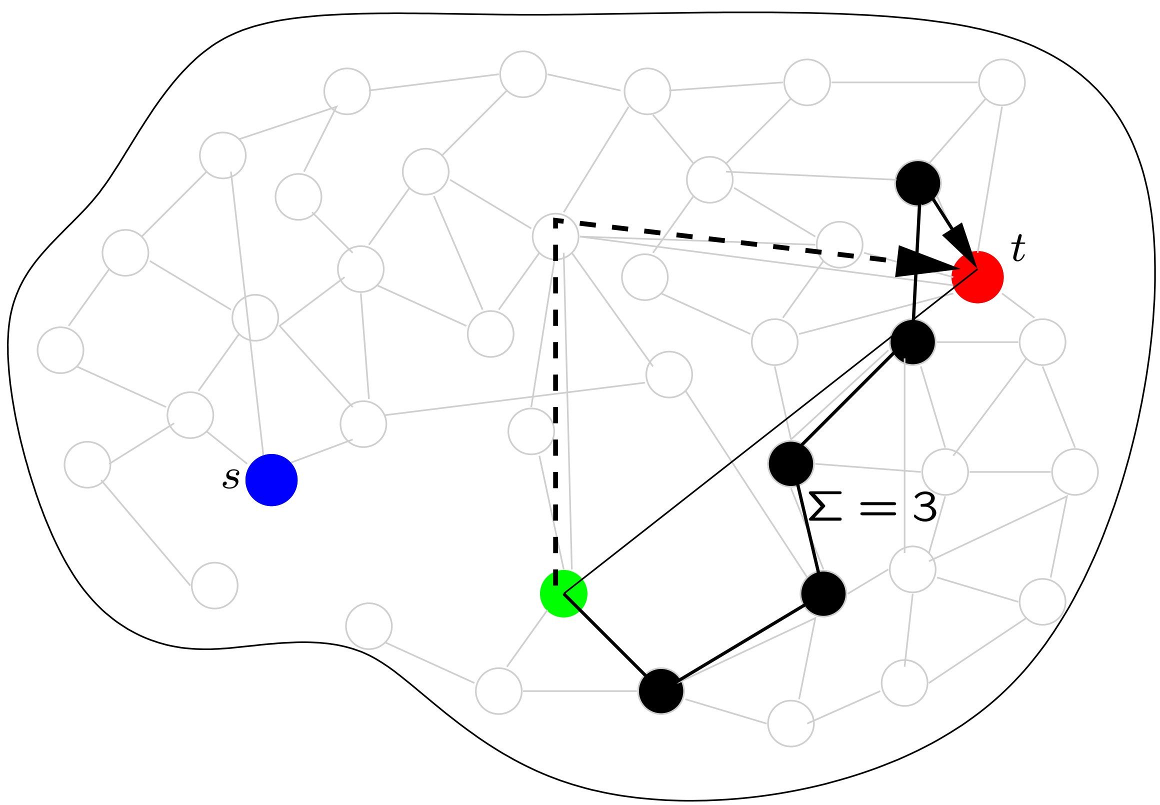


What Can Go Wrong?

Geodesic following paths have large stretch.

Σ **(Geodesic Stretch)**: Maximum stretch of a geodesic path.

– Want Σ to be small.



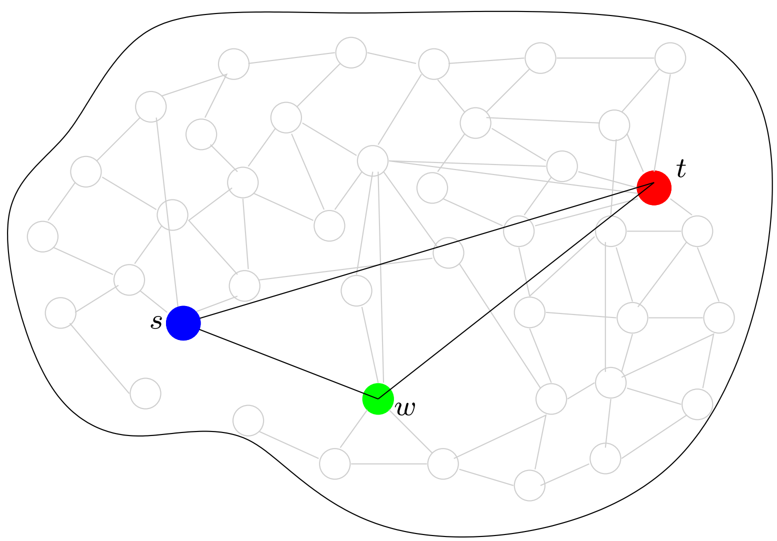
What Can Go Wrong?

Using an intermediate node is costly (large stretch).

$$\text{dist}_E(s,w) + \text{dist}_E(w,t) \leq \sqrt{2}\text{dist}_E(s,t).$$

Want $\text{dist}_G \approx \text{dist}_E$.

Distortion: $\alpha \leq \dfrac{\text{dist}_G(u,v)}{\text{dist}_E(u,v)} \leq \beta$ (w.l.o.g. $\alpha = 1$)



(dist_E=Euclidean distance, dist_G=Graph distance)

Graph Embedding Parameters

	Parameter	What is it?	Best If
γ	Pseudo-Convexity	Min. diffusive area in network	**large**
R	Coverage-Radius	Max. distance to intermediate node	**small**
Δ	Deviation	Max. stray of geodesic path	**small**
Σ	Geodesic Stretch	Max. stretch of geodesic paths	**small**
β	Distortion	How closely dist_G matches dist_E	**small**

Note: Embedding parameters are not independent.

eg. • γ and R are interdependent.

• Smaller deviation embedding may have a larger stretch.

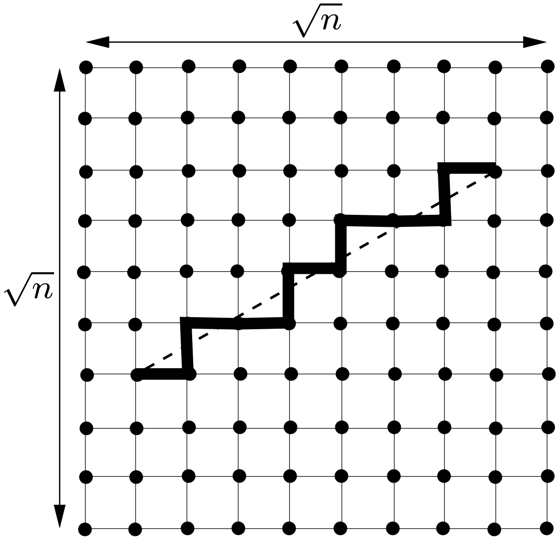
Examples

Certain networks have natural embeddings:

Mesh, sensor networks (disc graphs), ...

Examples

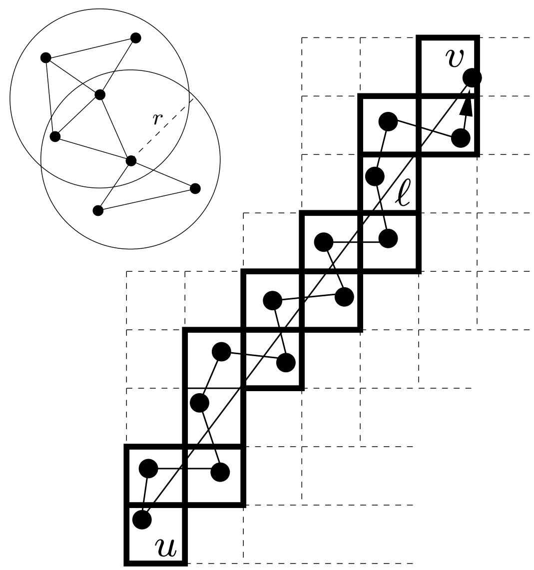
Mesh



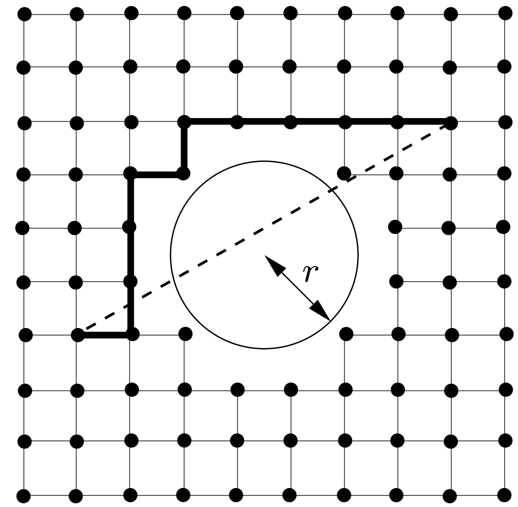
$$\gamma = \frac{1}{2}$$
$$R = \frac{1}{\sqrt{2}}$$
$$\Delta = \frac{1}{\sqrt{2}}$$
$$\Sigma = 1$$
$$\beta = \sqrt{2}$$

Geodesic following paths are shortest paths.

Sensor Network

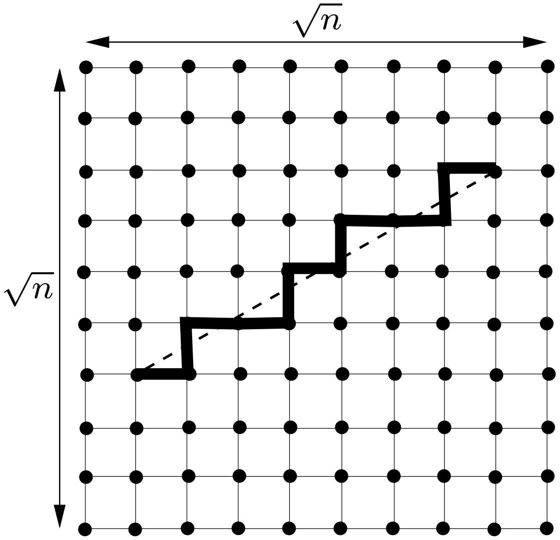


Mesh with Hole

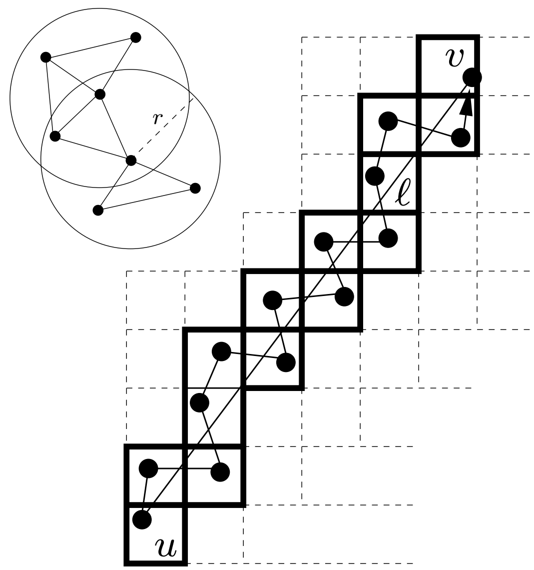


Examples

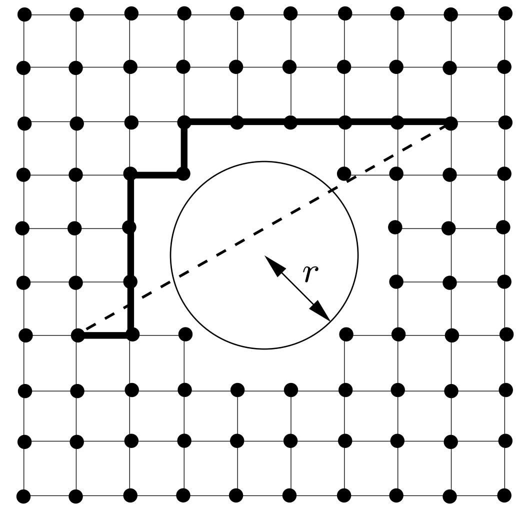
Mesh



Sensor Network



Mesh with Hole

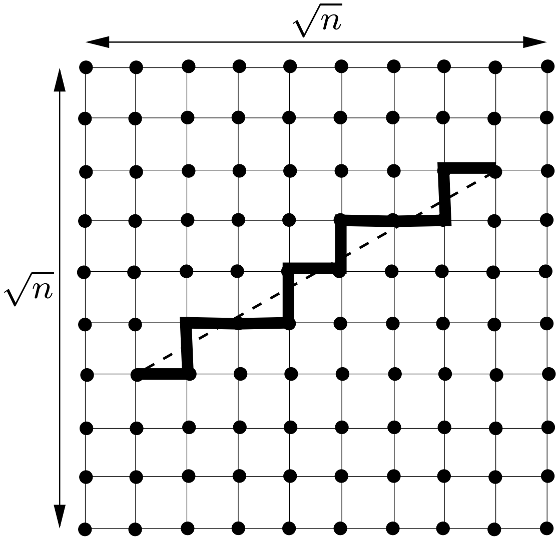


$$\gamma = \frac{1}{2}$$
$$R = \frac{1}{2}$$
$$\Delta = \frac{1}{2}$$
$$\Sigma = 2$$
$$\beta = \frac{2\sqrt{2}}{L}$$

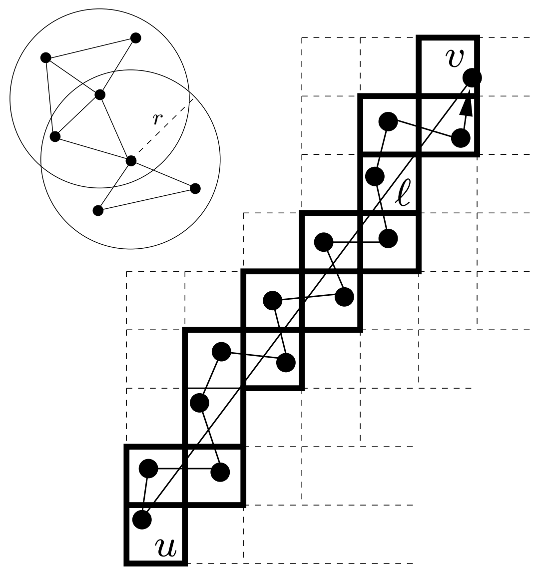
No two nodes are closer than L. Each unit square contains from 1 to $k = O(1/L^2)$ nodes. $r = 2\sqrt{2}$. (max. degree $\delta \leq 32k$.) Geodesic paths constructed from unit square path.

Examples

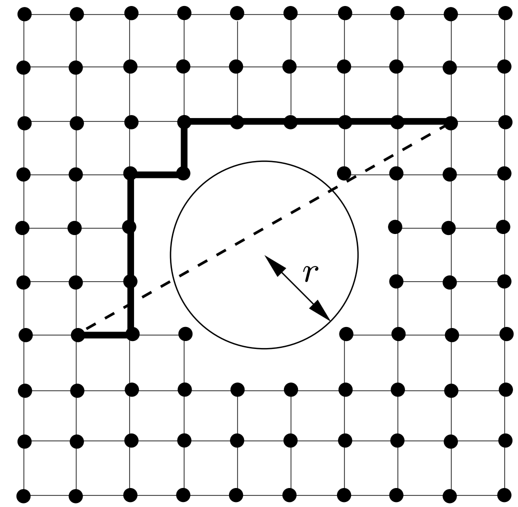
Mesh



Sensor Network



Mesh with Hole

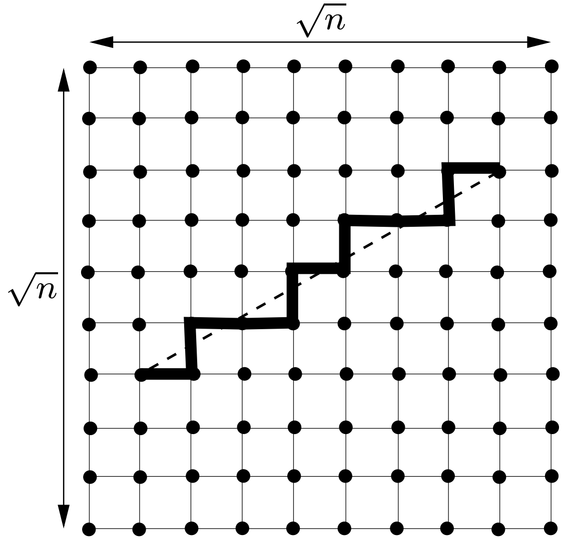


$$\begin{aligned} \gamma &= \frac{1}{2} \\ R &\to \frac{1}{\sqrt{2}} + r \\ \Delta &\to \frac{1}{\sqrt{2}} + r \\ \Sigma &= 1 \\ \beta &\leq 5 \end{aligned}$$

Geodesic following paths are shortest paths.

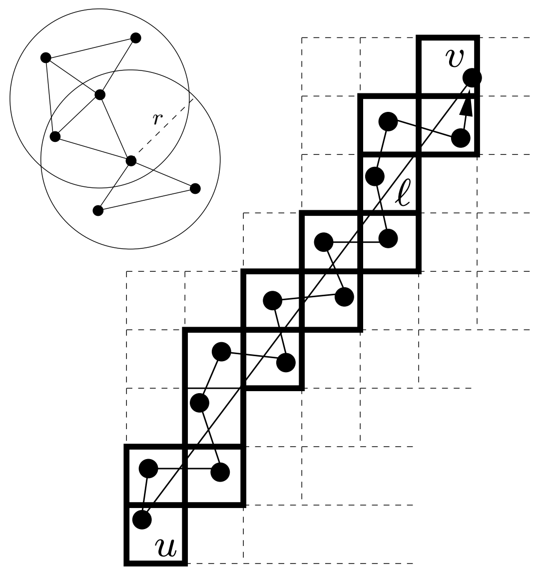
Examples

Mesh



$$\begin{aligned} \gamma &= O(1) \\ R &= O(1) \\ \Delta &= O(1) \\ \Sigma &= O(1) \\ \beta &= O(1) \end{aligned}$$

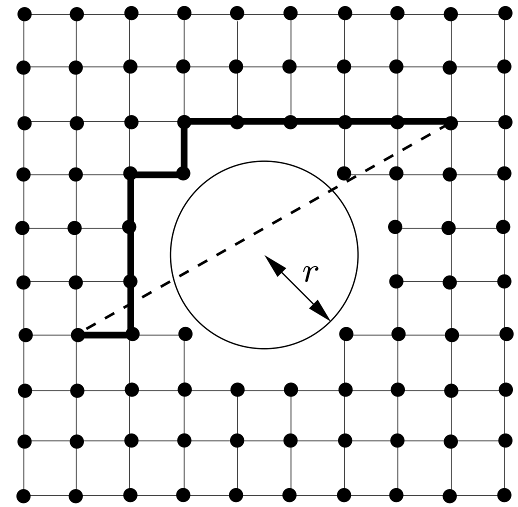
Sensor Network



$$\begin{aligned} \gamma &= O(1) \\ R &= O(1) \\ \Delta &= O(1) \\ \Sigma &= O(1) \\ \beta &= O(\tfrac{1}{L}) \end{aligned}$$

(L=min. node sep.)

Mesh with Hole



$$\begin{aligned} \gamma &= O(1) \\ R &= O(r) \\ \Delta &= O(r) \\ \Sigma &= O(1) \\ \beta &\leq O(1) \end{aligned}$$

(r=size of hole.)

Outline

- Oblivious Routing: Background and Our Contribution
- The Algorithm: Oblivious Routing with Single Intermediate Node
- Good Geometric (Metric) Embeddings; Examples
- Routing Result; Examples
- Discussion

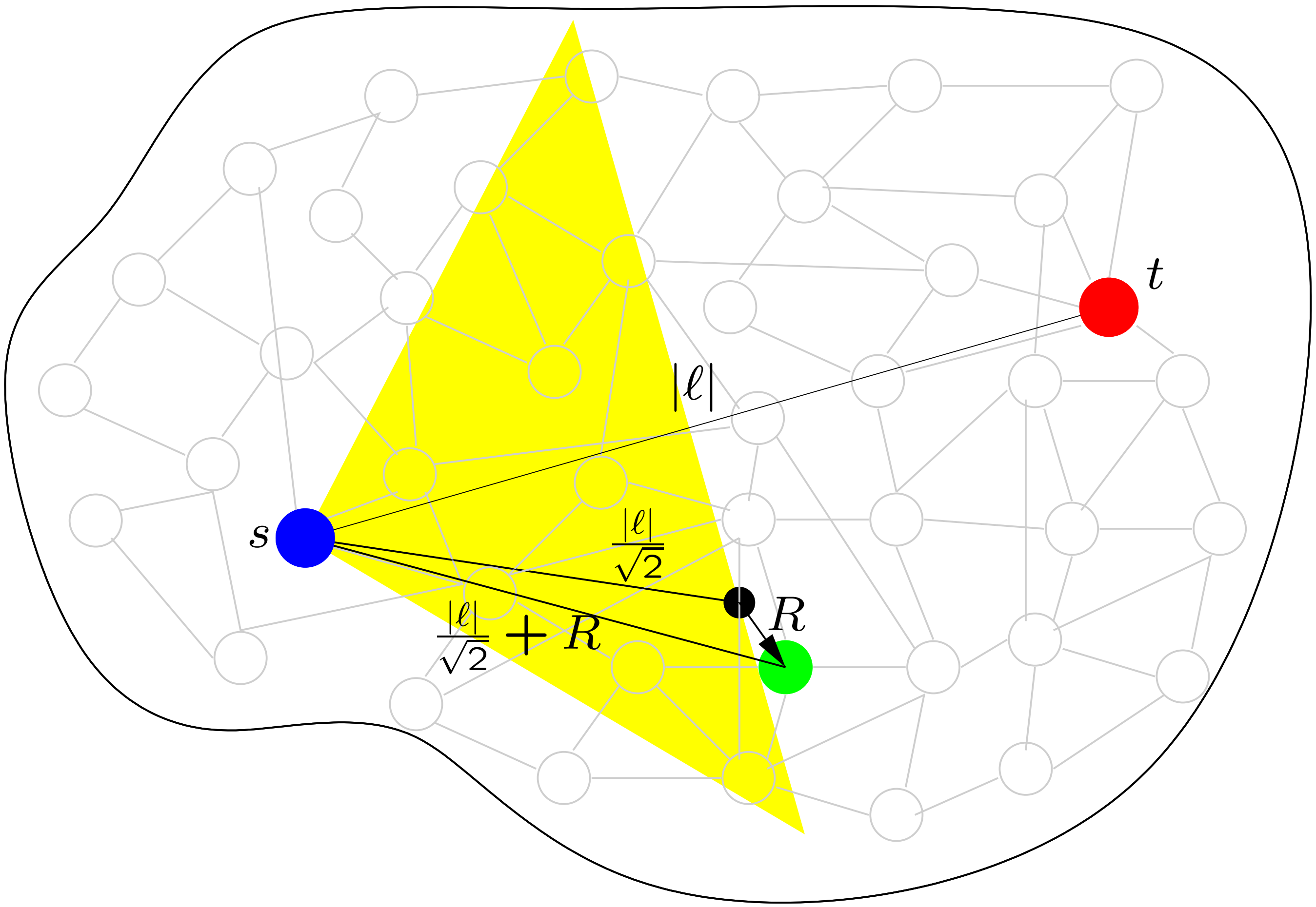
Stretch

Theorem.

$$\text{stretch} = O(\beta \cdot R \cdot \Sigma)$$

The stretch depends on :
- Quality of the embedding: β;
- Coverage density: R;
- Geodesic stretch: Σ.

Proof (Sketch)



- distance stretch is $O(1{+}R)$.
- β links distances to graph distances.
- Σ is stretch introduced by geodesic paths.

$\Big\} \rightarrow \beta \cdot R \cdot \Sigma$

Congestion

Theorem.

$$C \leq f(\gamma, R, \Delta, \beta; n) \cdot C^*$$

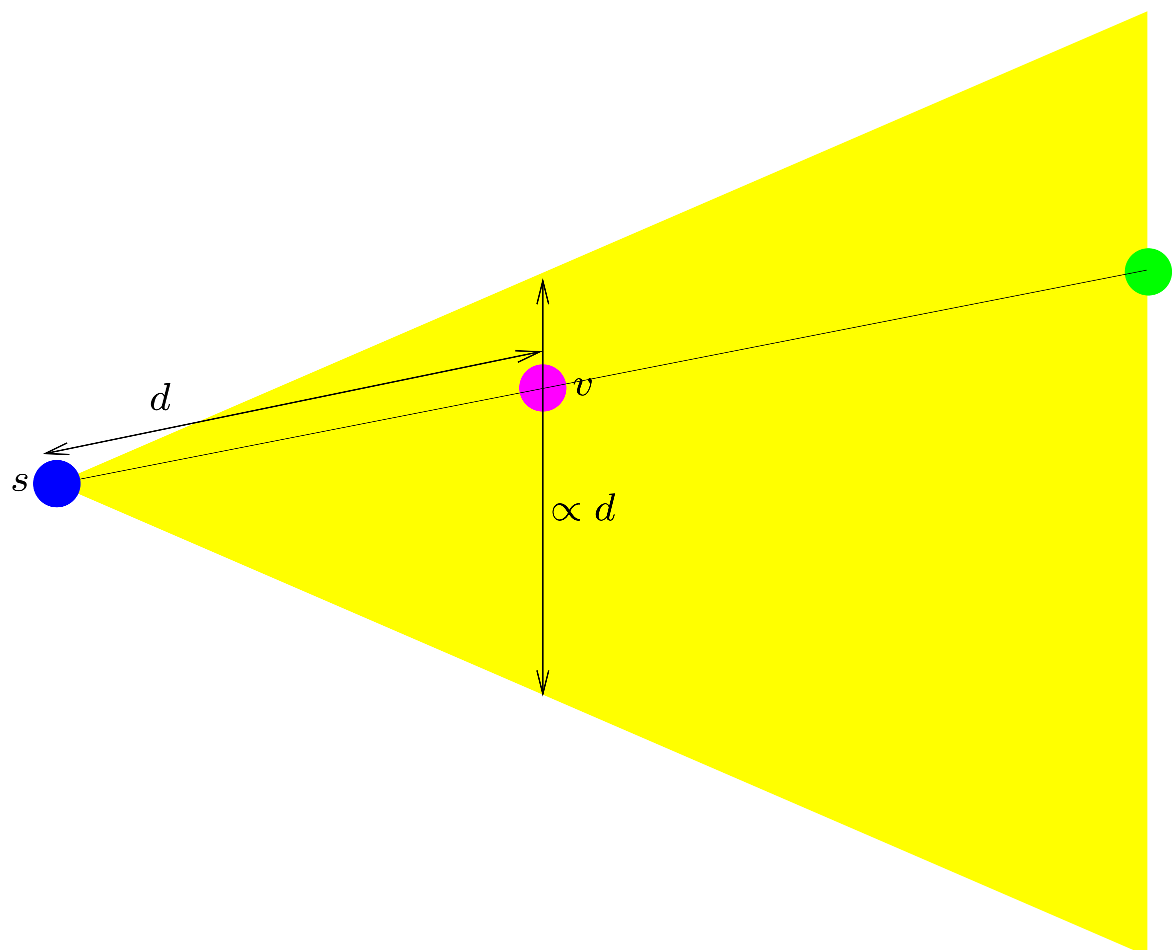
$$f = O\left(\frac{\beta^2(R+\Delta)}{\gamma} \cdot \left((\beta+\Delta)^2 + \log(n+R)\right)\right)$$

$$(f = O(\log n))$$

The Congestion depends on :
– Optimal Congestion: C^*;
– Extent of diffusion: γ;
– Quality of the embedding: β;
– Coverage density: R;
– Deodesic deviation: Δ.

Proof (Sketch)

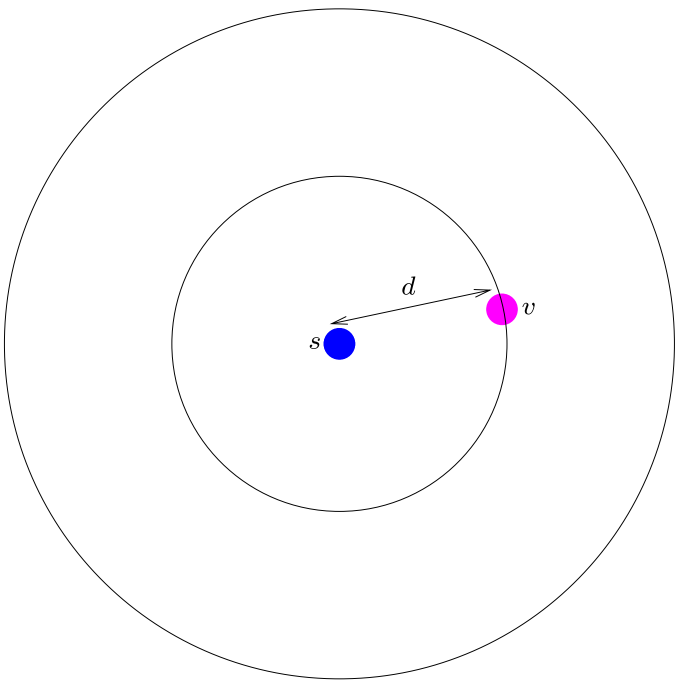
Lemma 1. Probability (P) source at distance d uses node v. $P \sim \frac{1}{d}$.
(so $E[C(v)] \sim \sum_d \frac{N_d}{d}$, where N_d = Number of sources distance d from v)



Proof (Sketch)

Lemma 2. $C^* \sim \frac{N_d}{d}$.

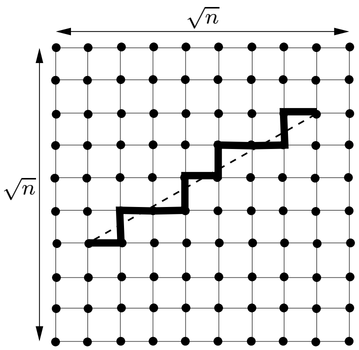
(so $E[C(v)] \sim \sum_d \frac{N_d}{d} \sim \sum_d C^* \sim C^* \log n$)



– destination is $\sim 2d$ away.
– use at least $\sim d$ nodes in $2d$-disc.
– total node usage $\sim N_d \cdot d$.
– number of nodes in disc $\sim d^2$.
– pigeonhole: $\exists$ node used $\sim \frac{N_d}{d}$ times.

Examples

Mesh

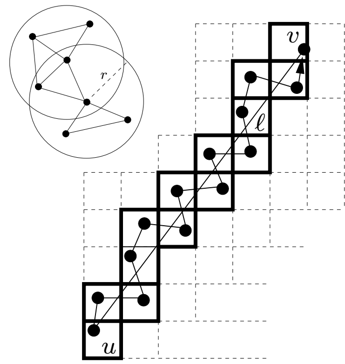


$$\begin{aligned} \gamma &= O(1) \\ R &= O(1) \\ \Delta &= O(1) \\ \Sigma &= O(1) \\ \beta &= O(1) \end{aligned}$$

stretch: $\boldsymbol{O(1)}$

$\boldsymbol{C : O(C^* \log n)}$

Sensor Network



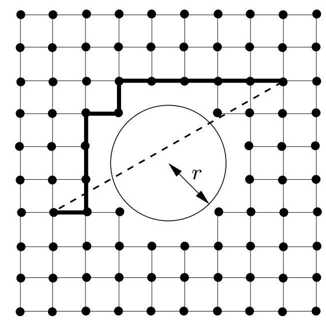
$$\begin{aligned} \gamma &= O(1) \\ R &= O(1) \\ \Delta &= O(1) \\ \Sigma &= O(1) \\ \beta &= O(\tfrac{1}{L}) \end{aligned}$$

stretch: $\boldsymbol{O(\frac{1}{L})}$

$\boldsymbol{C : O(C^* \frac{\log n}{L^2})}$

(L=min. node sep.)

Mesh with Hole



$$\begin{aligned} \gamma &= O(1) \\ R &= O(r) \\ \Delta &= O(r) \\ \Sigma &= O(1) \\ \beta &\leq O(1) \end{aligned}$$

stretch: $\boldsymbol{O(r)}$

$\boldsymbol{C : O(C^* \cdot r \log n)}$

(r=size of hole.)

Wrap Up

– Embedding Parameters: $\gamma, R, \Delta, \Sigma, \beta$.

– Good embedings: Good embedding parameters.

– Diffusive Routing: $\text{stretch} = O(1)$; $C = O(C^* \log n)$.

simple, scalable, efficient, near-optimal, general

Ongoing: Can we remove the dependence on γ.