Sparse Semi-Oblivious Routing:

Few Random Paths Suffice

Goran Zuzic[1, 2] ⓡ Bernhard Haeupler[1, 3] ⓡ Antti Roeyskoe[1]

1: ETH Zürich
2: Google Research
3: Carnegie Mellon University

PODC 2023

Packet Routing

Packet Routing

Given a graph and a set of packets (with fixed source and destination vertices), find a **low-congestion low-dilation** set of paths routing the packets

Packet Routing

Packet Routing

Given a graph and a set of packets (with fixed source and destination vertices), find a **low-congestion low-dilation** set of paths routing the packets

Congestion: maximum number of times any edge is used

Packet Routing

Packet Routing

Given a graph and a set of packets (with fixed source and destination vertices), find a **low-congestion low-dilation** set of paths routing the packets

Congestion: maximum number of times any edge is used
Dilation: maximum length of any path in P

Packet Routing

Packet Routing

Given a graph and a set of packets (with fixed source and destination vertices), find a **low-congestion low-dilation** set of paths routing the packets

Congestion: maximum number of times any edge is used
Dilation: maximum length of any path in P

Packet scheduling possible in $\mathcal{O}(\text{congestion} + \text{dilation})$ time [Lei+94]

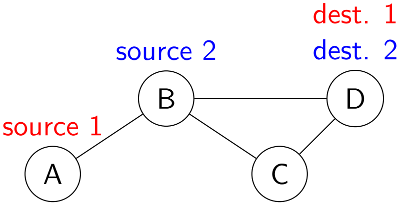
Packet Routing

Packet Routing

Given a graph and a set of packets (with fixed source and destination vertices), find a **low-congestion low-dilation** set of paths routing the packets

Congestion: maximum number of times any edge is used
Dilation: maximum length of any path in P

Packet scheduling possible in $\mathcal{O}(\text{congestion} + \text{dilation})$ time [Lei+94]



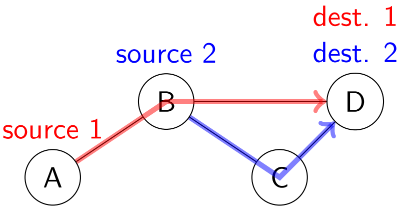
Packet Routing

Packet Routing

Given a graph and a set of packets (with fixed source and destination vertices), find a **low-congestion low-dilation** set of paths routing the packets

Congestion: maximum number of times any edge is used
Dilation: maximum length of any path in P

Packet scheduling possible in $\mathcal{O}(\text{congestion} + \text{dilation})$ time [Lei+94]



Oblivious Routing

Oblivious Routing

The path for any packet must be selected **obliviously**: without knowledge of other packets

Oblivious Routing

Oblivious Routing

The path for any packet must be selected **obliviously**: without knowledge of other packets

Oblivious routing = for every source and destination, distribution over paths

Oblivious Routing

Oblivious Routing

The path for any packet must be selected **obliviously**: without knowledge of other packets

Oblivious routing = for every source and destination, distribution over paths

Competitive: low **maximum expected edge congestion** compared to offline optimum

Oblivious Routing

Oblivious Routing

The path for any packet must be selected **obliviously**: without knowledge of other packets

Oblivious routing = for every source and destination, distribution over paths

Competitive: low **maximum expected edge congestion** compared to offline optimum

$\mathcal{O}(\log n)$-competitive oblivious routings

Oblivious Routing

Oblivious Routing

The path for any packet must be selected **obliviously**: without knowledge of other packets

Oblivious routing = for every source and destination, distribution over paths

Competitive: low **maximum expected edge congestion** compared to offline optimum

$\mathcal{O}(\log n)$-competitive oblivious routings

- exist [Räc08], and

Oblivious Routing

Oblivious Routing

The path for any packet must be selected **obliviously**: without knowledge of other packets

Oblivious routing = for every source and destination, distribution over paths

Competitive: low **maximum expected edge congestion** compared to offline optimum

$\mathcal{O}(\log n)$-competitive oblivious routings

- exist [Räc08], and
- are optimal [BL97; Mag+97]

Oblivious Routing

Oblivious Routing

The path for any packet must be selected **obliviously**: without knowledge of other packets

Oblivious routing = for every source and destination, distribution over paths

Competitive: low **maximum expected edge congestion** compared to offline optimum

$\mathcal{O}(\log n)$-competitive oblivious routings

- exist [Räc08], and
- are optimal [BL97; Mag+97]

Dilation can also be considered [GHZ21]

Possible Issues With Oblivious Routing

Inherently random!

Possible Issues With Oblivious Routing

Inherently random!

- Selecting single paths at best $\mathcal{O}(\sqrt{n})$-competitive [BH85; KKT91]

Possible Issues With Oblivious Routing

Inherently random!

- Selecting single paths at best $\mathcal{O}(\sqrt{n})$-competitive [BH85; KKT91]
- Need $\Omega(\sqrt{n})$ paths for every source-destination pair

Possible Issues With Oblivious Routing

Inherently random!

- Selecting single paths at best $\mathcal{O}(\sqrt{n})$-competitive [BH85; KKT91]
- Need $\Omega(\sqrt{n})$ paths for every source-destination pair

In practical settings, not all central control is expensive [Kum+18]:

Possible Issues With Oblivious Routing

Inherently random!

- Selecting single paths at best $\mathcal{O}(\sqrt{n})$-competitive [BH85; KKT91]
- Need $\Omega(\sqrt{n})$ paths for every source-destination pair

In practical settings, not all central control is expensive [Kum+18]:

- changing paths (support) is expensive, but

Possible Issues With Oblivious Routing

Inherently random!

- Selecting single paths at best $\mathcal{O}(\sqrt{n})$-competitive [BH85; KKT91]
- Need $\Omega(\sqrt{n})$ paths for every source-destination pair

In practical settings, not all central control is expensive [Kum+18]:

- changing paths (support) is expensive, but
- changing *sending ratios* (distribution) is cheap

Semi-Oblivious Routing

Semi-Oblivious Routing

For every source-destination pair, a set of *candidate paths* must be selected. Then, the *routing paths* are chosen centrally, with full information of the packets

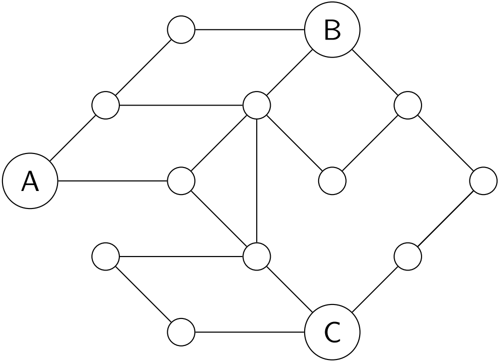
Semi-Oblivious Routing

Semi-Oblivious Routing

For every source-destination pair, a set of *candidate paths* must be selected. Then, the *routing paths* are chosen centrally, with full information of the packets

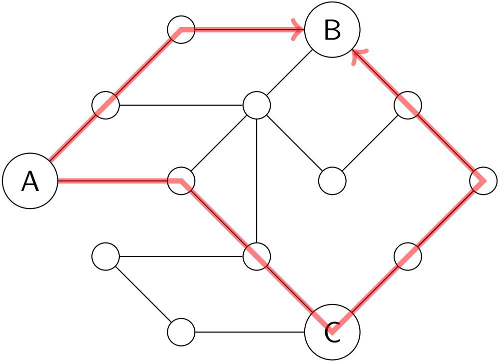
α-sparsity: can only select up to α paths between every source and destination

Semi-Oblivious Routing: Example



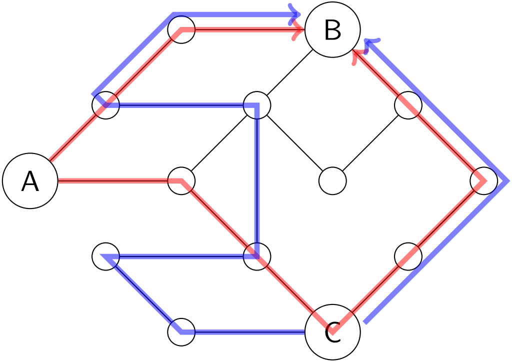
$$\alpha = 2$$

Semi-Oblivious Routing: Example



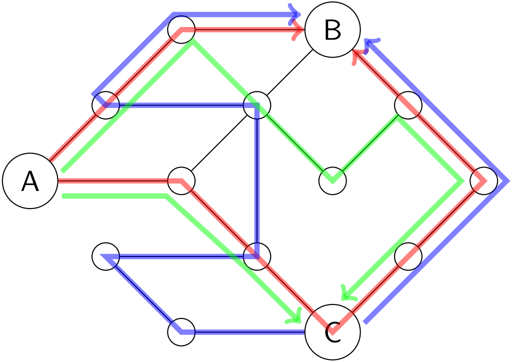
$$\alpha = 2$$

Semi-Oblivious Routing: Example



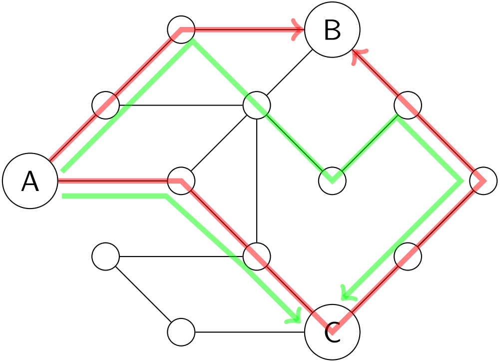
$\alpha = 2$

Semi-Oblivious Routing: Example



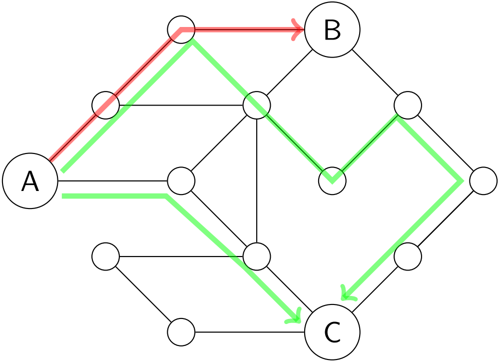
$\alpha = 2$

Semi-Oblivious Routing: Example



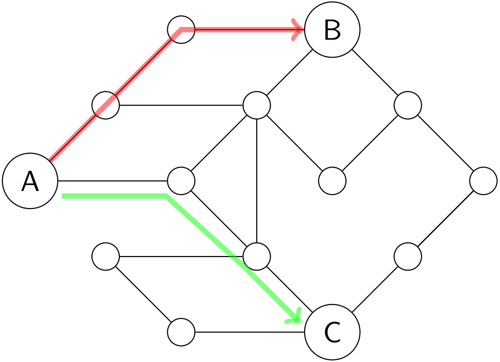
Packets: $A \to B$, $A \to C$

Semi-Oblivious Routing: Example



Packets: $A \to B$, $A \to C$

Semi-Oblivious Routing: Example



congestion 1
dilation 3

Practical Efficiency

Semi-oblivious routing is **efficient in practice** [Kum+18]

Practical Efficiency

Semi-oblivious routing is **efficient in practice** [Kum+18]

[Kum+18]'s construction is simple: sample $\alpha = 4$ paths from an oblivious routing, for every vertex pair

Practical Efficiency

Semi-oblivious routing is **efficient in practice** [Kum+18]

[Kum+18]'s construction is simple: sample $\alpha = 4$ paths from an oblivious routing, for every vertex pair

Practical efficiency despite **only negative** theoretical results

Semi-Oblivious Routing: What Is Known

Only negative results:

Only negative results:

- Polynomial α cannot be $o(\log(n)/\log\log n)$-competitive [HKL07]

Semi-Oblivious Routing: What Is Known

Only negative results:

- Polynomial α cannot be $o(\log(n)/\log\log n)$-competitive [HKL07]
- $\alpha = 1$ cannot be $\mathcal{O}(\sqrt{n})$-competitive [BH85; KKT91]

Our Results

Sparse semi-oblivious routings: $\alpha = \mathcal{O}(\log(n)/\log\log n)$ suffices for $\mathrm{poly}(\log n)$-competitiveness

Our Results

Sparse semi-oblivious routings: $\alpha = \mathcal{O}(\log(n)/\log\log n)$ suffices for $\mathrm{poly}(\log n)$-competitiveness

- Contrasts required support size of $\mathcal{O}(\sqrt{n})$ for oblivious routing.

Our Results

Sparse semi-oblivious routings: $\alpha = \mathcal{O}(\log(n)/\log\log n)$ suffices for $\mathrm{poly}(\log n)$-competitiveness

- Contrasts required support size of $\mathcal{O}(\sqrt{n})$ for oblivious routing.
- Construction by sampling: theoretical justification for practical efficiency!

Our Results

Sparse semi-oblivious routings: $\alpha = \mathcal{O}(\log(n)/\log\log n)$ suffices for $\mathrm{poly}(\log n)$-competitiveness

- Contrasts required support size of $\mathcal{O}(\sqrt{n})$ for oblivious routing.
- Construction by sampling: theoretical justification for practical efficiency!
- Congestion + dilation competitive with $\alpha = \mathcal{O}(\log^2 n)$

Our Results

Sparse semi-oblivious routings: $\alpha = \mathcal{O}(\log(n)/\log\log n)$ suffices for $\mathrm{poly}(\log n)$-competitiveness

- Contrasts required support size of $\mathcal{O}(\sqrt{n})$ for oblivious routing.
- Construction by sampling: theoretical justification for practical efficiency!
- Congestion + dilation competitive with $\alpha = \mathcal{O}(\log^2 n)$

A matching lower bound: $\alpha = o(\log(n)/\log\log n)$ cannot be $\mathrm{poly}(\log n)$-competitive

Construction

- Fix an oblivious routing
- Let $\alpha := \mathcal{O}(\log(n)/\log\log n)$
- For every vertex pair, sample α paths between those vertices from the oblivious routing

Construction

- Fix an oblivious routing
- Let $\alpha := \mathcal{O}(\log(n)/\log\log n)$
- For every vertex pair, sample α paths between those vertices from the oblivious routing

Theorem

A semi-oblivious routing constructed as above is $\mathcal{O}(\log^2 n)$-competitive with the sampled from oblivious routing

Construction

- Fix an oblivious routing
- Let $\alpha := \mathcal{O}(\log(n)/\log\log n)$
- For every vertex pair, sample α paths between those vertices from the oblivious routing

Theorem

A semi-oblivious routing constructed as above is $\mathcal{O}(\log^2 n)$-competitive with the sampled from oblivious routing

Note: competitive with the oblivious routing, not offline optimum

Proof Outline

Approach: Probabilistic method:

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n}{2}}$ possible packet sets

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n}{2}}$ possible packet sets
- But only $\binom{n}{2}^k = \exp(2k \log n)$ with k packets

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n}{2}}$ possible packet sets
- But only $\binom{n}{2}^k = \exp(2k \log n)$ with k packets

Fails catastrophically:

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n^2}{}}$ possible packet sets
- But only $\left(n^2\right)^k = \exp(2k \log n)$ with k packets

Fails catastrophically:

- Has to be extremely unlikely

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n}{2}}$ possible packet sets
- But only $\left(n^2\right)^k = \exp(2k \log n)$ with k packets

Fails catastrophically:

- Has to be extremely unlikely
- But no catastrophic failure has to guarantee progress

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n}{2}}$ possible packet sets
- But only $\left(n^2\right)^k = \exp(2k \log n)$ with k packets

Fails catastrophically:

- Has to be extremely unlikely
- But no catastrophic failure has to guarantee progress
- *Weak Routing:* route at least half of the packets

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n}{2}}$ possible packet sets
- But only $\left(n^2\right)^k = \exp(2k \log n)$ with k packets

Fails catastrophically:

- Has to be extremely unlikely
- But no catastrophic failure has to guarantee progress
- *Weak Routing:* route at least half of the packets
    - Competitively with oblivious routing congestion of **full** packet set

Proof Outline

Approach: Probabilistic method: Prove that for a fixed set of packets, a sampled semi-oblivious routing "fails catastrophically" with exponentially small probability in the number of packets

Easy finish: union bound over packet sets:

- Up to $2^{\binom{n}{2}}$ possible packet sets
- But only $\left(n^2\right)^k = \exp(2k \log n)$ with k packets

Fails catastrophically:

- Has to be extremely unlikely
- But no catastrophic failure has to guarantee progress
- *Weak Routing:* route at least half of the packets
  - Competitively with oblivious routing congestion of **full** packet set

  $\rightarrow$ can route any packet set, $\mathcal{O}(\log n)$ competitiveness loss

"Regular" Failure: Insufficient

Oblivious routing guarantees **every edge** has low expected congestion

"Regular" Failure: Insufficient

Oblivious routing guarantees **every edge** has low expected congestion

- Then, standard analysis gives *high probability* congestion is low

"Regular" Failure: Insufficient

Oblivious routing guarantees **every edge** has low expected congestion

- Then, standard analysis gives *high probability* congestion is low
  - High Probability: $1 - n^{-c}$ for an arbitrary constant c

"Regular" Failure: Insufficient

Oblivious routing guarantees **every edge** has low expected congestion

- Then, standard analysis gives *high probability* congestion is low
  - High Probability: $1 - n^{-c}$ for an arbitrary constant c
  - **Insufficient**!

"Regular" Failure: Insufficient

Oblivious routing guarantees **every edge** has low expected congestion

- Then, standard analysis gives *high probability* congestion is low
  - High Probability: $1 - n^{-c}$ for an arbitrary constant c
  - **Insufficient**!

"Cannot route everything" is too weak of a notion of failure

"Regular" Failure: Insufficient

Oblivious routing guarantees **every edge** has low expected congestion

- Then, standard analysis gives *high probability* congestion is low
    - High Probability: $1 - n^{-c}$ for an arbitrary constant c
    - **Insufficient**!

"Cannot route everything" is too weak of a notion of failure

Bounding probability of a catastrophic failure needs a more involved analysis

Dynamic Process

Fix a packet set, and sample $\alpha := \mathcal{O}(\log(n)/\log\log n)$ paths between every vertex pair a packet wants to traverse between

Dynamic Process

Fix a packet set, and sample $\alpha := \mathcal{O}(\log(n)/\log\log n)$ paths between every vertex pair a packet wants to traverse between

Loop over edges. If the number of enabled paths crossing the edge is greater than $C_{\max} := \mathrm{OBL}(\text{packet set}) \cdot \mathcal{O}(\log n)$, **disable** all paths over the edge

Dynamic Process

Fix a packet set, and sample $\alpha := \mathcal{O}(\log(n)/\log\log n)$ paths between every vertex pair a packet wants to traverse between

Loop over edges. If the number of enabled paths crossing the edge is greater than $C_{\max} := \mathrm{OBL}(\text{packet set}) \cdot \mathcal{O}(\log n)$, **disable** all paths over the edge

At the end: enabled paths have low congestion!

Dynamic Process

Fix a packet set, and sample $\alpha := \mathcal{O}(\log(n)/\log\log n)$ paths between every vertex pair a packet wants to traverse between

Loop over edges. If the number of enabled paths crossing the edge is greater than $C_{\max} := \mathrm{OBL}(\text{packet set}) \cdot \mathcal{O}(\log n)$, **disable** all paths over the edge

At the end: enabled paths have low congestion!

- Ideally: half of packets have at least one path

Dynamic Process

Fix a packet set, and sample $\alpha := \mathcal{O}(\log(n)/\log\log n)$ paths between every vertex pair a packet wants to traverse between

Loop over edges. If the number of enabled paths crossing the edge is greater than $C_{\max} := \mathrm{OBL}(\text{packet set}) \cdot \mathcal{O}(\log n)$, **disable** all paths over the edge

At the end: enabled paths have low congestion!

- Ideally: half of packets have at least one path
- Sufficient: at most half of total paths are disabled

Dynamic Process

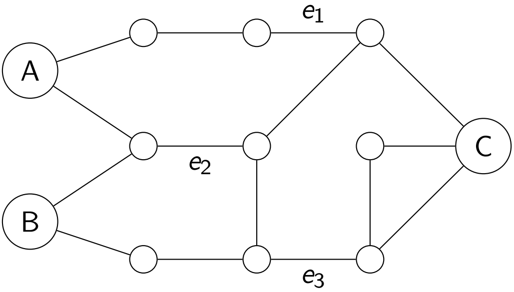
Fix a packet set, and sample $\alpha := \mathcal{O}(\log(n)/\log\log n)$ paths between every vertex pair a packet wants to traverse between

Loop over edges. If the number of enabled paths crossing the edge is greater than $C_{\max} := \mathrm{OBL}(\text{packet set}) \cdot \mathcal{O}(\log n)$, **disable** all paths over the edge

At the end: enabled paths have low congestion!

- Ideally: half of packets have at least one path
- Sufficient: at most half of total paths are disabled
- Bound this probability!

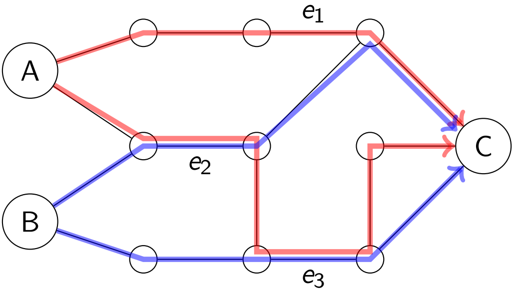
Dynamic Process: Example



$\alpha = 2,\ C_{\max} = 1$

Packets: $A \rightarrow C,\ B \rightarrow C$

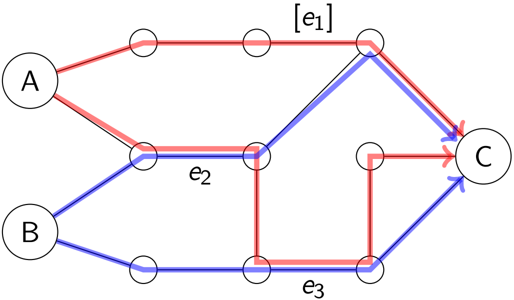
Dynamic Process: Example



$\alpha = 2,\ C_{\text{max}} = 1$

Packets: $A \to C,\ B \to C$

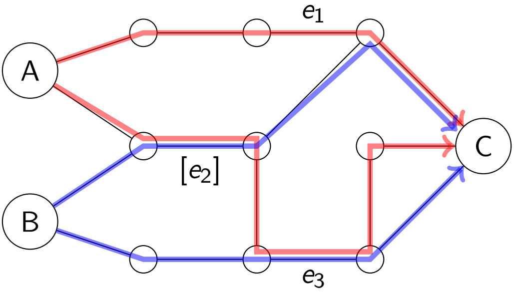
Dynamic Process: Example



$\alpha = 2,\ C_{\max} = 1$

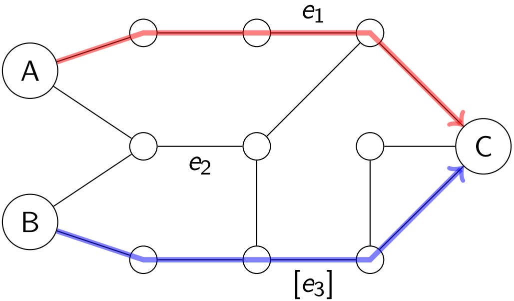
Packets: $A \to C,\ B \to C$

Dynamic Process: Example



$$\alpha = 2,\ C_{\max} = 1$$
Packets: $A \to C,\ B \to C$

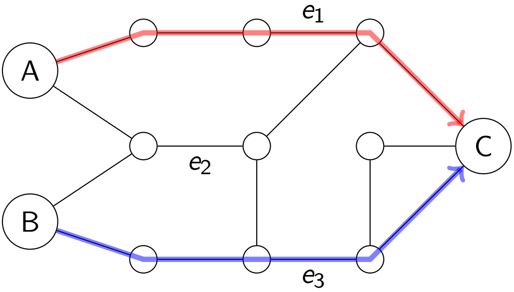
Dynamic Process: Example



$\alpha = 2,\ C_{\max} = 1$

Packets: $A \to C,\ B \to C$

Dynamic Process: Example



$\alpha = 2,\ C_{\max} = 1$

Packets: $A \rightarrow C,\ B \rightarrow C$

Intuitive Analysis

How many paths are disabled at a single edge?

Intuitive Analysis

How many paths are disabled at a single edge?

- "Does fixed sample cross the edge": $\{0, 1\}$-random variable

Intuitive Analysis

How many paths are disabled at a single edge?

- "Does fixed sample cross the edge": $\{0,1\}$-random variable
- Number of paths disabled at $e \leq$ number of initial sampled paths crossing e

Intuitive Analysis

How many paths are disabled at a single edge?

- "Does fixed sample cross the edge": $\{0, 1\}$-random variable
- Number of paths disabled at $e \leq$ number of initial sampled paths crossing e

Intuitive Analysis

How many paths are disabled at a single edge?

- "Does fixed sample cross the edge": $\{0, 1\}$-random variable
- Number of paths disabled at $e \leq$ number of initial sampled paths crossing e

$X_e :=$ sampled paths that cross edge e

$$\mathbb{E}[X_e] \leq \mathrm{OBL}(\text{packet set}) \cdot \alpha$$

Intuitive Analysis

How many paths are disabled at a single edge?

- "Does fixed sample cross the edge": $\{0,1\}$-random variable
- Number of paths disabled at $e \leq$ number of initial sampled paths crossing e

$X_e :=$ sampled paths that cross edge e

$$\mathbb{E}[X_e] \leq \mathrm{OBL}(\text{packet set}) \cdot \alpha$$

Recall $C_{\max} := \mathrm{OBL}(\text{packet set}) \cdot \mathcal{O}(\log n) > 2\mathbb{E}[X_e]$

$$\mathbb{I}[k \geq C_{\max}] \cdot \mathbb{P}(X_e > k) \leq \exp(-k/2)$$

Intuitive Analysis 2

For a single edge, we have

$$\mathbb{I}[k \geq C_{\max}] \cdot \mathbb{P}(X_e > k) \leq \exp(-k/2)$$

Intuitive Analysis 2

For a single edge, we have

$$\mathbb{I}[k \geq C_{\max}] \cdot \mathbb{P}(X_e > k) \leq \exp(-k/2)$$

Would be nice to have:

$$\mathbb{P}\left(\sum_e X_e \cdot \mathbb{I}[X_e \geq C_{\max}] > k\right) \leq \exp(-k/2)$$

Intuitive Analysis 2

For a single edge, we have

$$\mathbb{I}[k \geq C_{\max}] \cdot \mathbb{P}(X_e > k) \leq \exp(-k/2)$$

Would be nice to have:

$$\mathbb{P}\left(\sum_e X_e \cdot \mathbb{I}[X_e \geq C_{\max}] > k\right) \leq \exp(-k/2)$$

Then: more than half disabled $\rightarrow$ probability $\exp(-\alpha|\text{packet set}|/4)$

Intuitive Analysis 2

For a single edge, we have

$$\mathbb{I}[k \geq C_{\max}] \cdot \mathbb{P}(X_e > k) \leq \exp(-k/2)$$

Would be nice to have:

$$\mathbb{P}\left(\sum_e X_e \cdot \mathbb{I}[X_e \geq C_{\max}] > k\right) \leq \exp(-k/2)$$

Then: more than half disabled $\rightarrow$ probability $\exp(-\alpha|\text{packet set}|/4)$

- Recall: $\exp(2|\text{packet set}|\log n)$ packet sets of size $|\text{packet set}|$

Intuitive Analysis 2

For a single edge, we have

$$\mathbb{I}[k \geq C_{\max}] \cdot \mathbb{P}(X_e > k) \leq \exp(-k/2)$$

Would be nice to have:

$$\mathbb{P}\left(\sum_e X_e \cdot \mathbb{I}[X_e \geq C_{\max}] > k\right) \leq \exp(-k/2)$$

Then: more than half disabled $\rightarrow$ probability $\exp(-\alpha|\text{packet set}|/4)$

- Recall: $\exp(2|\text{packet set}| \log n)$ packet sets of size $|\text{packet set}|$
- $\alpha = \mathcal{O}(\log n)$ sufficient!

Summary & Open Problems

Main Result

In any graph, there exists a semi-oblivious routing with sparsity $\alpha = \mathcal{O}(\log(n)/\log\log n)$ that is $\mathrm{poly}(\log n)$-competitive with the offline optimum

Summary & Open Problems

Main Result

In any graph, there exists a semi-oblivious routing with sparsity $\alpha = \mathcal{O}(\log(n)/\log\log n)$ that is $\mathrm{poly}(\log n)$-competitive with the offline optimum

Open Problems:

- Dependence on centralised control: can you deterministically, competitively deliver packets in a distributed model with semi-oblivious routing?

Summary & Open Problems

Main Result

In any graph, there exists a semi-oblivious routing with sparsity $\alpha = \mathcal{O}(\log(n)/\log\log n)$ that is $\mathrm{poly}(\log n)$-competitive with the offline optimum

Open Problems:

- Dependence on centralised control: can you deterministically, competitively deliver packets in a distributed model with semi-oblivious routing?
  - Coming soon: **yes!**

Summary & Open Problems

Main Result

In any graph, there exists a semi-oblivious routing with sparsity $\alpha = \mathcal{O}(\log(n)/\log\log n)$ that is $\mathrm{poly}(\log n)$-competitive with the offline optimum

Open Problems:

- Dependence on centralised control: can you deterministically, competitively deliver packets in a distributed model with semi-oblivious routing?
  - Coming soon: **yes!**
  - But each vertex needs to know the semi-oblivious routing, can you construct one?

Summary & Open Problems

Main Result

In any graph, there exists a semi-oblivious routing with sparsity $\alpha = \mathcal{O}(\log(n)/\log\log n)$ that is $\mathrm{poly}(\log n)$-competitive with the offline optimum

Open Problems:

- Dependence on centralised control: can you deterministically, competitively deliver packets in a distributed model with semi-oblivious routing?
  - Coming soon: **yes!**
  - But each vertex needs to know the semi-oblivious routing, can you construct one?
- Do sparse semi-oblivious routings with *small routing tables* exist?

Thank you!
Questions?

Full version of paper