A simpler proof for $O(\textbf{congestion} + \textbf{dilation})$ packet routing

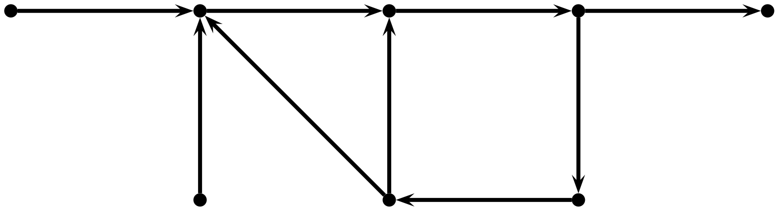
Thomas Rothvoß

Department of Mathematics, MIT

IPCO 2013

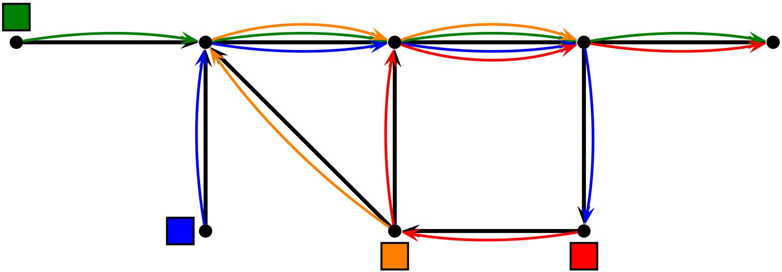
Massachusetts Institute of Technology

Packet Routing



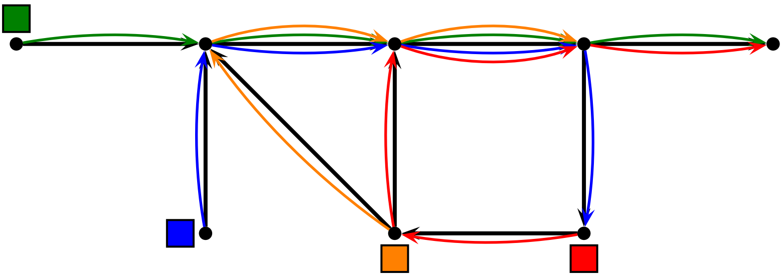
- **Input:** directed graph $G = (V, E)$

Packet Routing



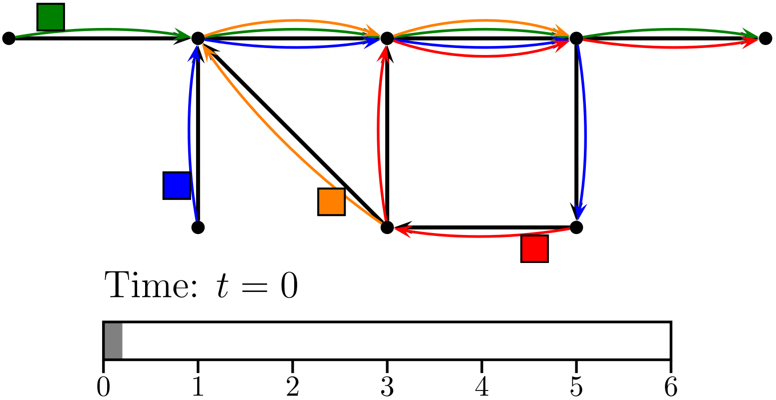
- **Input:** Paths P_i in a directed graph $G = (V, E)$

Packet Routing



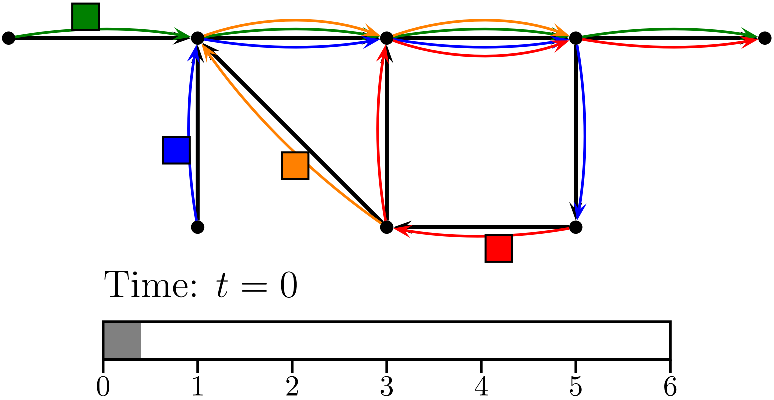
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



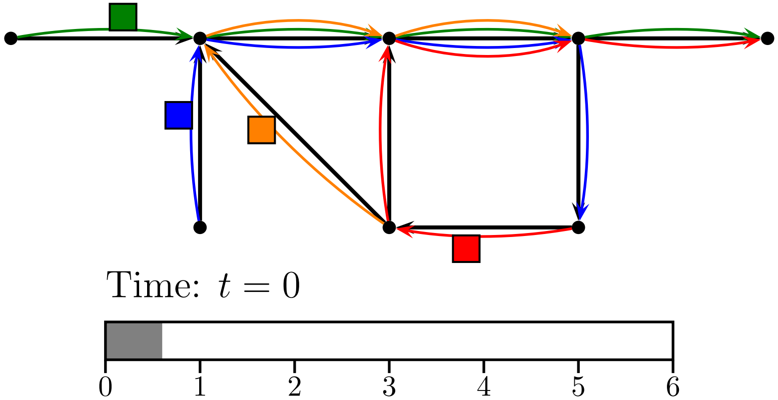
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



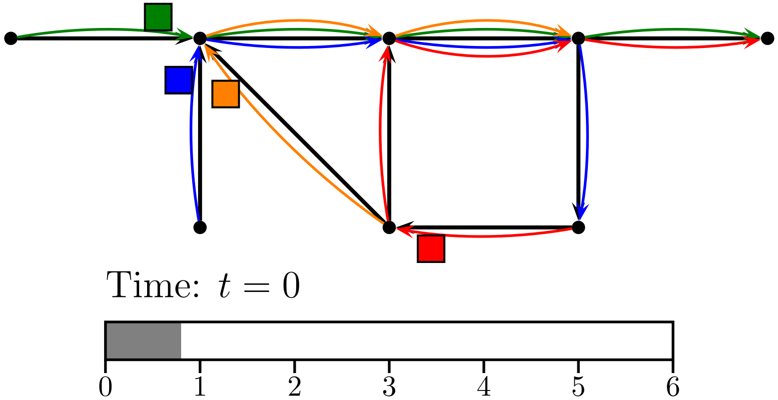
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



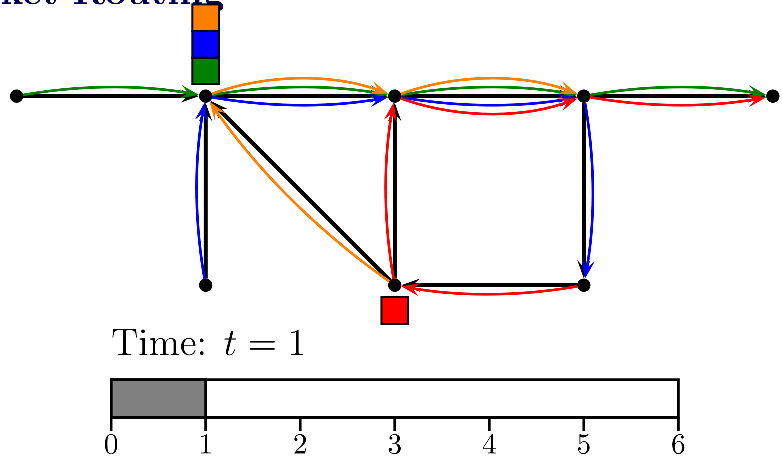
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



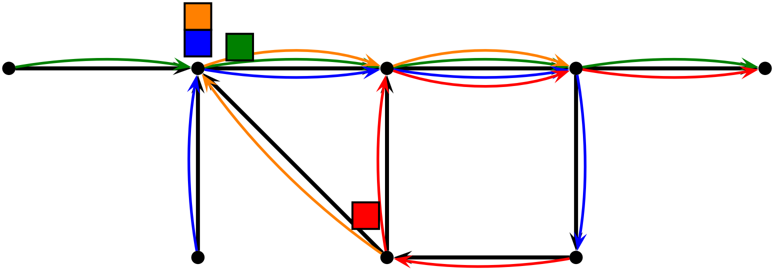
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

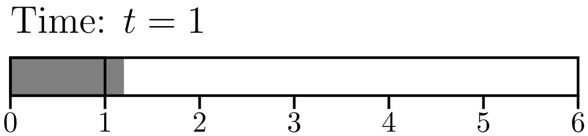
Packet Routing



- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

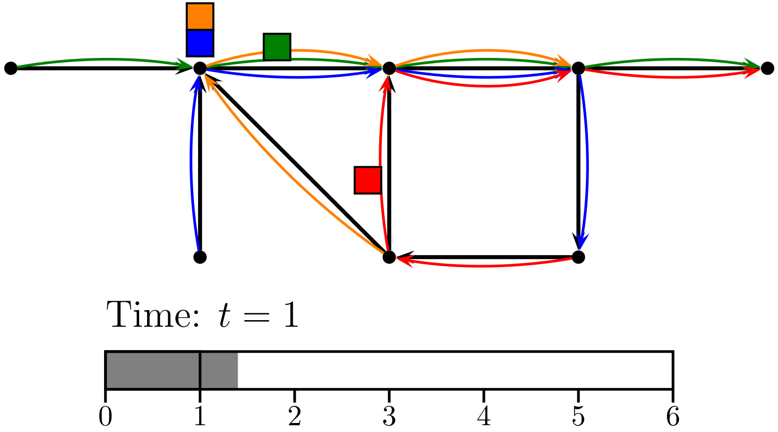
Packet Routing





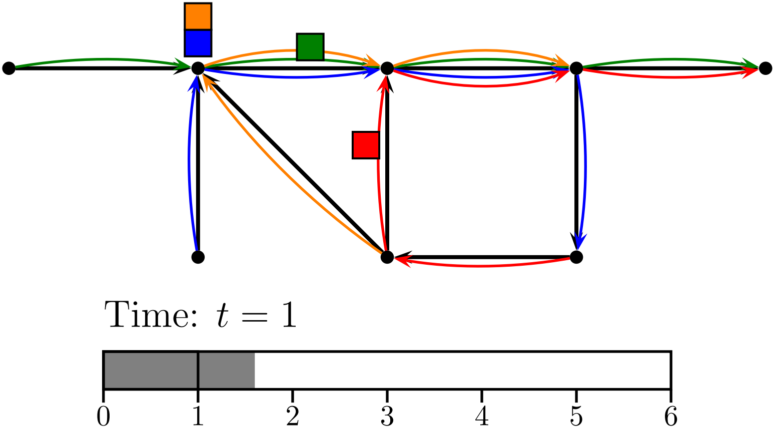
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



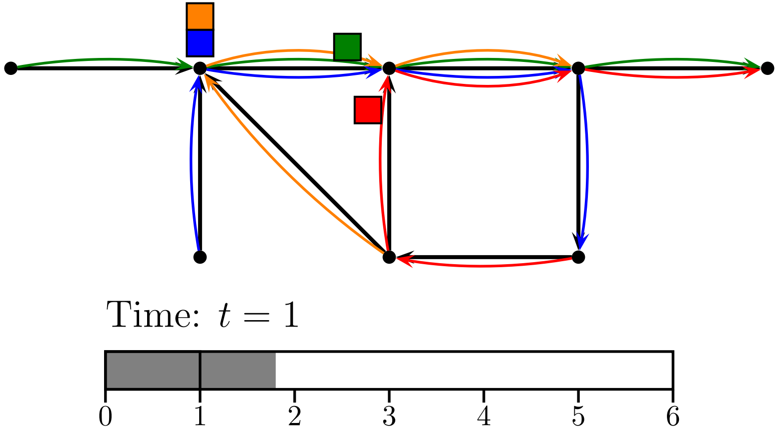
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



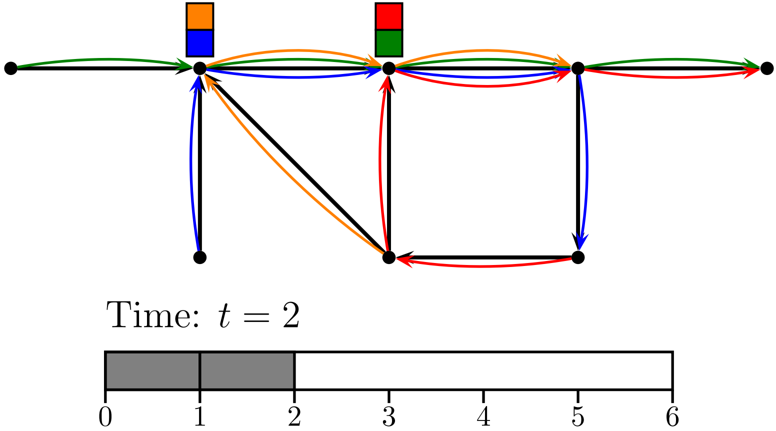
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



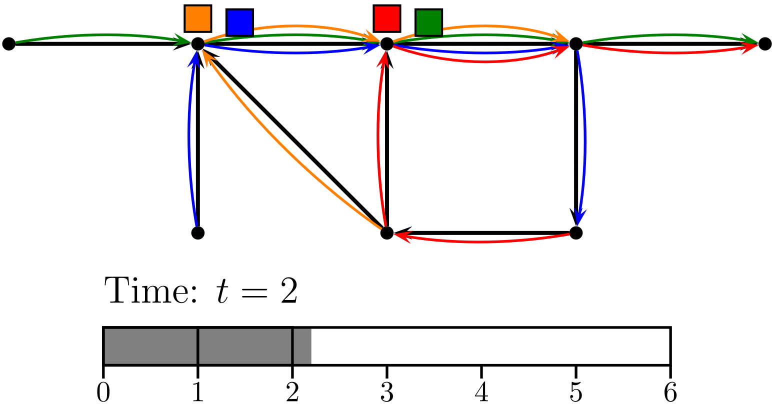
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



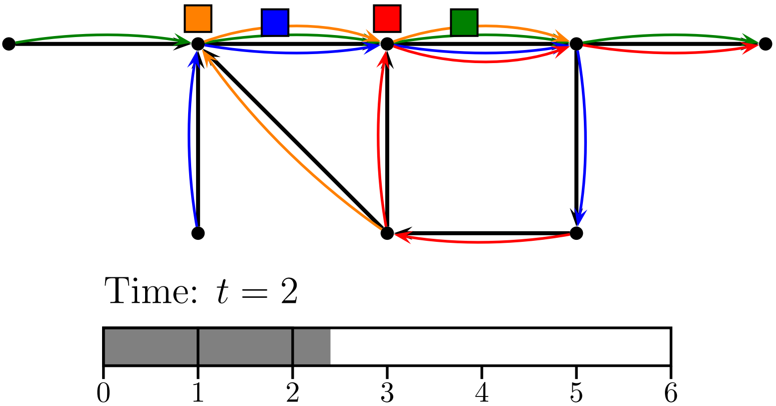
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



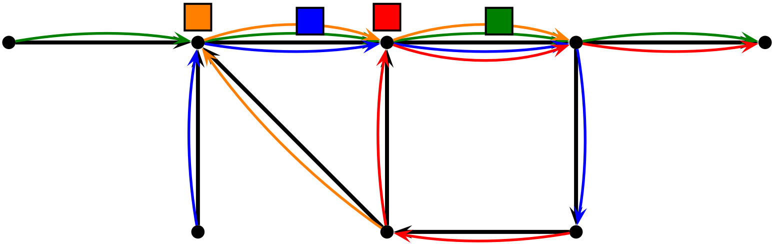
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

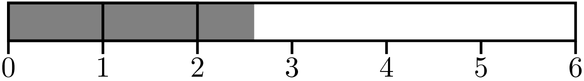


- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

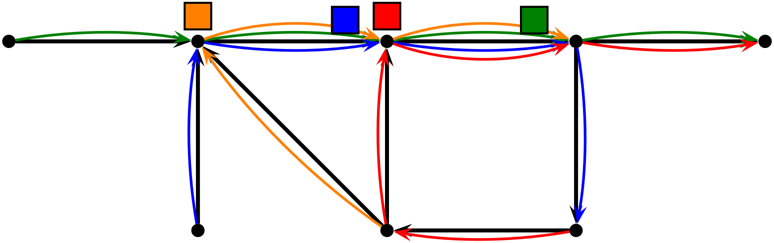


Time: $t = 2$

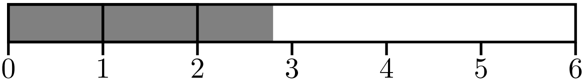


- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

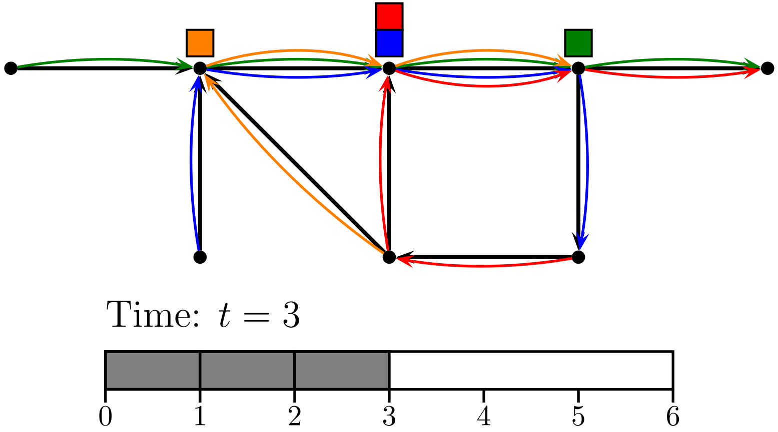


Time: $t = 2$



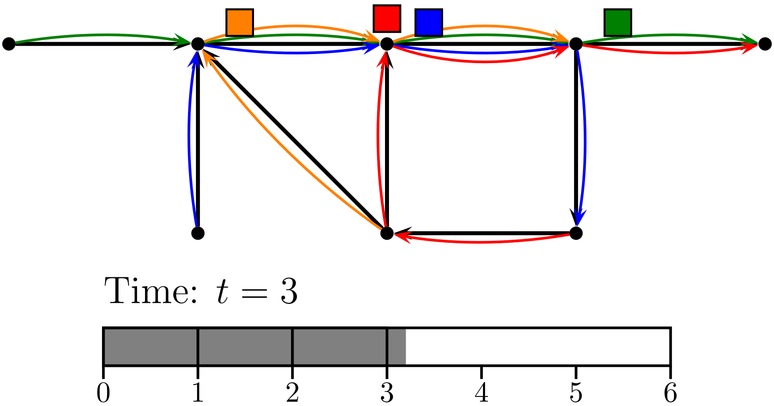
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



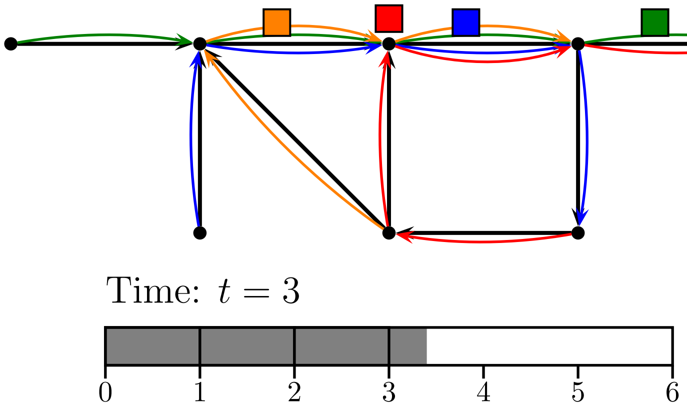
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



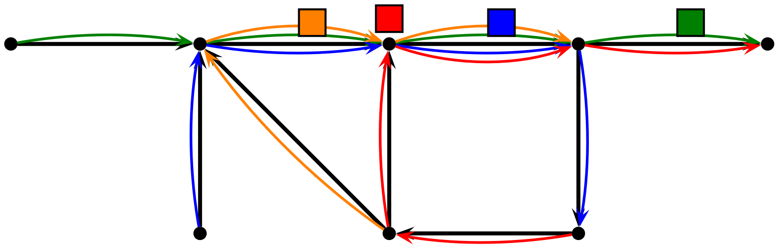
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

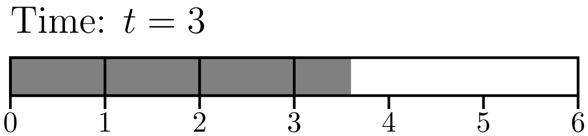
Packet Routing



- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

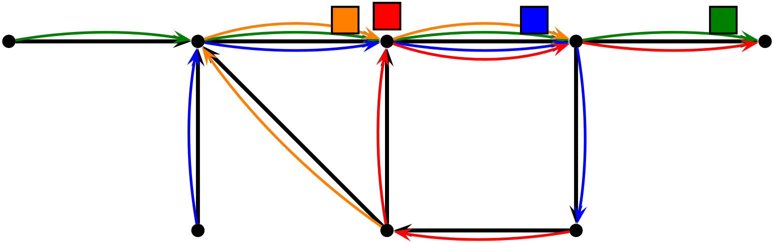
Packet Routing

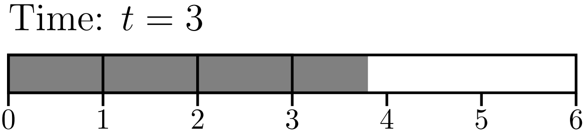




- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

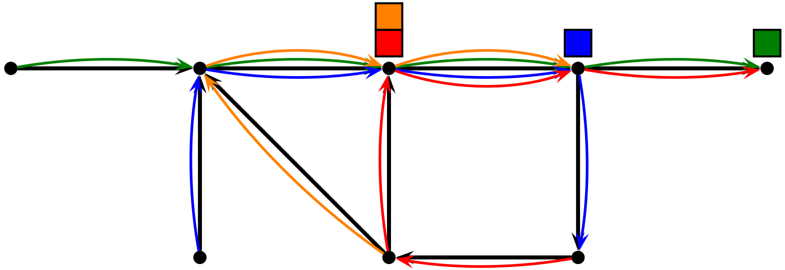
Packet Routing



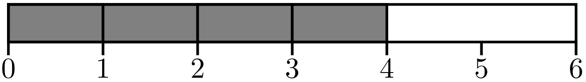


- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

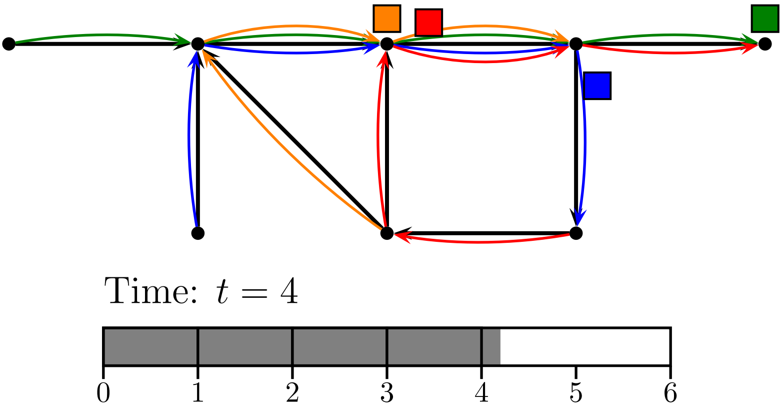


Time: $t = 4$



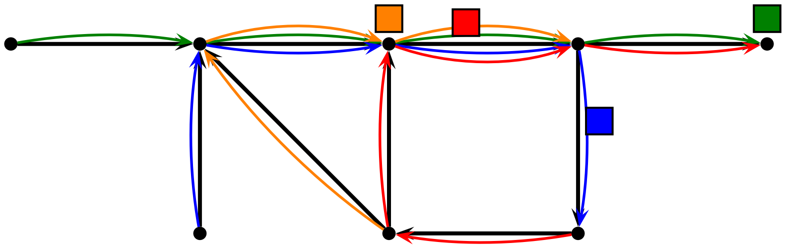
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

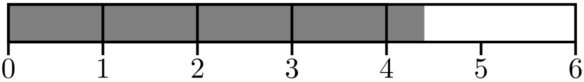


- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

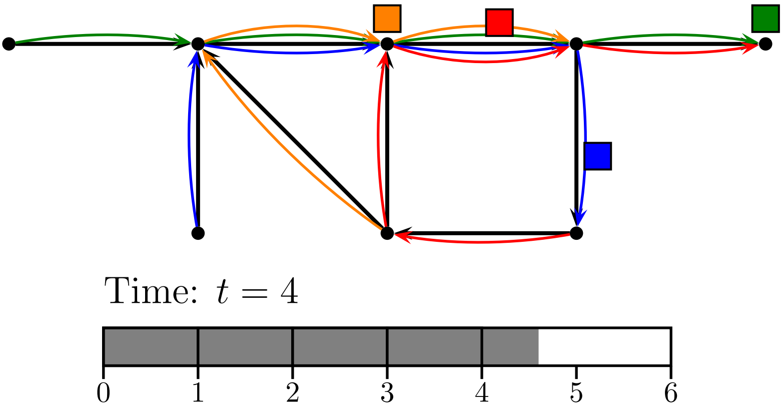


Time: $t = 4$



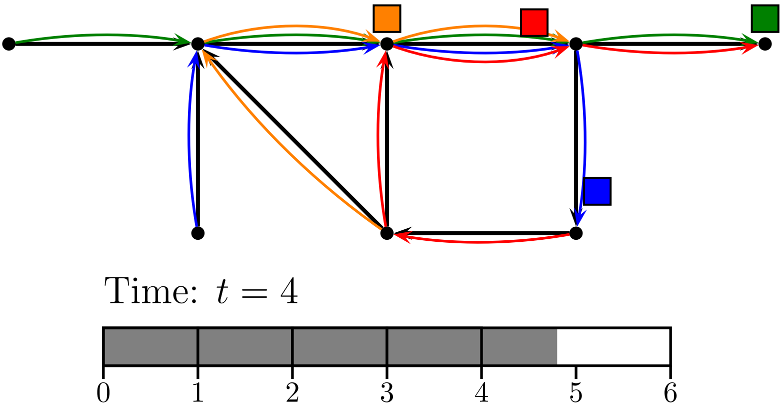
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



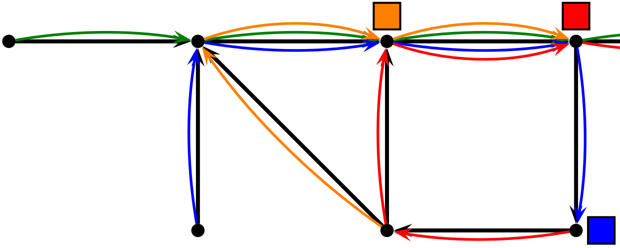
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

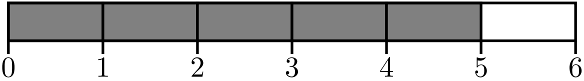


- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

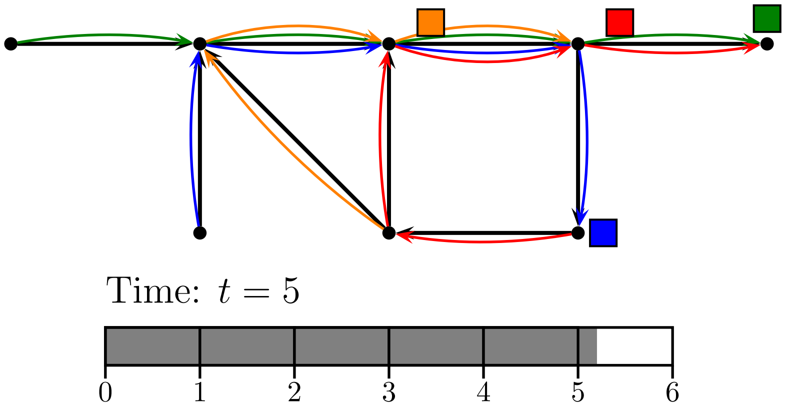


Time: $t = 5$



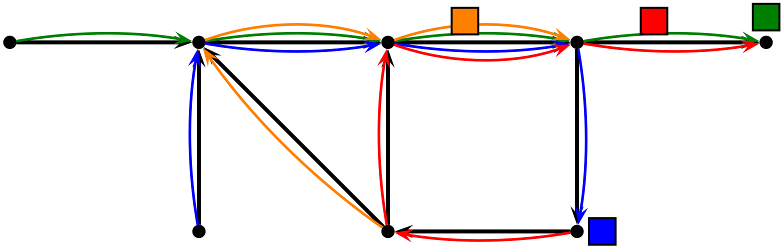
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

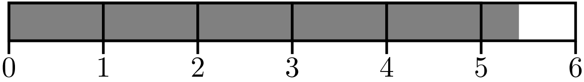


- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

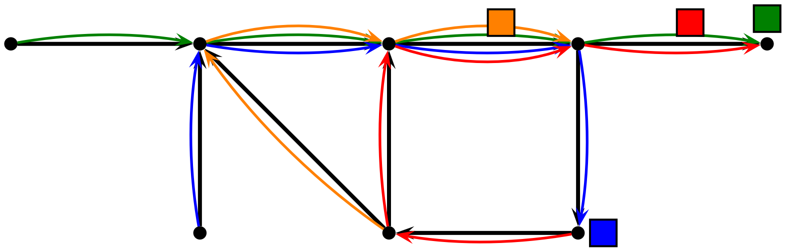


Time: $t = 5$

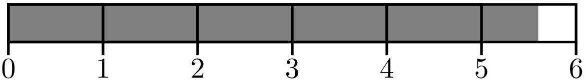


- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing

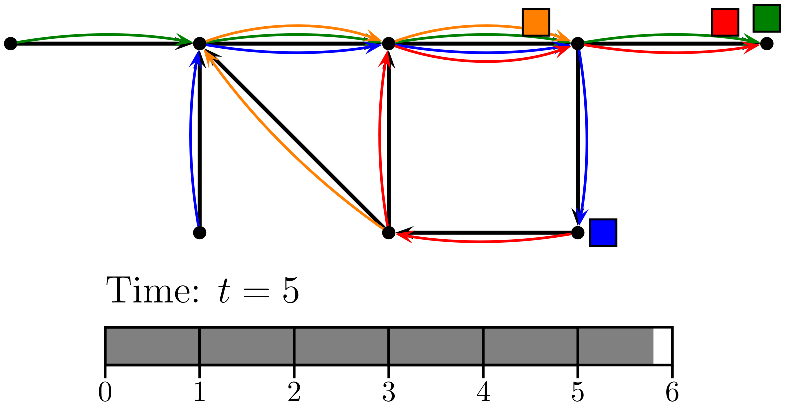


Time: $t = 5$



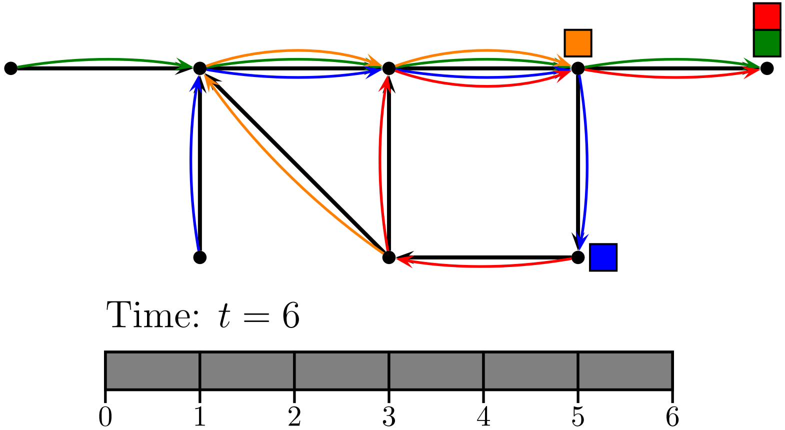
- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing



- **Input:** Paths P_i in a directed graph $G = (V, E)$
- **Goal:** Route packets along paths to minimize makespan
  Constraint: edge can be crossed by 1 packet per time unit

Packet Routing (2)

Lower bounds:

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E} \{\#i : e \in P_i\}$

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E}\{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E} \{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E} \{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

"*highly involved*"

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E} \{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

"highly involved"

"very difficult to understand"

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E}\{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

"highly involved"

"very difficult to understand"

"technical tour de force"

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E} \{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

"highly involved"

"very difficult to understand"

"technical tour de force"

- Polytime algorithm [Leighton, Maggs, Richa '99]

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E} \{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

"*highly involved*"

"*very difficult to understand*"

"*technical tour de force*"

- Polytime algorithm [Leighton, Maggs, Richa '99]
- $39 \cdot (C + D)$ suffices [Scheideler '98]

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E}\{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

"highly involved"

"very difficult to understand"

"technical tour de force"

- Polytime algorithm [Leighton, Maggs, Richa '99]
- $39 \cdot (C + D)$ suffices [Scheideler '98]
- $24 \cdot (C + D)$ suffices [Peis, Wiese '11]

Packet Routing (2)

Lower bounds:

- **congestion** $= \max_{e \in E} \{\#i : e \in P_i\}$
- **dilation** $= \max_i |P_i|$

Theorem (Leighton, Maggs, Rao '94)

$\exists$ schedule of length $\boldsymbol{O}$**(congestion + dilation)**
(even with $O(1)$-size edge buffers).

"highly involved"

"very difficult to understand"

"technical tour de force"

- Polytime algorithm [Leighton, Maggs, Richa '99]
- $39 \cdot (C + D)$ suffices [Scheideler '98]
- $24 \cdot (C + D)$ suffices [Peis, Wiese '11]
- $O(1)$-apx for finding paths + schedule [Srinivasan, Teo '00]

Main result

Theorem (R. '13)

Much simpler proof of $O(\text{congestion} + \text{dilation})$-packet routing (also with $O(1)$-size edge buffers).

Main result

Theorem (R. '13)

Much simpler proof of O(congestion + dilation)-packet routing (also with $O(1)$-size edge buffers).

- [Wiese '12]: Is $(1 + o(1)) \cdot$ (congestion + dilation) possible?
- **True** if congestion $\gg$ dilation!

Main result

Theorem (R. '13)

Much simpler proof of O(congestion + dilation)-packet routing (also with $O(1)$-size edge buffers).

- [Wiese '12]: Is $(1 + o(1)) \cdot$ (congestion + dilation) possible? No!!
- **True** if congestion $\gg$ dilation!

Theorem (R. '13)

$\exists$ instance requiring $(\mathbf{1} + \boldsymbol{\varepsilon}) \cdot$ (congestion + dilation) time.

Main result

Theorem (R. '13)

Much simpler proof of O(congestion + dilation)-packet routing (also with $O(1)$-size edge buffers).

- [Wiese '12]: Is $(1 + o(1)) \cdot$ (congestion + dilation) possible? No!!
- **True** if congestion $\gg$ dilation!

Theorem (R. '13)

$\exists$ instance requiring $(\mathbf{1} + \boldsymbol{\varepsilon}) \cdot$ (congestion + dilation) time.

Assumptions:

- $D :=$ dilation = congestion = $|P_i| \ \forall i$

Main result

Theorem (R. '13)

Much simpler proof of O(congestion + dilation)-packet routing (also with $O(1)$-size edge buffers).

- [Wiese '12]: Is $(1 + o(1)) \cdot$ (congestion + dilation) possible? No!!
- **True** if congestion $\gg$ dilation!

Theorem (R. '13)

$\exists$ instance requiring $(\mathbf{1} + \boldsymbol{\varepsilon}) \cdot$ (congestion + dilation) time.

Assumptions:

- $D :=$ dilation = congestion = $|P_i|\ \forall i$
- $O(1)$ packets can cross an edge per time unit

Preliminaries

Lemma (Lovász Local Lemma)

Let $A_1, \ldots, A_m$ *be events such that*

(1) $\Pr[A_i] \le p$

(2) *each* A_i *depends on* $\le d$ *other events*

(3) $4 \cdot p \cdot d \le 1$

Then $\Pr\left[\bigcap_{i=1}^{m} \bar{A}_i\right] > 0$.

- Constructive via [Moser, Tardos '10]

Preliminaries

Lemma (Lovász Local Lemma)

Let $A_1, \ldots, A_m$ be events such that

(1) $\Pr[A_i] \leq p$

(2) *each A_i depends on $\leq d$ other events*

(3) $4 \cdot p \cdot d \leq 1$

Then $\Pr\left[\bigcap_{i=1}^m \bar{A}_i\right] > 0$.

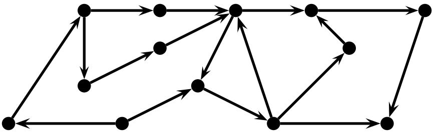
- Constructive via [Moser, Tardos '10]

Lemma (Chernov-Hoeffding)

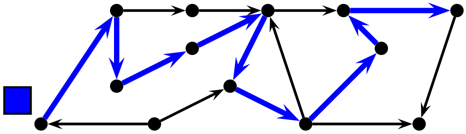
Let $Z_1, \ldots, Z_k \in [0, \delta]$ be independently RV, sum $Z := \sum_{i=1}^k Z_i$. Then

$$\Pr[Z > (1+\varepsilon)\,\mathbb{E}[Z]] \leq \exp\left(-\frac{\varepsilon^2}{3} \cdot \frac{\mathbb{E}[Z]}{\delta}\right).$$

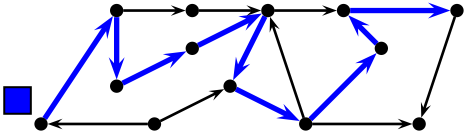
A probabilistic schedule



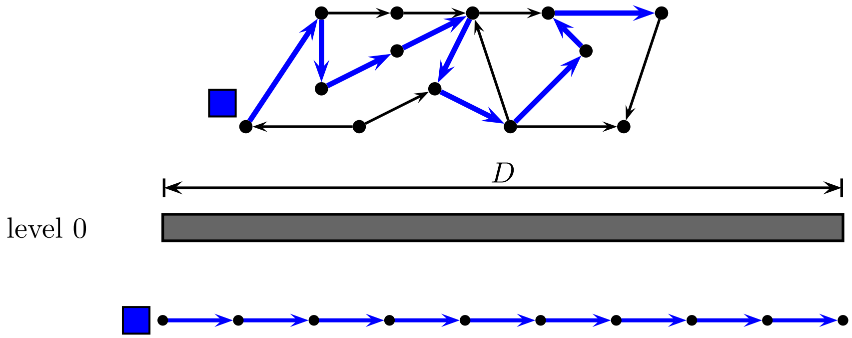
A probabilistic schedule



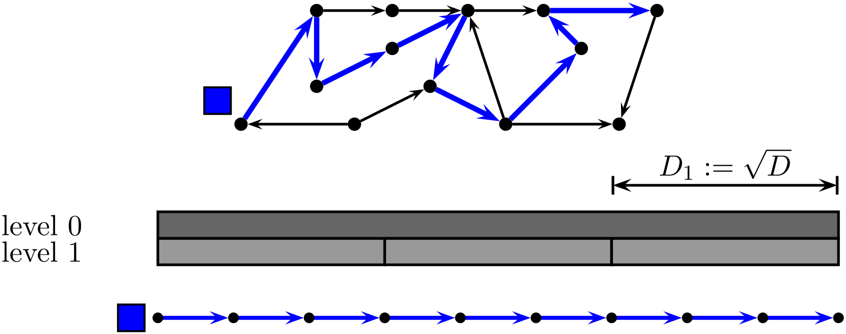
A probabilistic schedule



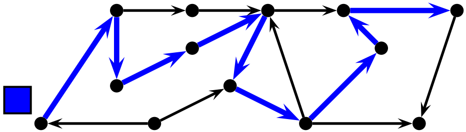
A probabilistic schedule

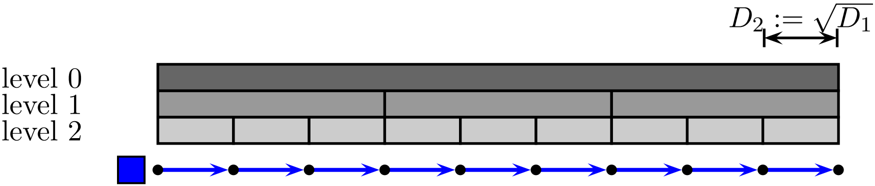


A probabilistic schedule

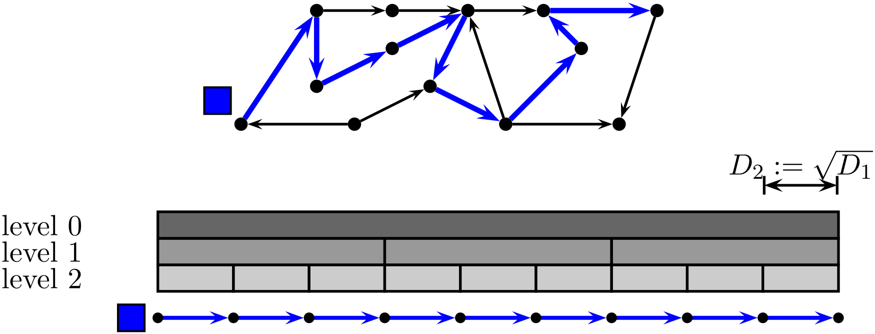


A probabilistic schedule





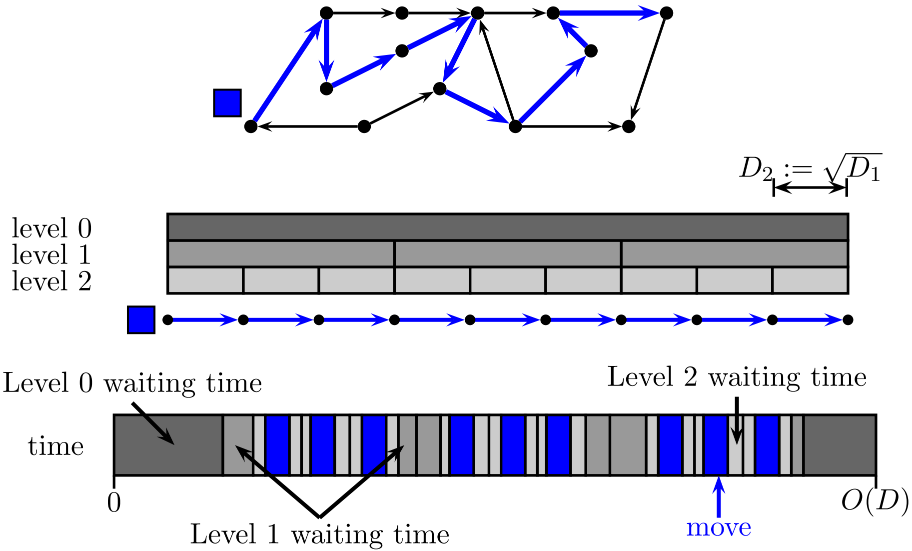
A probabilistic schedule



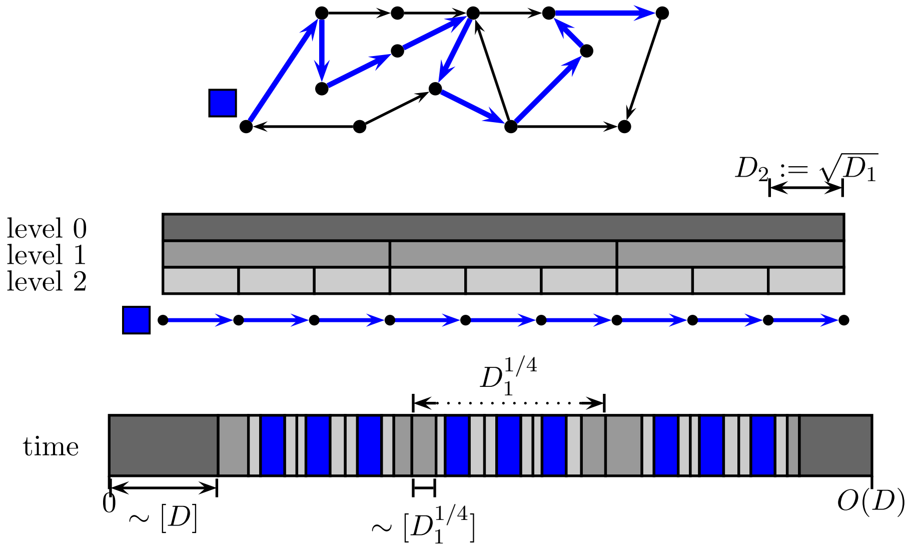
Waiting rule:

- At source: wait $\alpha_0 \sim [D]$
- When **entering** kth level ℓ interval: Wait $\alpha_{\ell,k} \sim [D_\ell^{1/4}]$
- When **leaving** kth level ℓ interval: Wait $D_\ell^{1/4} - \alpha_{\ell,k}$

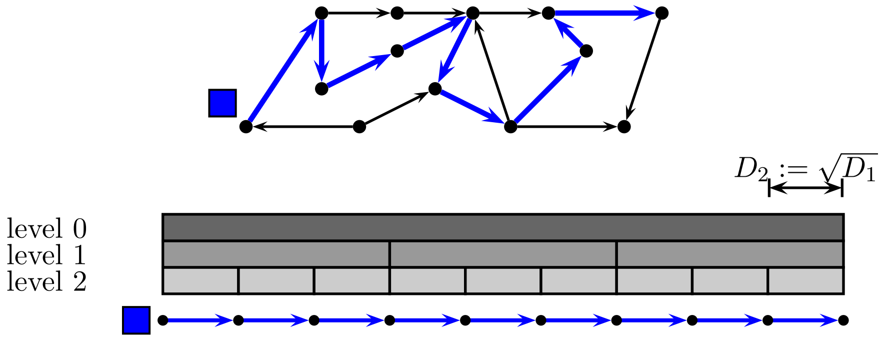
A probabilistic schedule



A probabilistic schedule



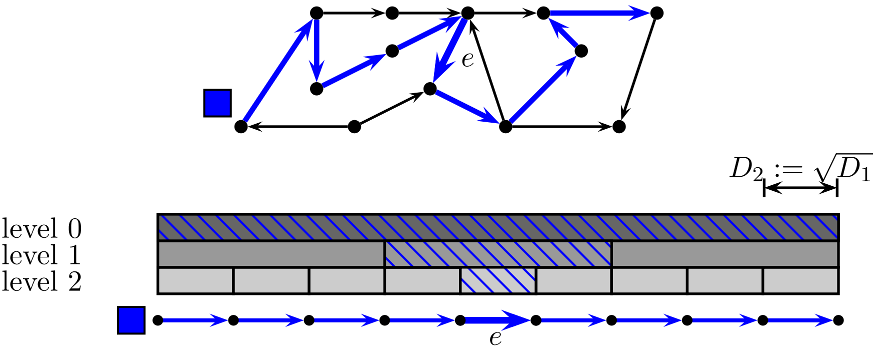
A probabilistic schedule



Observations:

- total waiting time: $D + \sum_{\ell \geq 1} \frac{D}{D_\ell} \cdot D_\ell^{1/4} = O(D)$

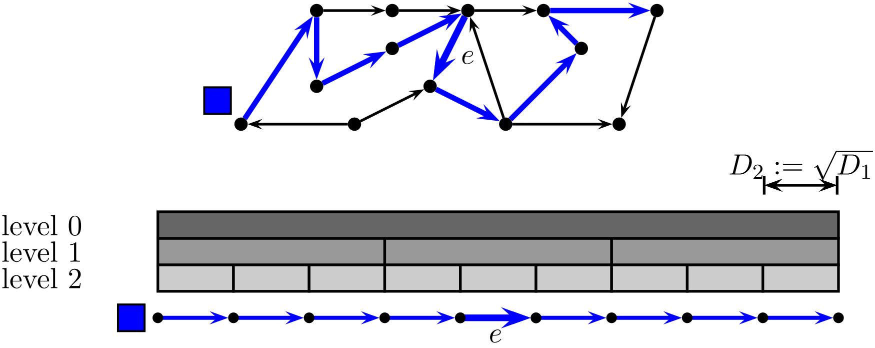
A probabilistic schedule



Observations:

- total waiting time: $D + \sum_{\ell \geq 1} \frac{D}{D_\ell} \cdot D_\ell^{1/4} = O(D)$
- time that i crosses e depends only on waiting times of intervals containing e

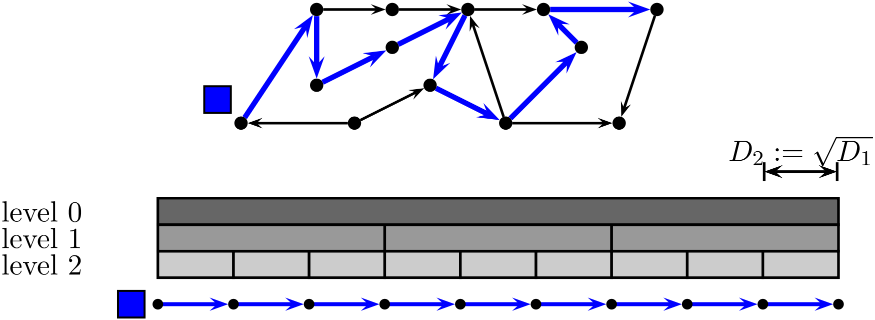
A probabilistic schedule



Observations:

- total waiting time: $D + \sum_{\ell \geq 1} \frac{D}{D_\ell} \cdot D_\ell^{1/4} = O(D)$
- time that i crosses e depends only on waiting times of intervals containing e
- $\Pr[\text{packet } i \text{ crosses } e \text{ at time } t] \leq \frac{1}{D}$ & $\mathbb{E}[\text{load}(e,t)] \leq 1$

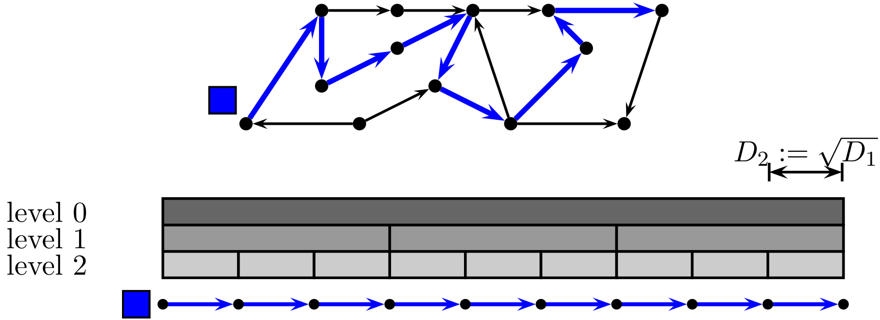
A probabilistic schedule



Claim

There **exist** waiting times s.t. load$(e,t) \leq O(1)$ $\forall e, t$

A probabilistic schedule



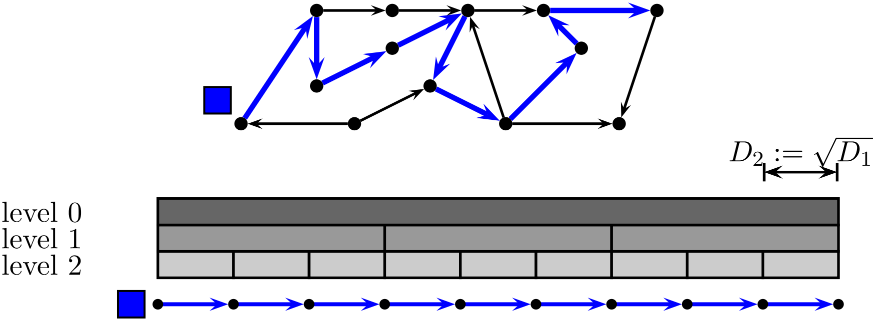
Claim

There **exist** waiting times s.t. $\text{load}(e,t) \leq O(1)\ \forall e,t$

Idea:

- Fix waiting times on level $\ell = 0, 1, 2, \ldots$ iteratively.

A probabilistic schedule



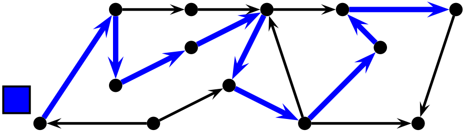
Claim

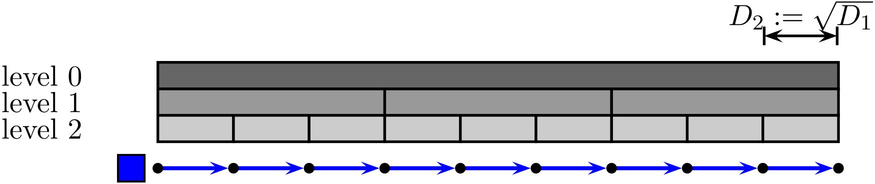
There **exist** waiting times s.t. $\text{load}(e,t) \leq O(1)\ \forall e,t$

Idea:

- Fix waiting times on level $\ell = 0, 1, 2, \ldots$ iteratively.
- Show $\max_{e,t}\{\mathbb{E}[\text{load}(e,t)]\}$ increases $\leq D_\ell^{-\frac{1}{32}}$ in step ℓ.

A probabilistic schedule





Claim

There **exist** waiting times s.t. $\text{load}(e,t) \leq O(1)\ \forall e,t$

Idea:

- Fix waiting times on level $\ell = 0, 1, 2, \ldots$ iteratively.
- Show $\max_{e,t}\{\mathbb{E}[\text{load}(e,t)]\}$ increases $\leq D_\ell^{-\frac{1}{32}}$ in step ℓ.
- Eventually $\text{load}(e,t) \leq 1 + \sum_{\ell \geq 0} D_\ell^{-\frac{1}{32}} \leq O(1)$.

Proof

- Pick level-0 waiting times $\boldsymbol{\alpha} \sim [D]^n$.
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] =$ ave. load on e at t dep. on $\boldsymbol{\alpha}$

Lemma

$\Pr[Y(e,t) \leq 1 + D^{-\frac{1}{32}}\ \forall e,t] > 0$

Proof

- Pick level-0 waiting times $\boldsymbol{\alpha} \sim [D]^n$.
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] = \text{ave. load on } e \text{ at } t \text{ dep. on } \boldsymbol{\alpha}$

Lemma

$\Pr[Y(e,t) \leq 1 + D^{-\frac{1}{32}}\ \forall e,t] > 0$

- $\mathbb{E}[Y(e,t)] \leq 1$

Proof

- Pick level-0 waiting times $\boldsymbol{\alpha} \sim [D]^n$.
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] = \text{ave. load on } e \text{ at } t \text{ dep. on } \boldsymbol{\alpha}$

Lemma

$\Pr[Y(e,t) \leq 1 + D^{-\frac{1}{32}} \; \forall e,t] > 0$

- $\mathbb{E}[Y(e,t)] \leq 1$
- $Y(e,t) = \sum_{i=1}^{n} \underbrace{\Pr[i \text{ crosses } e \text{ at } t \mid \boldsymbol{\alpha}]}_{\in [0, \frac{1}{(\sqrt{D})^{1/4}}]}$

Proof

- Pick level-0 waiting times $\boldsymbol{\alpha} \sim [D]^n$.
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] = \text{ave. load on } e \text{ at } t \text{ dep. on } \boldsymbol{\alpha}$

Lemma

$\Pr[Y(e,t) \leq 1 + D^{-\frac{1}{32}} \ \forall e,t] > 0$

- $\mathbb{E}[Y(e,t)] \leq 1$
- $Y(e,t) = \sum_{i=1}^{n} \underbrace{\Pr[i \text{ crosses } e \text{ at } t \mid \boldsymbol{\alpha}]}_{\in [0, \frac{1}{(\sqrt{D})^{1/4}}]}$
- $\Pr[Y(e,t) > 1 + D^{-\frac{1}{32}}] \leq e^{-\Omega(D^{1/16})}$

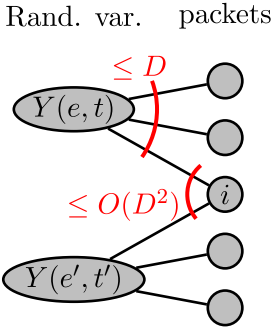
Proof

- Pick level-0 waiting times $\boldsymbol{\alpha} \sim [D]^n$.
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] =$ ave. load on e at t dep. on $\boldsymbol{\alpha}$

Lemma

$\Pr[Y(e,t) \leq 1 + D^{-\frac{1}{32}}\ \forall e,t] > 0$

- $\mathbb{E}[Y(e,t)] \leq 1$
- $Y(e,t) = \sum_{i=1}^{n} \underbrace{\Pr[i \text{ crosses } e \text{ at } t \mid \boldsymbol{\alpha}]}_{\in[0,\frac{1}{(\sqrt{D})^{1/4}}]}$
- $\Pr[Y(e,t) > 1 + D^{-\frac{1}{32}}] \leq e^{-\Omega(D^{1/16})}$



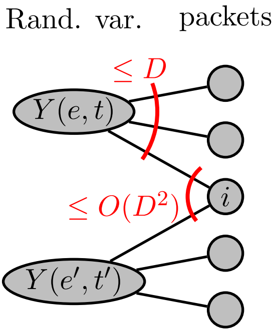
Proof

- Pick level-0 waiting times $\boldsymbol{\alpha} \sim [D]^n$.
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] =$ ave. load on e at t dep. on $\boldsymbol{\alpha}$

Lemma

$\Pr[Y(e,t) \leq 1 + D^{-\frac{1}{32}} \ \forall e,t] > 0$

- $\mathbb{E}[Y(e,t)] \leq 1$
- $Y(e,t) = \sum_{i=1}^{n} \underbrace{\Pr[i \text{ crosses } e \text{ at } t \mid \boldsymbol{\alpha}]}_{\in [0, \frac{1}{(\sqrt{D})^{1/4}}]}$
- $\Pr[Y(e,t) > 1 + D^{-\frac{1}{32}}] \leq e^{-\Omega(D^{1/16})}$
- Dependence degree $\leq O(D^3)$



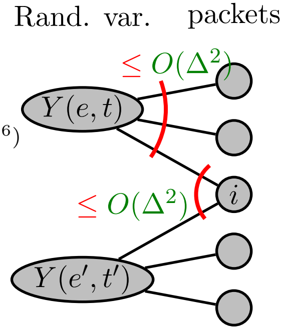
Proof

- Suppose waiting times on level $0, \ldots, \ell - 1$ already fixed.
- Pick level-ℓ waiting times $\boldsymbol{\alpha} \sim [\Delta^{1/4}]^{n \times \frac{D}{\Delta}}$. $\Delta := D_\ell$
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] =$ ave. load on e at t dep. on $\boldsymbol{\alpha}$

Lemma

$\Pr[Y(e,t) \leq \mathbb{E}[Y(e,t)] + \Delta^{-\frac{1}{32}} \; \forall e,t] > 0$

- $\mathbb{E}[Y(e,t)] \leq 1 + o(1)$
- $Y(e,t) = \sum_{i=1}^{n} \underbrace{\Pr[i \text{ crosses } e \text{ at } t \mid \boldsymbol{\alpha}]}_{\in [0, \frac{1}{(\sqrt{\Delta})^{1/4}}]}$
- $\Pr[Y(e,t) > \mathbb{E}[Y(e,t)] + \Delta^{-\frac{1}{32}}] \leq e^{-\Omega(\Delta^{1/16})}$
- If nonzero, $\Pr[i \text{ crosses } e \text{ at } t] \geq \prod_{\ell' \geq \ell} \frac{1}{D_{\ell'}^{1/4}} \geq \frac{1}{\Delta^2}$
- Possible positions & time frame $\leq O(\Delta)$
- Dependence degree $\leq O(\Delta^4)$



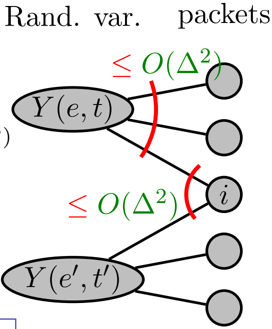
Proof

- Suppose waiting times on level $0, \dots, \ell-1$ already fixed.
- Pick level-ℓ waiting times $\boldsymbol{\alpha} \sim [\Delta^{1/4}]^{n \times \frac{D}{\Delta}}$. $\Delta := D_\ell$
- $Y(e,t) := \mathbb{E}[\text{load}(e,t) \mid \boldsymbol{\alpha}] =$ ave. load on e at t dep. on $\boldsymbol{\alpha}$

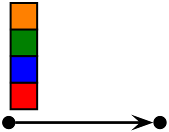
Lemma

$\Pr[Y(e,t) \leq \mathbb{E}[Y(e,t)] + \Delta^{-\frac{1}{32}} \; \forall e,t] > 0$

- $\mathbb{E}[Y(e,t)] \leq 1 + o(1)$
- $Y(e,t) = \sum_{i=1}^{n} \underbrace{\Pr[i \text{ crosses } e \text{ at } t \mid \boldsymbol{\alpha}]}_{\in[0,\frac{1}{(\sqrt{\Delta})^{1/4}}]}$
- $\Pr[Y(e,t) > \mathbb{E}[Y(e,t)] + \Delta^{-\frac{1}{32}}] \leq e^{-\Omega(\Delta^{1/16})}$
- If nonzero,
  $\Pr[i \text{ crosses } e \text{ at } t] \geq \prod_{\ell' \geq \ell} \frac{1}{D_{\ell'}^{1/4}} \geq \frac{1}{\Delta^2}$
- Possible positions & time frame $\leq O(\Delta)$
- Dependence degree $\leq O(\Delta^4)$ □



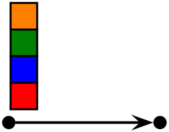
$O(1)$-size edge buffers



$O(1)$-size edge buffers

Theorem

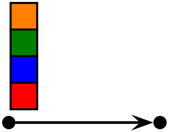
$\exists$ $O(\text{congestion} + \text{dilation})$-time, $O(1)$-load schedule where packets wait $\{0, 1\}$ time units per node.

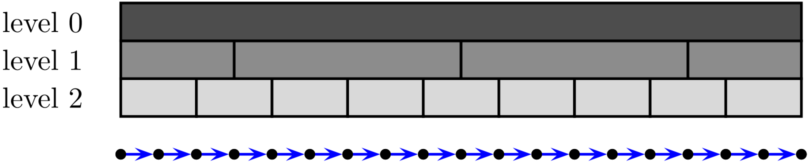


$O(1)$-size edge buffers

Theorem

$\exists$ O(congestion + dilation)-time, $O(1)$-load schedule where packets wait $\{0, 1\}$ time units per node.



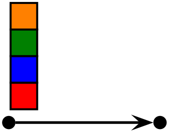


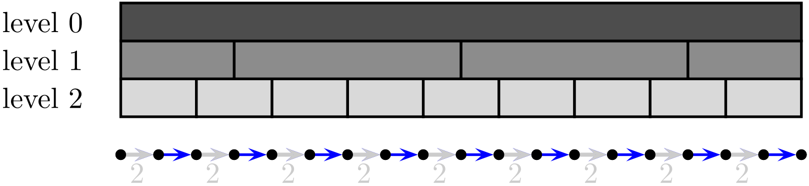
- Assign path edges to intervals s.t. level ℓ interval gets $D_\ell^{1/4}$ edges

$O(1)$-size edge buffers

Theorem

$\exists$ O(congestion + dilation)-time, $O(1)$-load schedule where packets wait $\{0, 1\}$ time units per node.



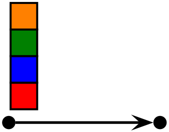


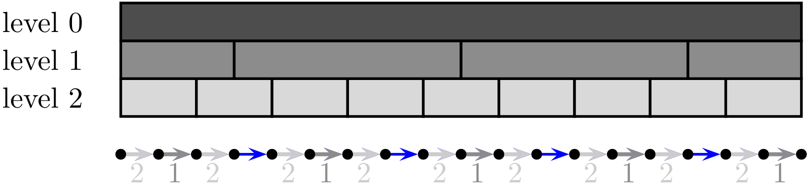
- Assign path edges to intervals s.t. level ℓ interval gets $D_\ell^{1/4}$ edges

$O(1)$-size edge buffers

Theorem

$\exists$ $O(\text{congestion} + \text{dilation})$-time, $O(1)$-load schedule where packets wait $\{0, 1\}$ time units per node.



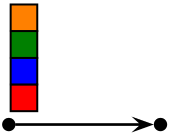


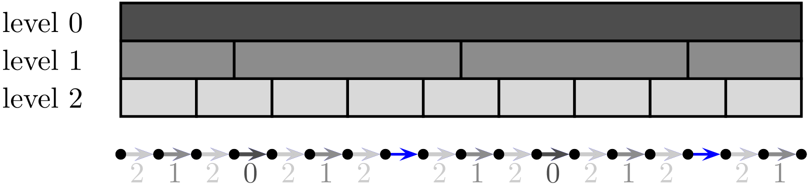
- Assign path edges to intervals s.t. level ℓ interval gets $D_\ell^{1/4}$ edges

$O(1)$-size edge buffers

Theorem

$\exists\ O(\text{congestion} + \text{dilation})$-time, $O(1)$-load schedule where packets wait $\{0, 1\}$ time units per node.



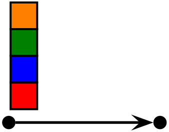


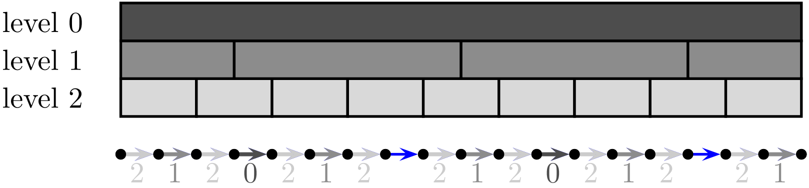
- Assign path edges to intervals s.t. level ℓ interval gets $D_\ell^{1/4}$ edges

$O(1)$-size edge buffers

Theorem

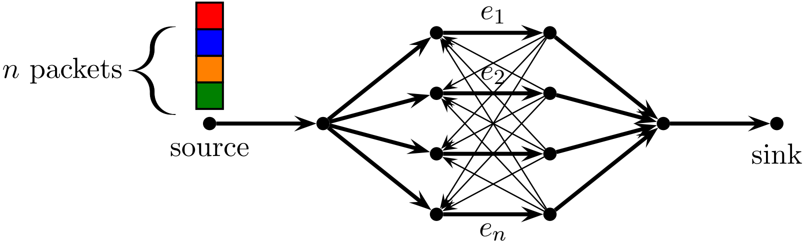
$\exists$ $O(\text{congestion} + \text{dilation})$-time, $O(1)$-load schedule where packets wait $\{0, 1\}$ time units per node.



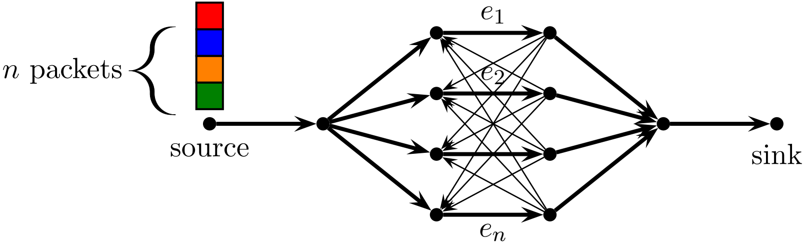


- Assign path edges to intervals s.t. level ℓ interval gets $D_\ell^{1/4}$ edges
- Wait on first $\alpha \sim [D_\ell^{1/4}]$ assigned edges

A lower bound construction

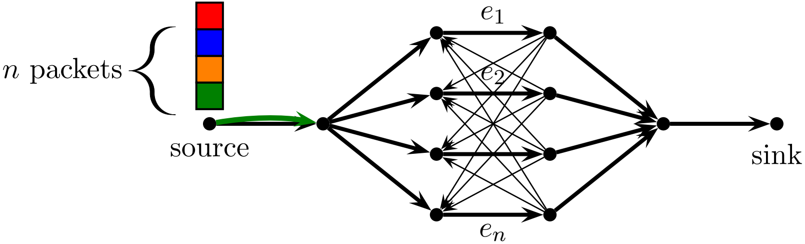


A lower bound construction



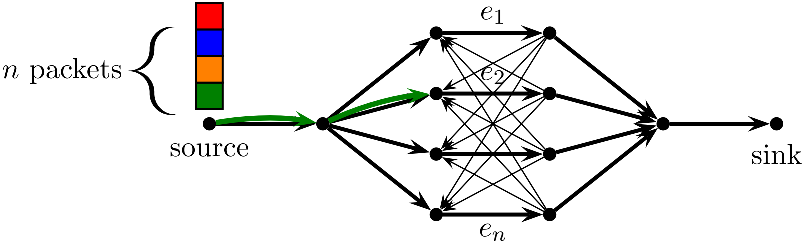
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



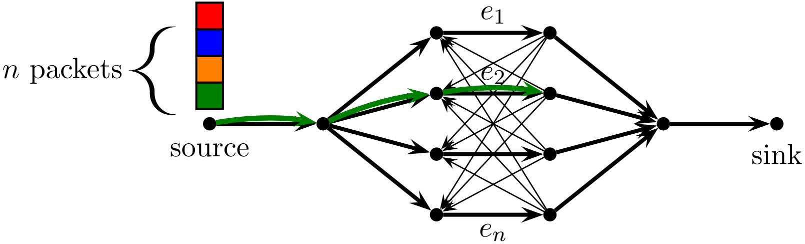
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



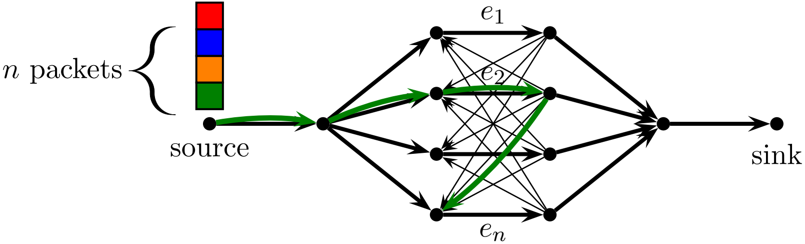
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



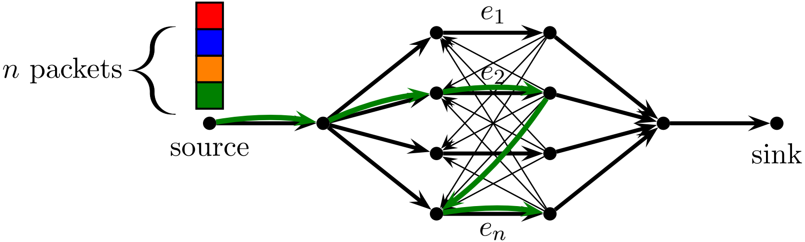
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



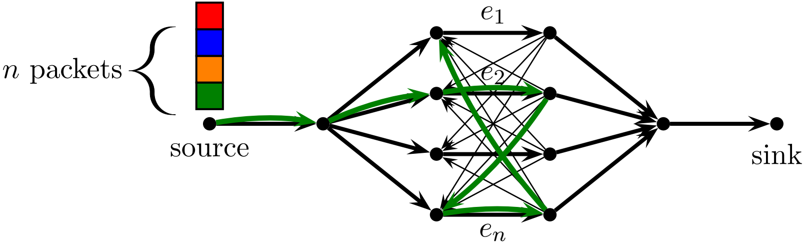
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



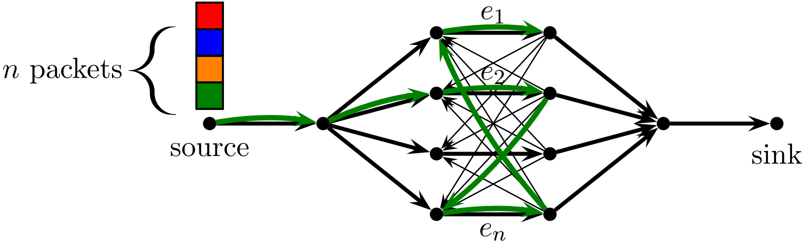
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



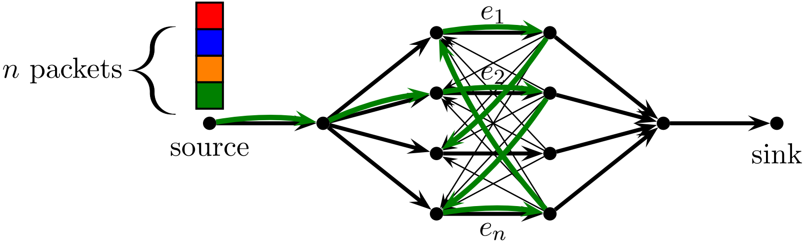
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



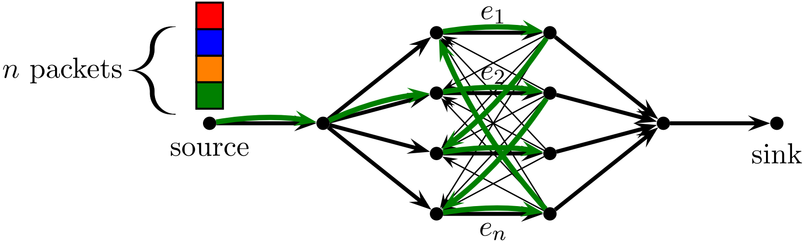
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



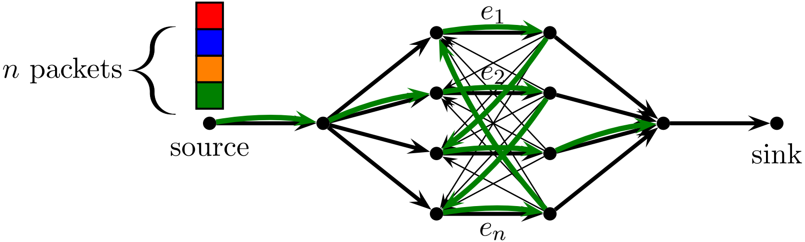
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



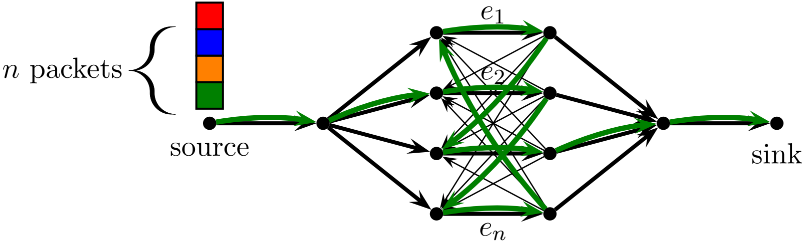
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



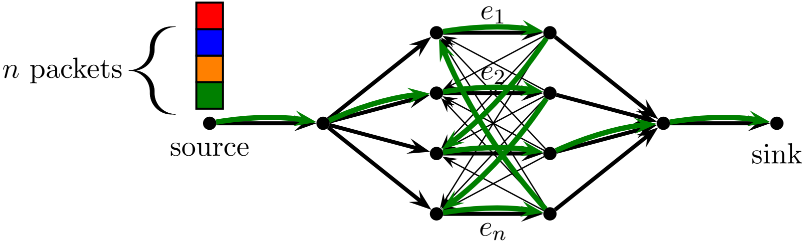
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction



- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

A lower bound construction

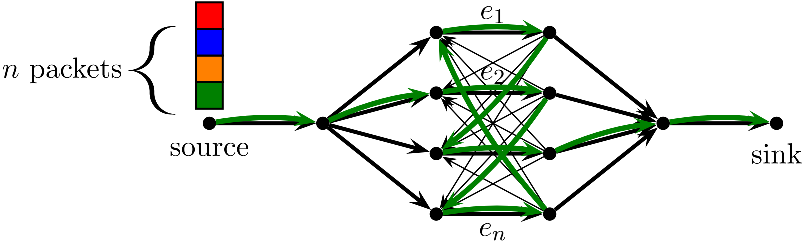


- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n

A lower bound construction

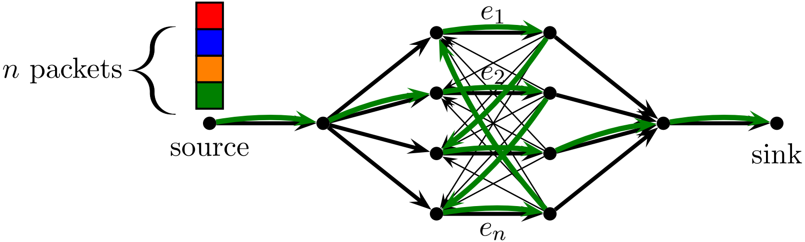


- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$

A lower bound construction

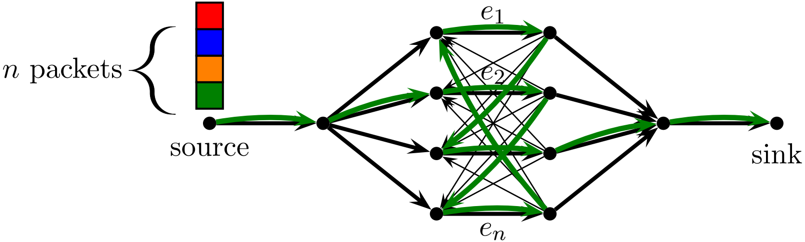


- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n+3$
- makespan $\geq 3n$

A lower bound construction

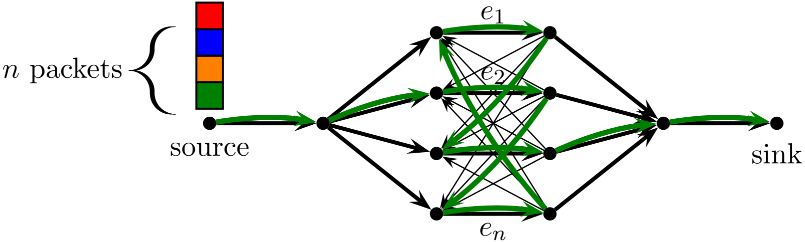


- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$

A lower bound construction

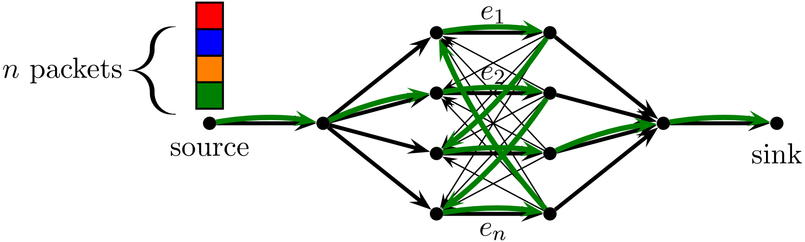


- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$
- **Goal:** # schedules $\cdot$ Pr[fixed schedule collision free] $\ll 1$

A lower bound construction



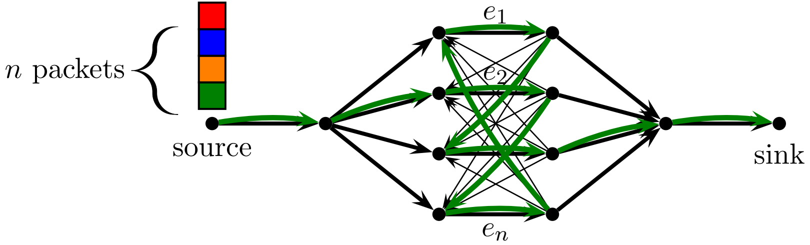
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$
- **Goal:** # schedules $\cdot$ Pr[fixed schedule collision free] $\ll 1$
- A routing strategy for a single packet is of the form

(park, park, go, wait, go, go, wait, wait, go, go, park)

A lower bound construction



- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

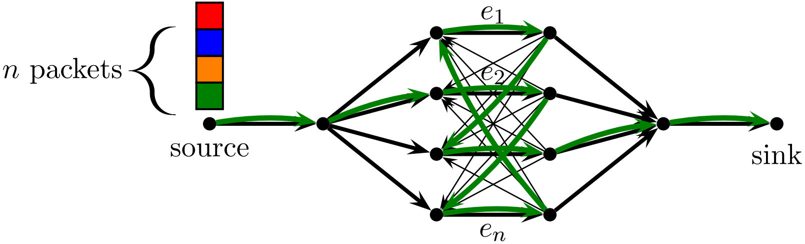
Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$
- **Goal:** # schedules $\cdot$ Pr[fixed schedule collision free] $\ll 1$
- A routing strategy for a single packet is of the form

(park, park, go, wait, go, go, wait, wait, go, go, park)

time horizon $(3 + \varepsilon)n$

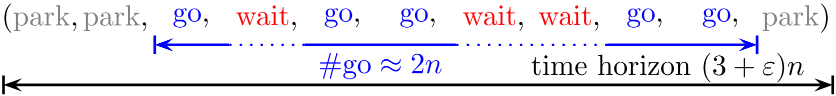
A lower bound construction



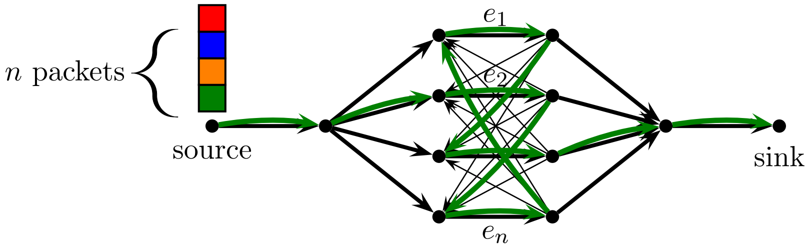
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$
- **Goal:** # schedules $\cdot$ Pr[fixed schedule collision free] $\ll 1$
- A routing strategy for a single packet is of the form



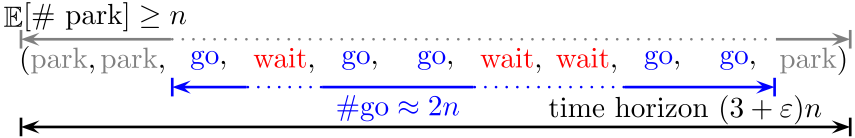
A lower bound construction



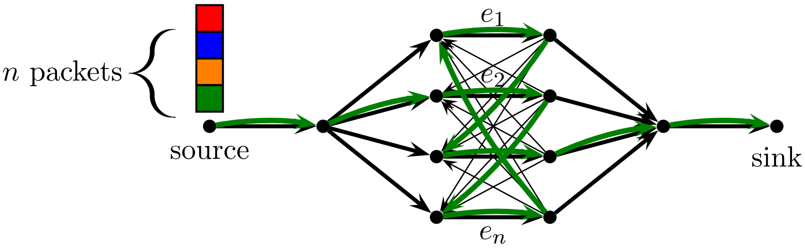
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$
- **Goal:** # schedules $\cdot$ Pr[fixed schedule collision free] $\ll 1$
- A routing strategy for a single packet is of the form



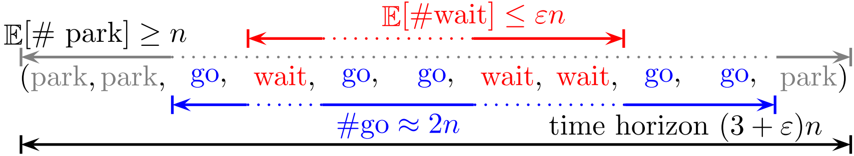
A lower bound construction



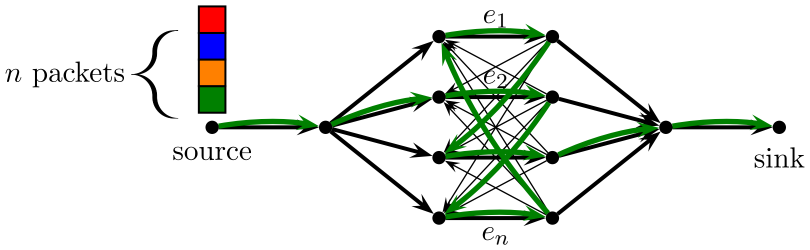
- Choose $P_1, \ldots, P_n$: go through $e_1, \ldots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$
- **Goal:** # schedules $\cdot$ Pr[fixed schedule collision free] $\ll 1$
- A routing strategy for a single packet is of the form



A lower bound construction



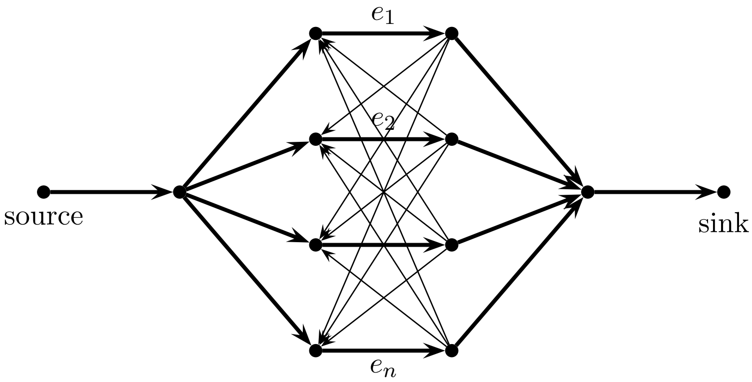
- Choose $P_1, \dots, P_n$: go through $e_1, \dots, e_n$ in random order

Observations:

- Congestion n
- Dilation $2n + 3$
- makespan $\geq 3n$
- Suppose makespan $\leq (3 + \varepsilon)n$
- **Goal:** # schedules $\cdot$ Pr[fixed schedule collision free] $\ll 1$
- total # routing strategies $\leq (2n \cdot \binom{4n}{\varepsilon n})^n \leq 2^{o(n^2)}$ for $\varepsilon \to 0$

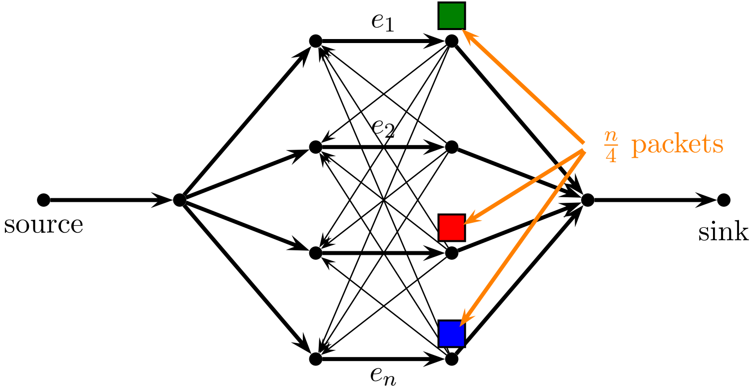
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



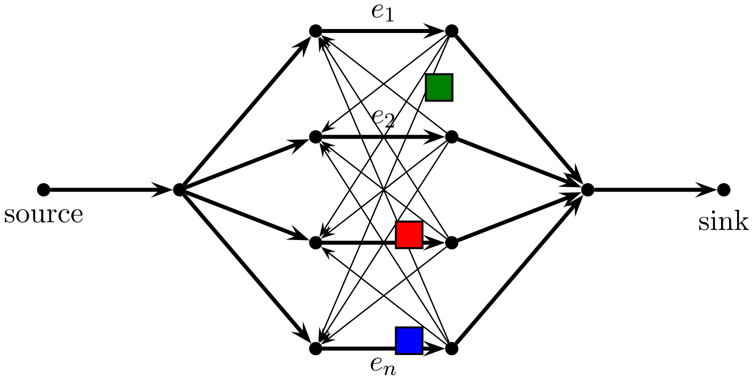
Lemma

Fix a schedule. $\Pr[\text{schedule feasible}] \leq (\frac{1}{2})^{\Theta(n^2)}$



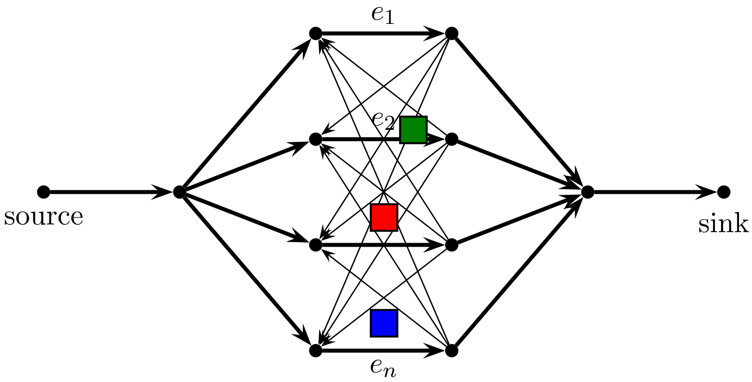
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



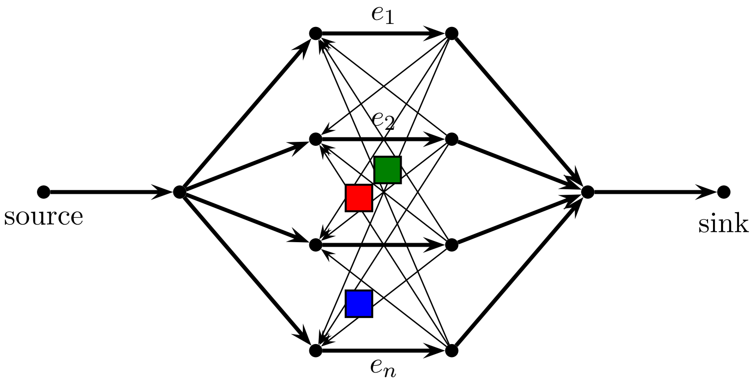
Lemma

Fix a schedule. $\Pr[\text{schedule feasible}] \leq (\frac{1}{2})^{\Theta(n^2)}$



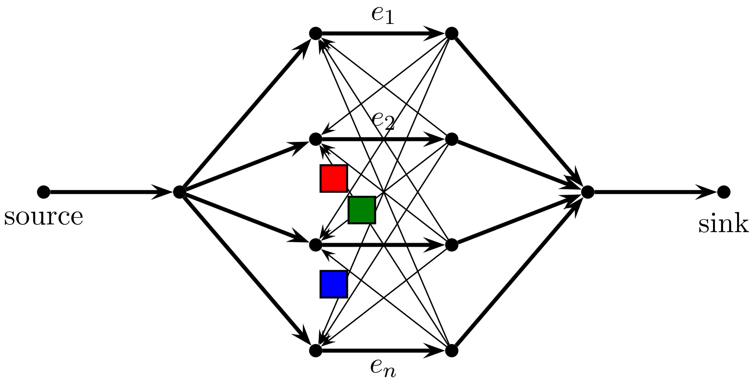
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



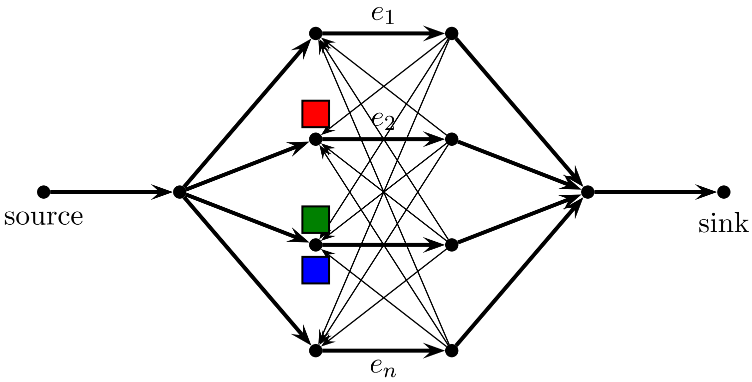
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



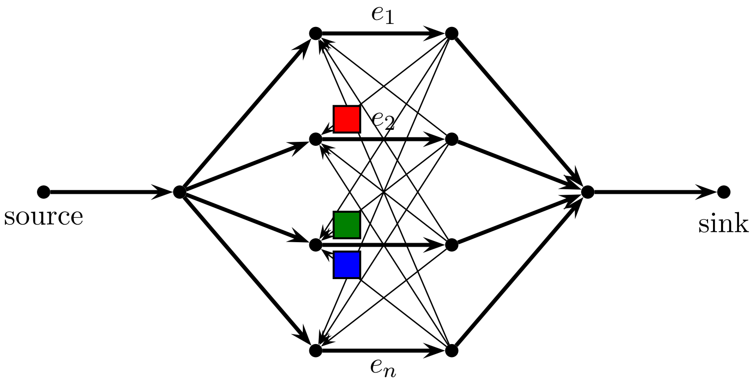
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



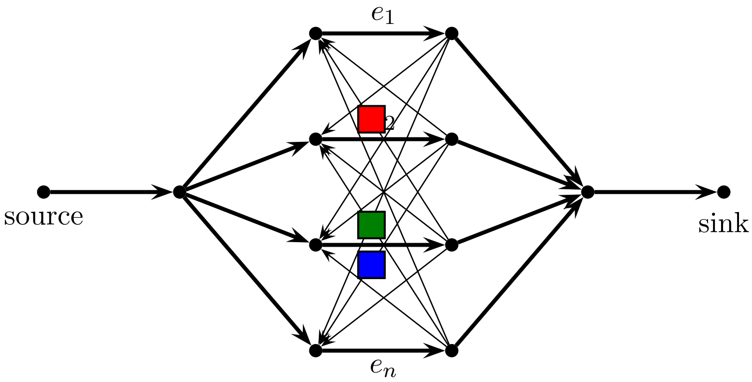
Lemma

Fix a schedule. $\Pr[\text{schedule feasible}] \leq (\frac{1}{2})^{\Theta(n^2)}$



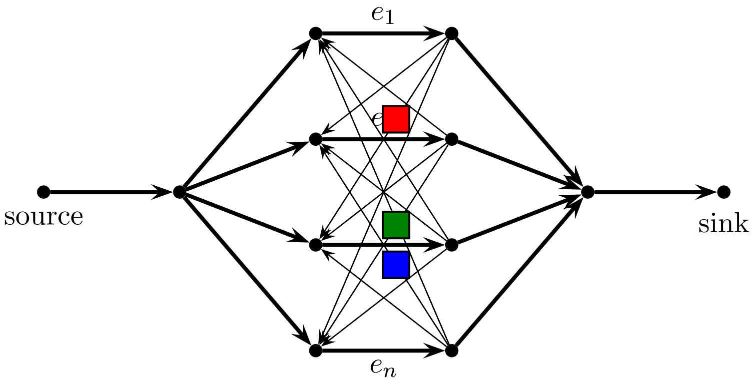
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



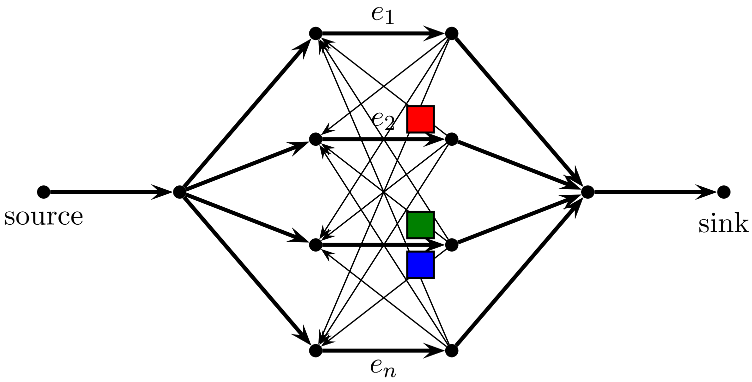
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



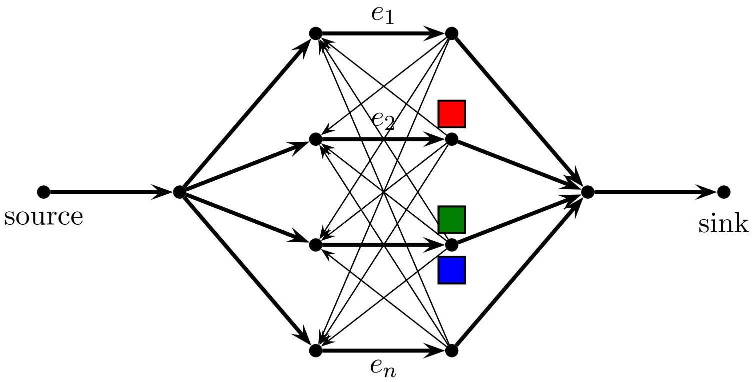
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



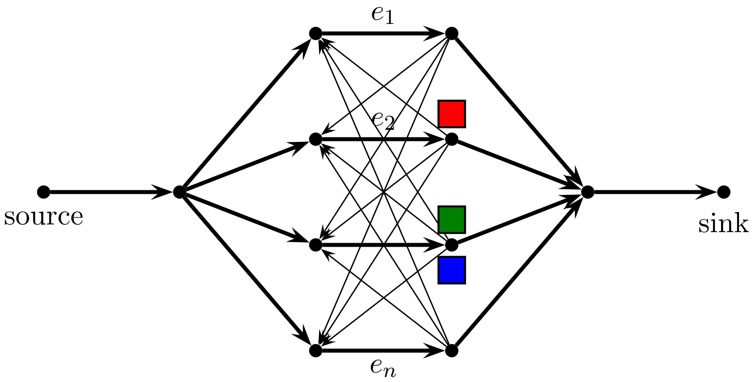
Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



Lemma

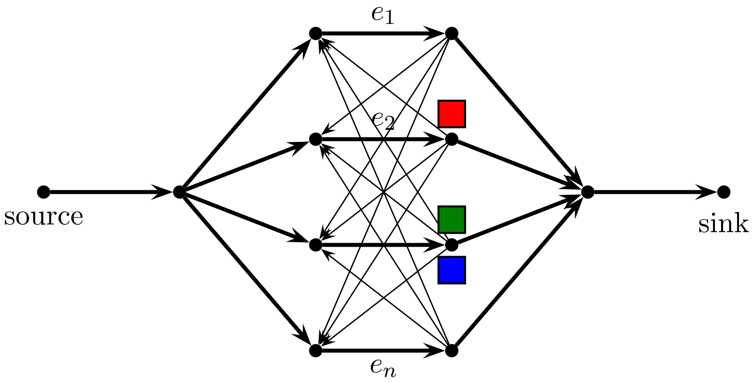
Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



- Pr[no collision] $\leq (\frac{1}{8})^{n/8}$

Lemma

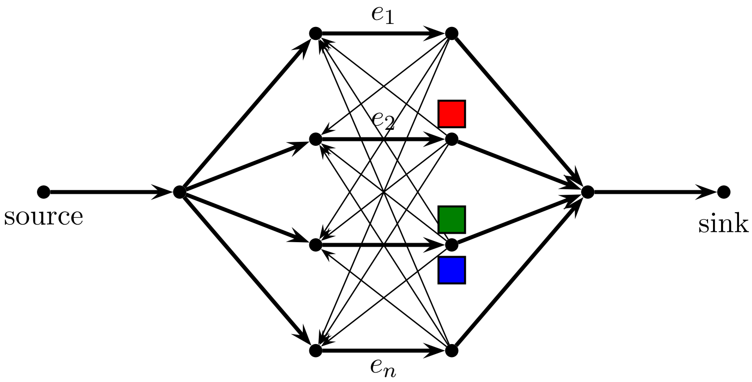
Fix a schedule. $\Pr[\text{schedule feasible}] \leq (\frac{1}{2})^{\Theta(n^2)}$



- $\Pr[\text{no collision}] \leq (\frac{1}{8})^{n/8}$
- $\exists \frac{n}{16}$ time steps in which $\frac{n}{4}$ packets cross a random edge e_j.

Lemma

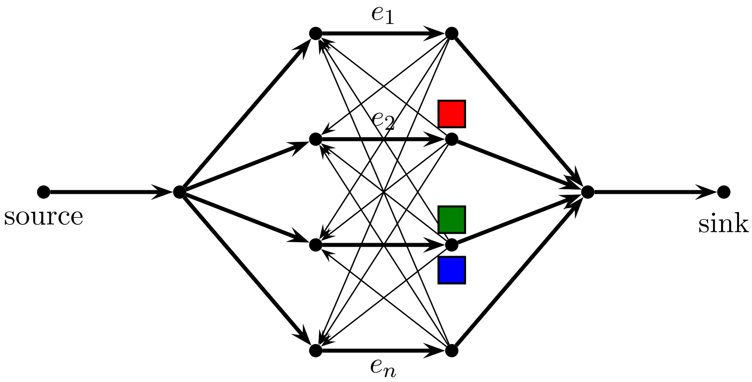
Fix a schedule. $\Pr[\text{schedule feasible}] \leq (\frac{1}{2})^{\Theta(n^2)}$



- $\Pr[\text{no collision}] \leq (\frac{1}{8})^{n/8}$
- $\exists \frac{n}{16}$ time steps in which $\frac{n}{4}$ packets cross a random edge e_j.
- **Problem:** Steps not independent!

Lemma

Fix a schedule. Pr[schedule feasible] $\leq (\frac{1}{2})^{\Theta(n^2)}$



- Pr[no collision] $\leq (\frac{1}{8})^{n/8}$
- $\exists \frac{n}{16}$ time steps in which $\frac{n}{4}$ packets cross a random edge e_j.
- **Problem:** Steps not independent! But more careful analysis works. □

The end

Open question

Can **acyclic job shop with preemption** be done in $O(\text{congestion} + \text{dilation})$?

- $O((C + D) \cdot \log\log(C + D))$ suffices [Feige, Scheideler '02]

The end

Open question

Can **acyclic job shop with preemption** be done in $O(\text{congestion} + \text{dilation})$?

- $O((C + D) \cdot \log\log(C + D))$ suffices [Feige, Scheideler '02]

Thanks for your attention

Acyclic job shop with preemption

Given:

- Directed (simple) paths P_i
- Processing times $p_{i,e}\ \forall i \in [n]\ \forall e \in P_i$

Constraints:

- Packet i takes time $p_{i,e}$ to cross $e \in P_i$
- At most one packet can actively move on an edge per time unit
- **Preemption:** Packet can "stop" in the middle of an edge (and another packet can be processed)

Parameters:

- Congestion $C := \max_{e \in E}\{\sum_{i: e \in P_i} p_{i,e}\}$
- Dilation $D := \max_i\{\sum_{e \in P_i} p_{i,e}\}$
- $L := \max\{C, D\}$

Question: Is $O(L)$ possible? (Known: $O(L \cdot \log\log L)$)