How Bad is Selfish Traffic Routing

Malik Magdon-Ismail
magdon@cs.rpi.edu

(Joint Work with **Costas Busch**)

July 6, 2006.

Rensselaer

Disclaimer: Theoretical talk; no theorems proved.

Huge Numbers

Many people believe, King Gelon, that the grains of sand are infinite in multitude; and I mean by the sand not only that which exists around Syracuse and the rest of Sicily, but also that which is found in every region, whether inhabited or unhabited. Others think that although their number is not without limit, no number can ever be named which will be greater than the number of grains of sand. But I shall try to prove to you that among the numbers which I have named there are those which exceed the number of grains in a heap of sand the size not only of the earth, but even of the universe – **Archimedes**, The Sand Reckoner

Archimedes estimated as follows: one poppy-head, diameter $\geq$ 1/40th a finger's breadth, contains $\leq$ 10,000 grains. The sphere of the fixed stars (to Archimedes, the boundary of the universe), was less than 10^7 times the sphere with sun orbit as a great circle. Thus, $\leq 10^{63}$ grains fill the whole universe.

Archimedes had to invent notation for expressing large numbers!

Number of neutrons in universe $\leq 10^{128}$; 1 $googol = 10^{100}$; Skewes–Littlewood number $\leq 10^{371}$.

Largest useful number [G. Book of Records]: Grahams number (no succint representation).

The Practical Theorists Toolkit

$\log n =$ "constant" ≤ 128.

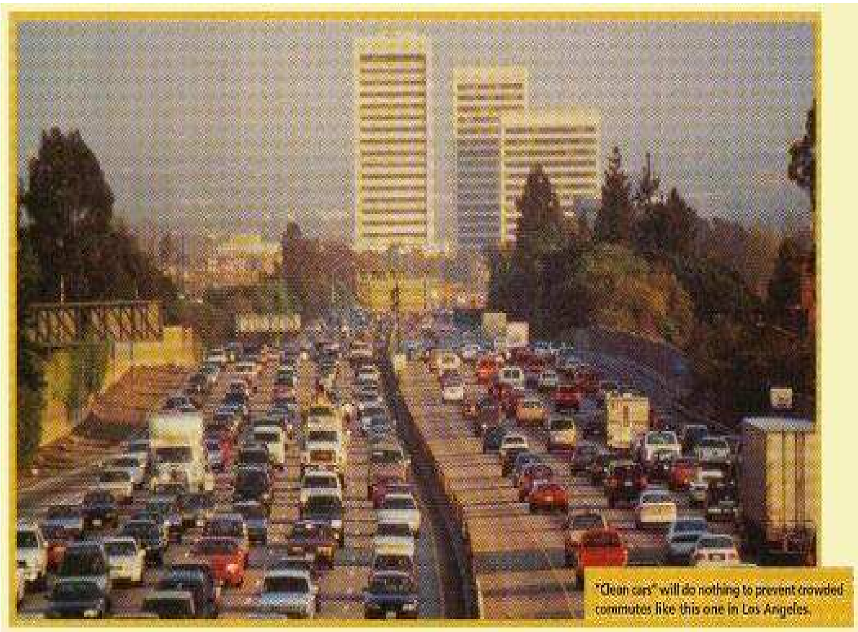
$poly(n) =$ "large" .

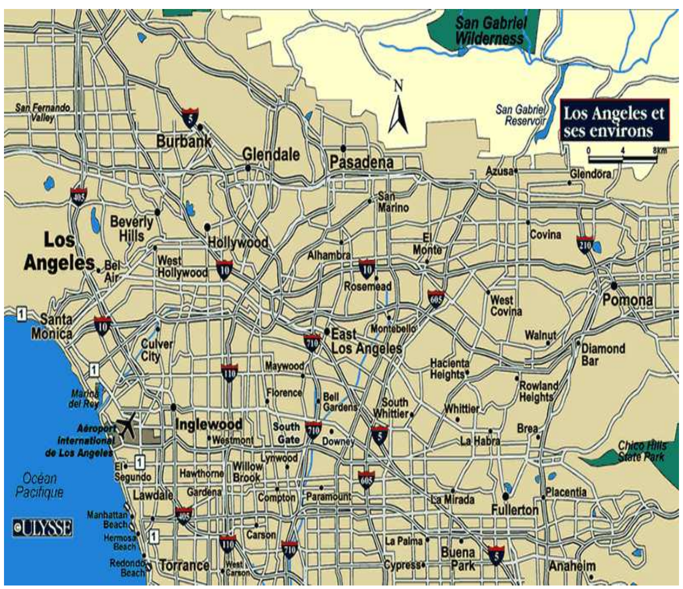
A "=" B means $(constant \cdot B) \leq A (\leq constant \cdot B)$.

Theorist is a pessimist

Pessimist: "We are living in the worst possible world"
Optimist: "The world could possibly be worse"

Congested Urban Areas





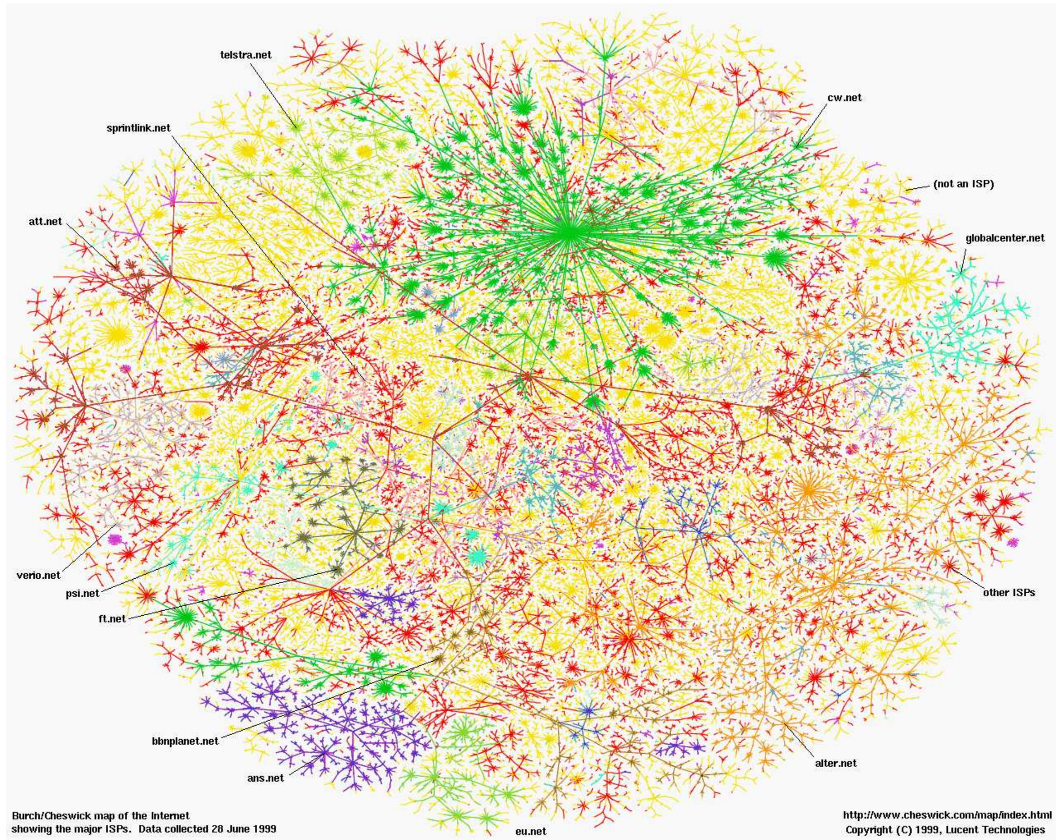
"Clean cars will do nothing to prevent crowded commutes like this one in LA"

– **What will?**

2000 census:

1.5 million commuters per day.

Congested Internet

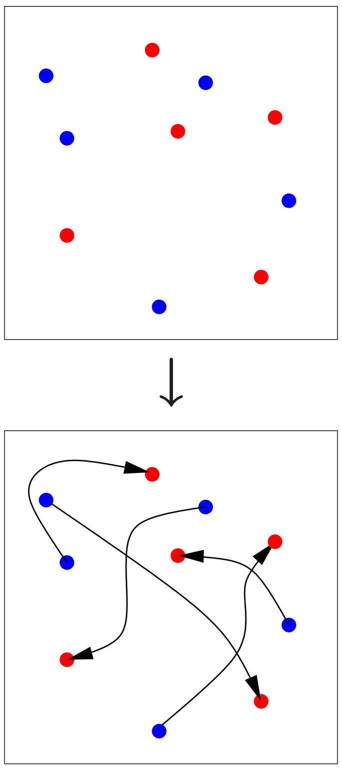


[www.research.lumeta.com]

Est. 224 million Internet users as of Oct. 2005 in N. America

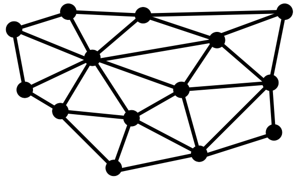
Routing

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

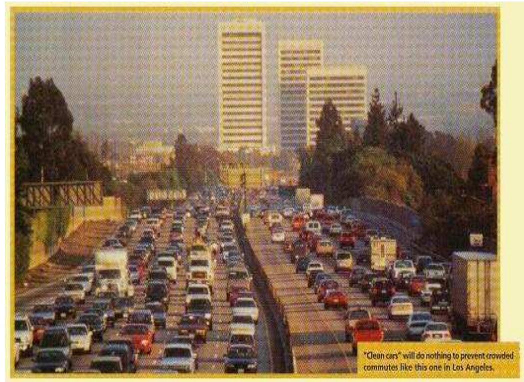


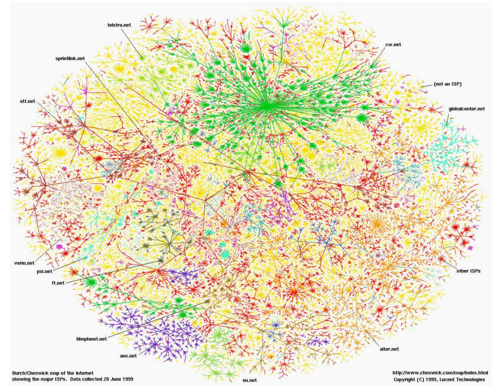
- Communication Networks – eg. **Internet**.
- City Traffic Networks.
- Ad-hoc Networks – eg. sensor networks.
- Parallel Architectures – eg. Mesh.
- . . .

sources and *destinations* are **nodes** in the **network/graph**
nodes are interconnected by **edges** (eg. roads, fibre-optics)



The Goal

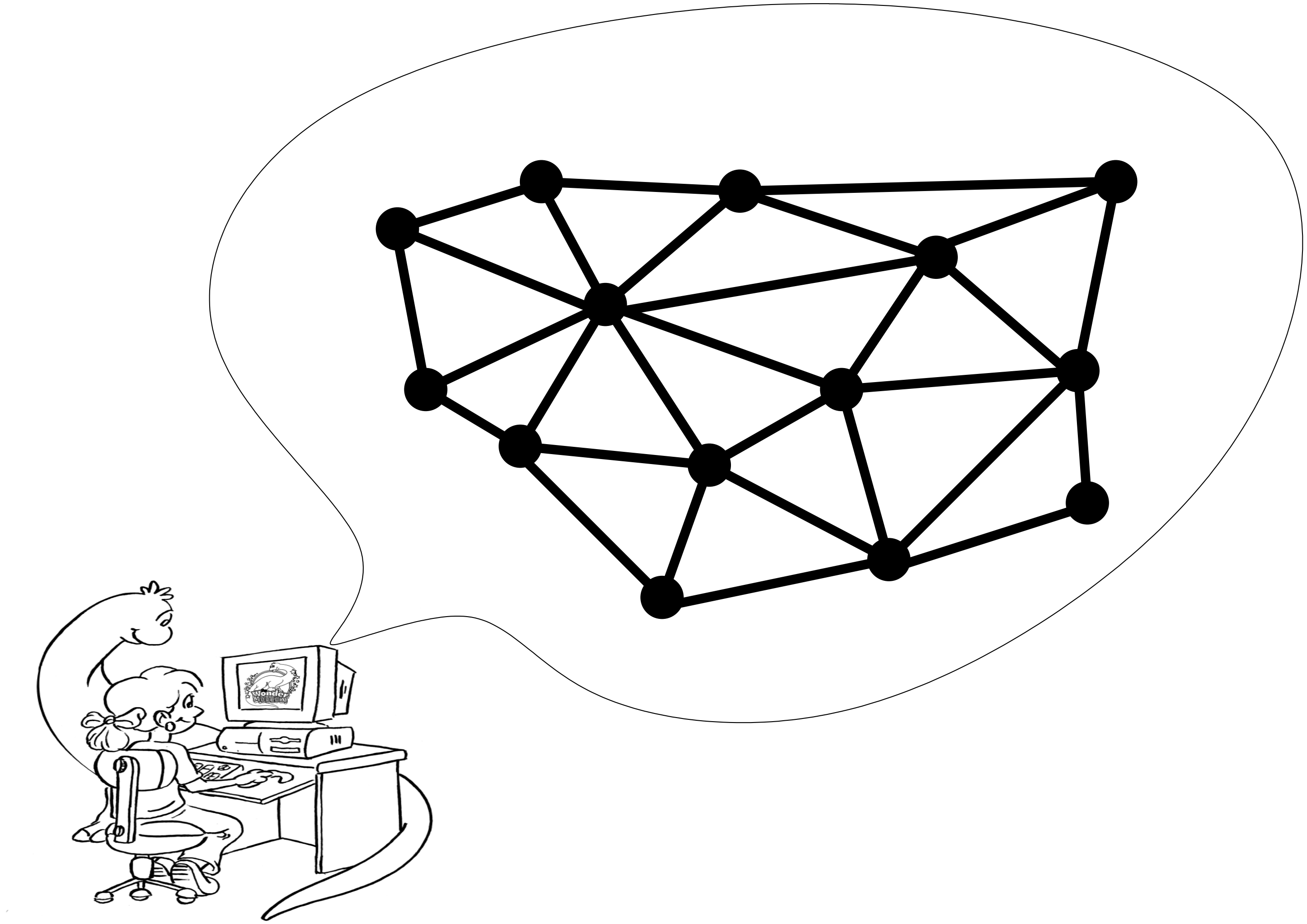




Develop "Universal" routing algorithms to avoid congestion!

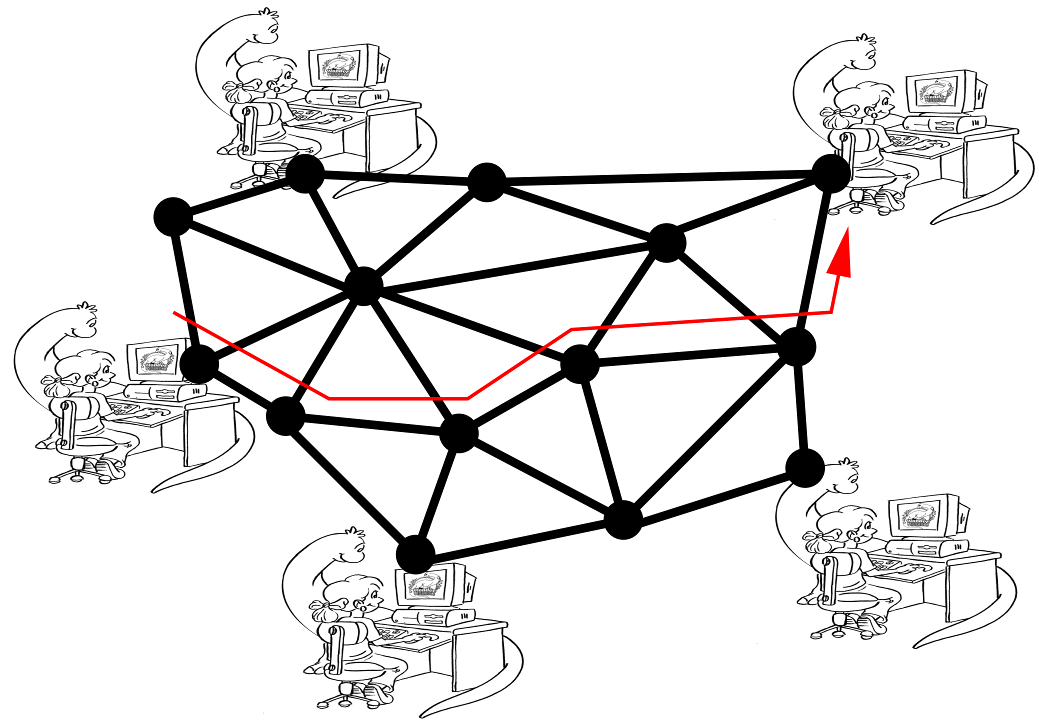
– What is universal?
– What is congestion?
– What does "avoid" mean?
– How?
– At what costs?

Centralized vs. Oblivious Routing



Centralized: optimal(?); impractical; needs traffic information.

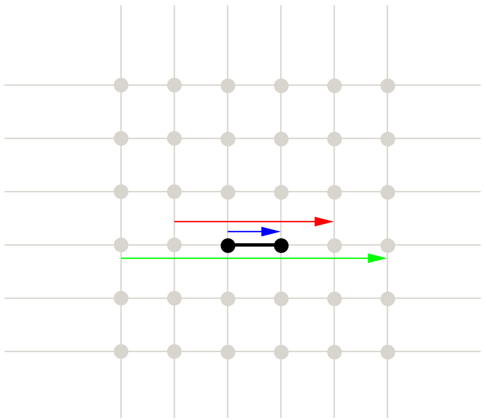
Centralized vs. Oblivious Routing



Oblivious: "packets" **independently** select paths.
– practical;
– no need for traffic information.

Optimal(**?**)

Oblivious Routing – Examples

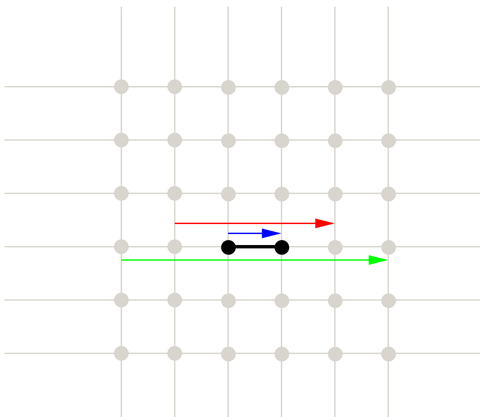


Each packet selects the shortest path.

$$Congestion = 3$$
$$Stretch = 1$$

Congestion can be **BAD**!

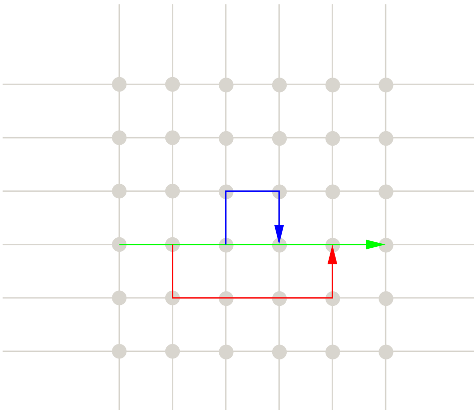
Oblivious Routing – Examples



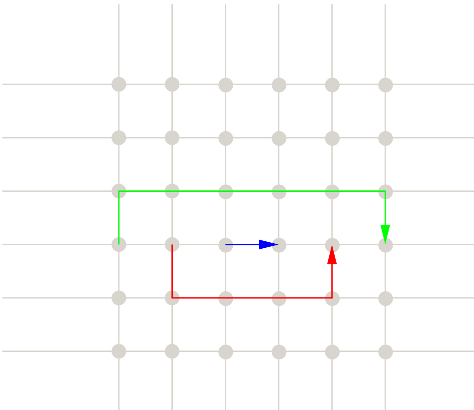
Each packet selects the shortest path.

$$Congestion = 3;$$
$$Stretch = 1.$$

Congestion can be **BAD**!



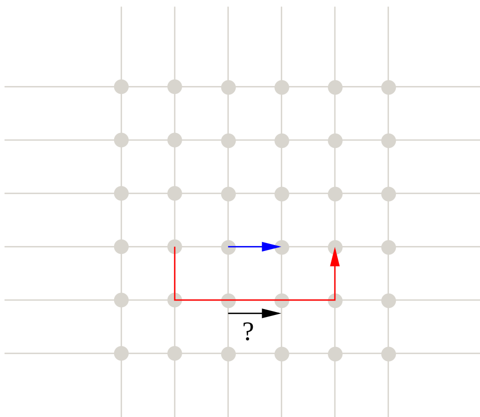
$Congestion = 1; Stretch = 3$



$Congestion = 1; Stretch = 1.66$

It is better to expand long paths than short paths – **small stretch**.

Oblivious Routing – Examples



Fixed, not necessarily shortest, paths.

Deterministic oblivious routing

Theorem [Congestion can be BAD] For *any* deterministic oblivious routing algorithm, there exists a routing instance for which this algorithm has congestion $\geq poly(n) \cdot optimal$ **[Busch, M., Xi 2005a]**

Examples:

Routing on the internet (fixed routing tables).

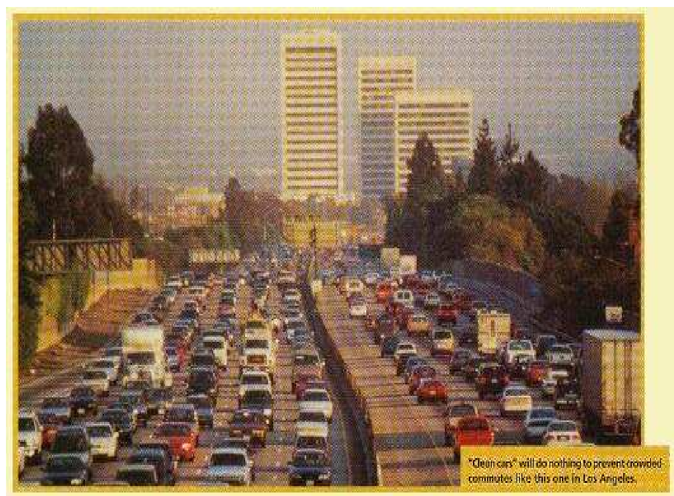
Mapquest, Google-maps, etc.

Random Inputs

Why does internet routing/ Mapquest not fail?

source-destination pairs are “somewhat” random

– sensitive to traffic distribution.



. . . or does it fail?

. . . Alternative?

Randomized Algorithms

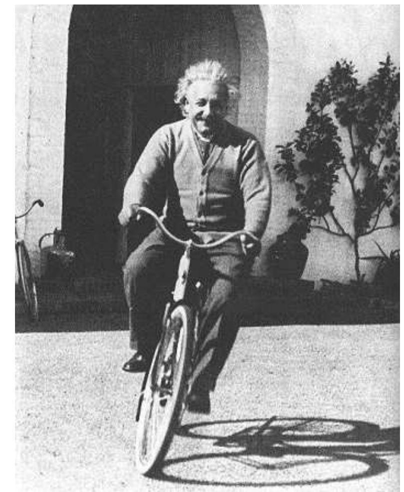
Randomized algorithms are not uncommon.

eg. QuickSort:

There are inputs which take long (quadratic time)

Random inputs: expected to be much faster

Randomized pivots: fast (w.h.p.) for *any* input.



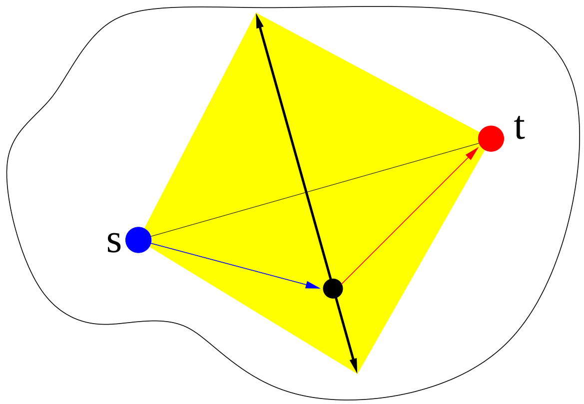
"God does not play dice with the universe" – *A. E.*

Optimal on usual inputs

vs.

Usually optimal on all inputs

Randomized Oblivious Routing



1. Select *random* intermediate stops;
2. Go to intermediate stops;
3. Then go to destination.

"God does not play dice with the universe" – *A. E.*

[Valiant 1981,1982] Hypercube networks.
[Maggs et. al. 1997] Manhattan networks, unbounded stretch.
[Räcke et. al. 2002] General networks, unbounded stretch.
[Busch, M., Xi 2005a] Manhattan networks, constant stretch.
[Busch, M., Xi 2005b] Embeddable networks, constant stretch.

Randomized Oblivious Routing

Theorem 2 [Busch, M., Xi 2005b] On embeddable networks, one can suitably choose *one* random intermediate point (obliviously) for each packet. The resulting congestion is *optimal* and the stretch is $constant$ ($\sqrt{2}$ on Manhattan style networks).

Theorem 3 [Busch, M., Xi 2005a] Significant randomization is necessary to obtain optimal congestion. For a source-destination which are distance D apart, *at least* $\log D$ random bits are required.

Optimal oblivious routing *must* play dice

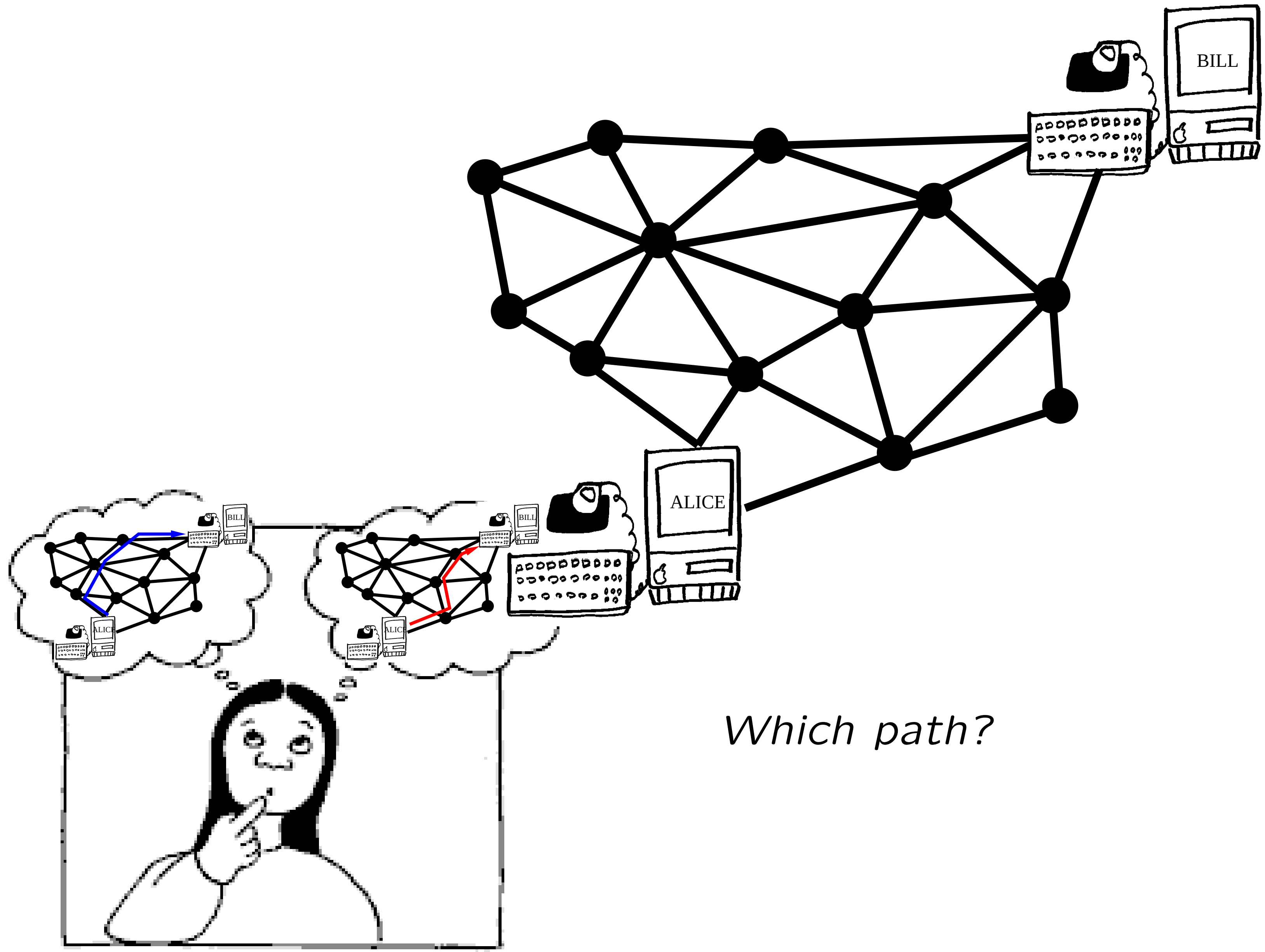
Recap

1. Optimal oblivious routing is possible.
2. Random intermediate point is *required*.
3. Can get small stretch and *optimal* congestion.

So What?

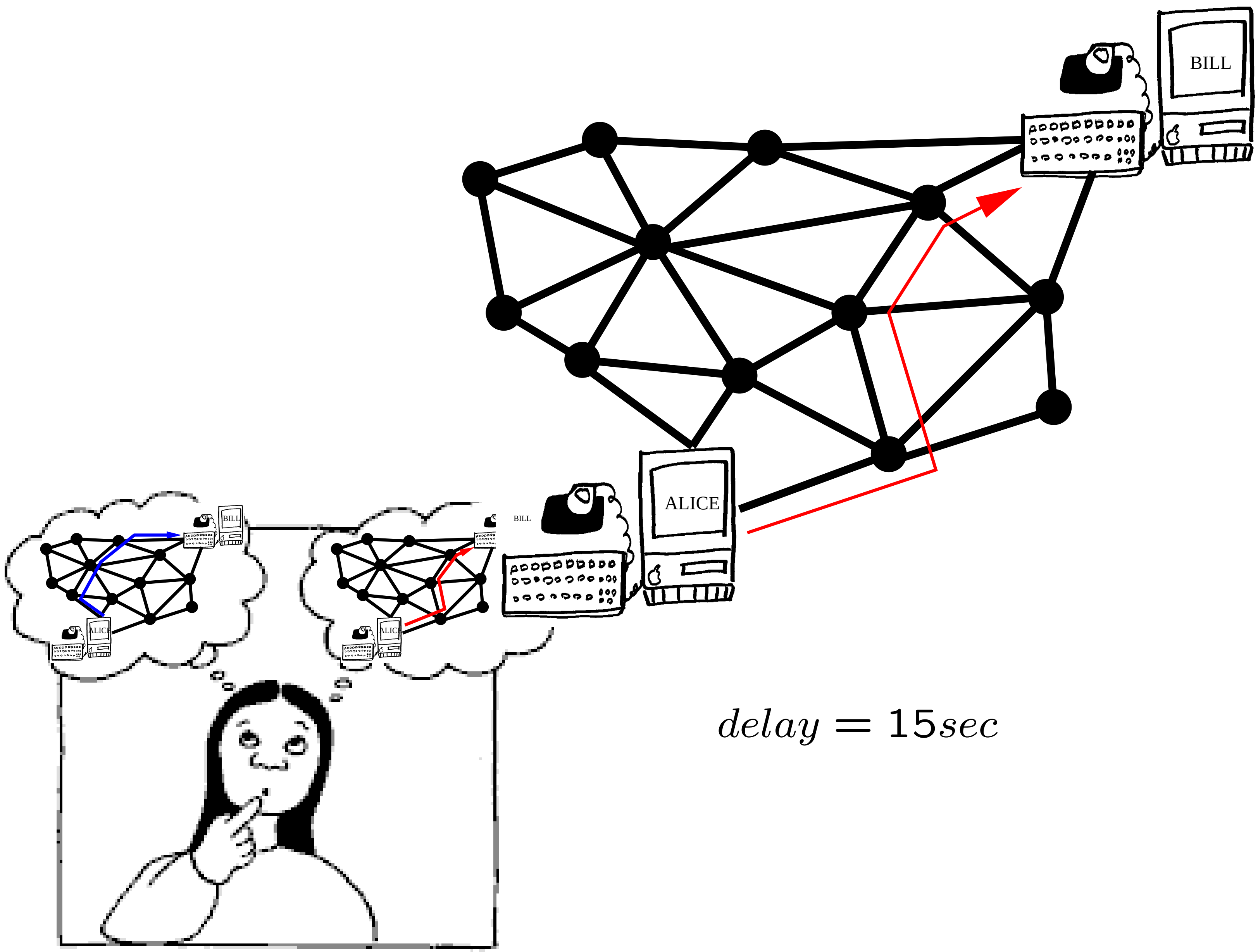
...will people follow?

Selfish Routing



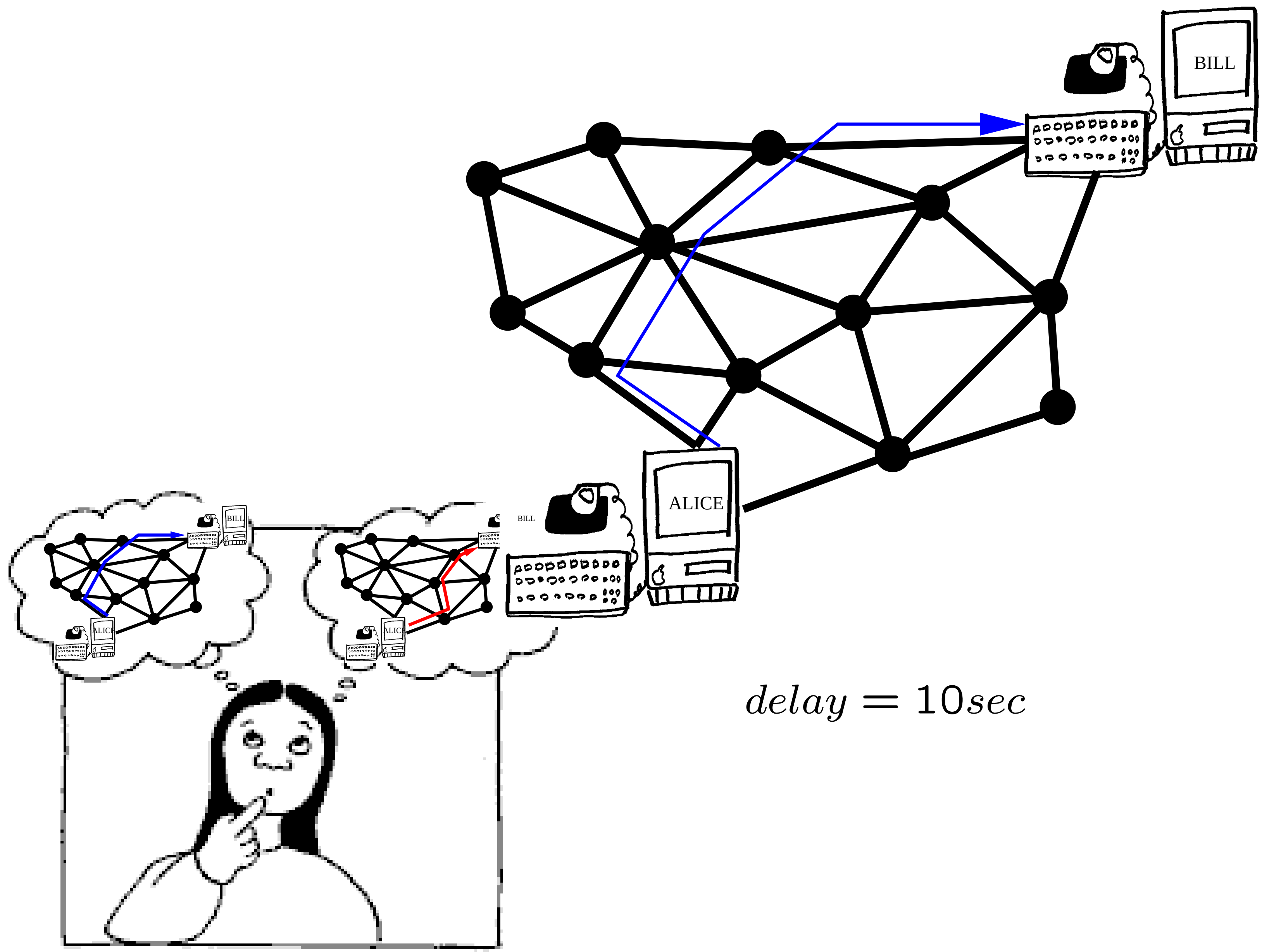
Which path?

Selfish Routing



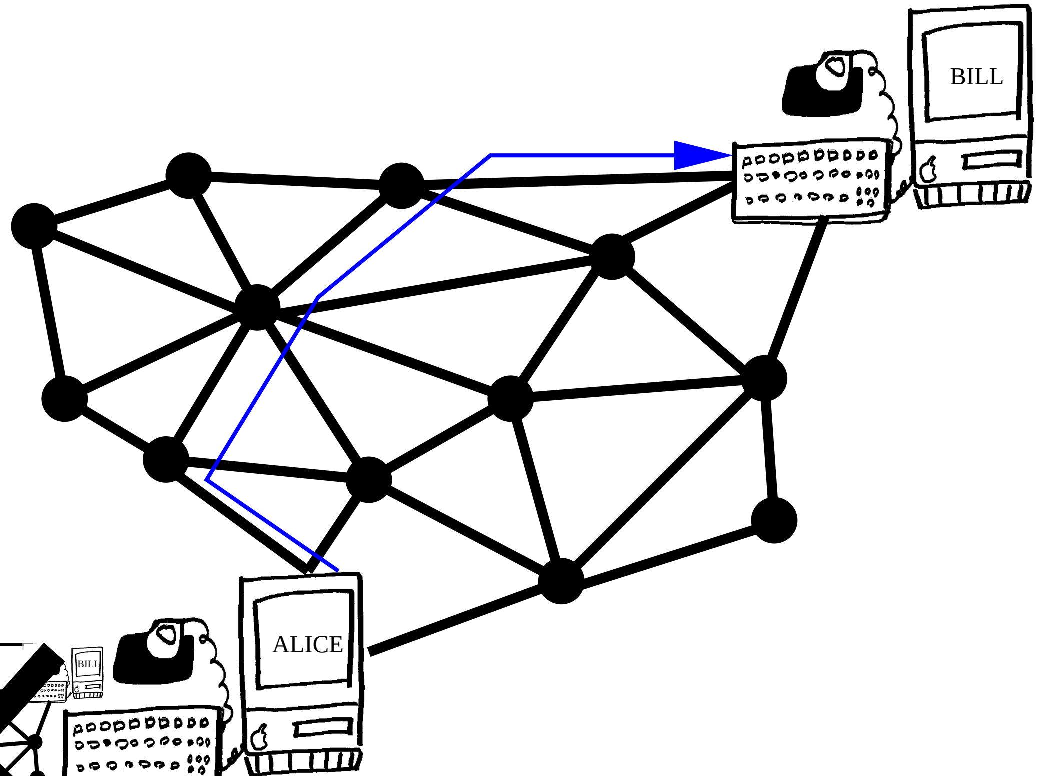
$delay = 15sec$

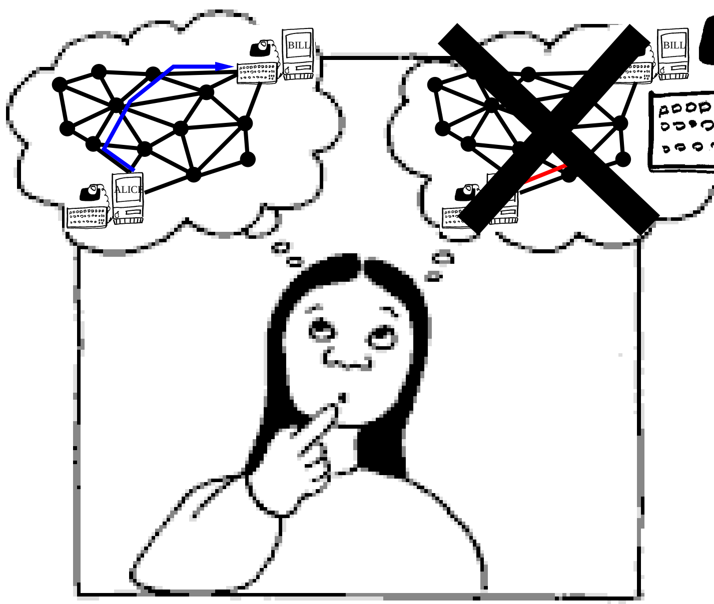
Selfish Routing



$delay = 10sec$

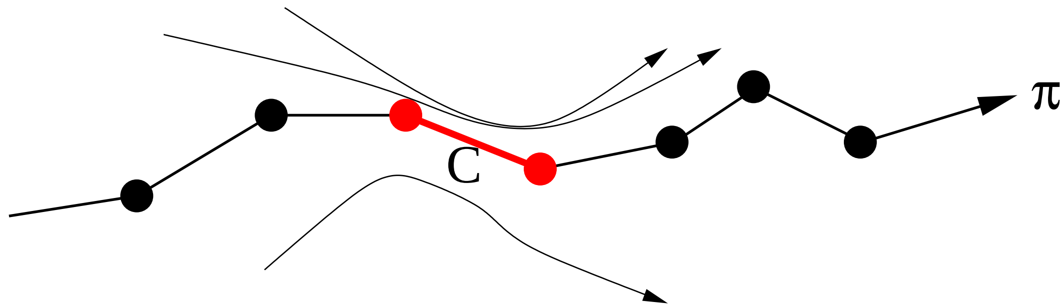
Selfish Routing





$delay = 10sec$

Delay ∼ Congestion



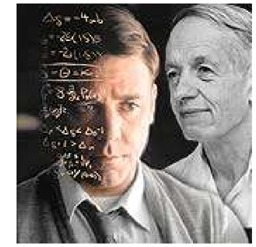
In congested networks,

$$delay(\pi) \sim C(\pi).$$

[Leighton, Maggs, Rao 1995]
[Berenbrink, Scheideler 1999]

Selfish players minimize the maximum congestion along their path.
Satisfied only if *all* alternative paths have as large congestion.

Routing Games



- **Non-cooperative selfish players**: players change paths to minimize their delay.

The Best Response Dynamic

- **Nash-Routing**: no-one wishes to change her path selection, given what everyone else is doing.

Stable States

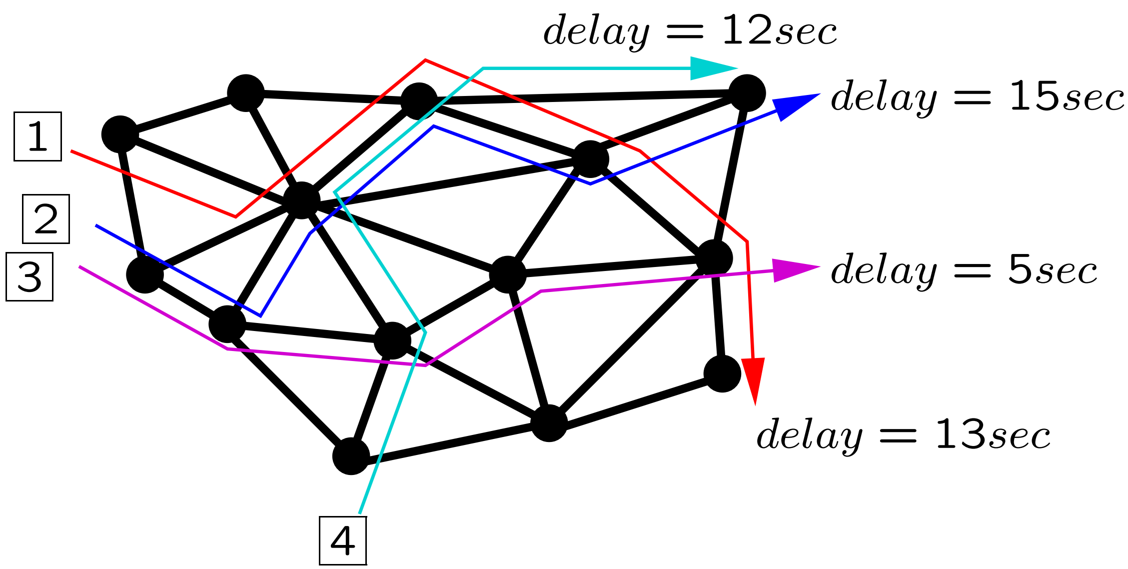
What are properties of this process?

Player vs. Social Cost

Player cost pc: her delay.

Social cost SC: maximum delay over all players.

Players minimize their player cost selfishly
Ideally, social cost should be minimized.



$SC = 15sec$

Formal Setup

Routing (Congestion) Game: $(\mathbf{N}, G, \{\mathcal{P}_i\}_{i \in \mathbf{N}})$.

$\mathbf{N} = \{1, 2, \ldots, N\}$ – players, i.e. $(source, dest)$ pairs;
$G = (V, E)$ – network;
$\mathcal{P}_i$ – strategy sets (*edge-simple* paths).

Routing: $\mathbf{p} = [p_1, p_2, \cdots, p_N]$ – *pure strategy profile*.

Congestion: $C_e(\mathbf{p}) = \#$ paths using edge e.

Path Congestion: $C_i(\mathbf{p}) = \max_{e \in p_i} C_e(\mathbf{p})$;
Network Congestion: $C(\mathbf{p}) = \max_i C_i(\mathbf{p})$;

Social Cost: $SC(\mathbf{p}) = C(\mathbf{p})$ (Network Congestion).

Player Cost: $pc_i(\mathbf{p}) = C_i(\mathbf{p})$ (Player's Path Congestion).

Nash-routing $\mathbf{p}$: $pc_i(\mathbf{p}) \leq pc_i(\mathbf{p}')$ ($\mathbf{p}'$ differs from $\mathbf{p}$ only in p_i).
(No one can unilaterally inprove her situation in a Nash-routing.)

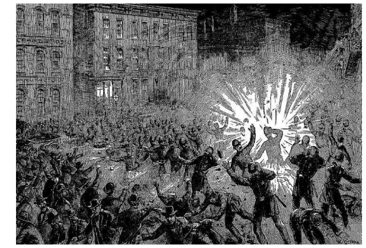
Quality of Nash-Routings

A Nash-routing is the only viable stable state in a selfish society.

In a Nash-routing, the social cost will attain some value.

How do the values of Nash-routings compare to optimal social cost?

Price of Anarchy (PoA)

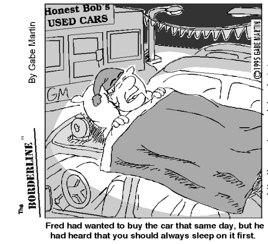


PoA: **Worst possible selfish outcome.**

$$\text{Price of Anarchy } PoA = \frac{SC \text{ of worst Nash-routing}}{SC \text{ of optimal}}.$$

Ideal: $PoA = 1$.

Price of Stability (PoS)



PoS: **Best possible selfish outcome.**

$$\text{Price of Stability } PoS = \frac{SC \text{ of best Nash-routing}}{SC \text{ of optimal}}.$$

Ideal: $PoS = 1$.

Existing Work

	Atomic Flow	Splittable Flow
Pure	[redacted], [redacted], **[BM06]**	[redacted]
Mixed	[redacted]	[redacted], [redacted]

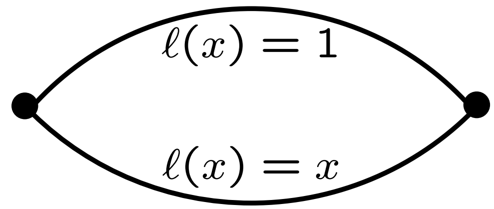
	Max SC	Sum SC	Other SC	
Max pc	**[BM06]**	–	–	[redacted]
Sum pc	[redacted], [redacted]	[redacted], [redacted]	[redacted]	[redacted]

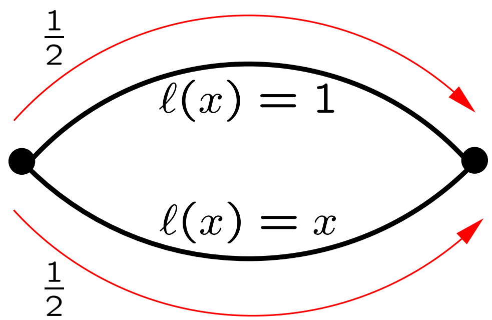
■: specific network or strategy sets (eg. parallel links or singleton sets).

■: existence or convergence to equilibrium (do not look at quality (SC)).

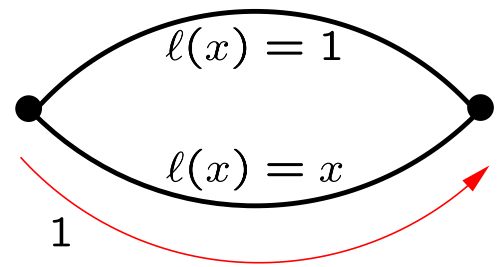
Note: sum SC is relevent when network resources, not max. player delay is important.

Linear Delay Functions





Optimal
Average Delay: $\frac{3}{4}$

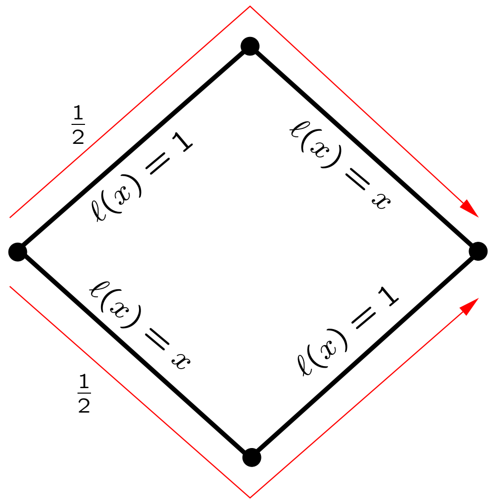


Unique Nash-routing
Average Delay: 1

$$PoA = \frac{4}{3}, \qquad PoS = \frac{4}{3}$$

This is worst possible case for: splittable flow; linear delay functions; general networks. [Roughgarden, Tardos 2002]

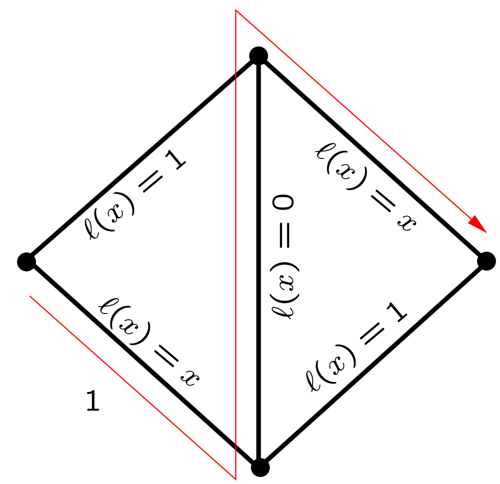
Braess' Paradox



Optimal=Nash-routing

Average Delay: $\frac{3}{2}$

$PoA = PoS = 1$



Unique Nash-routing

Average Delay: 2

$PoA = PoS = \frac{4}{3}$

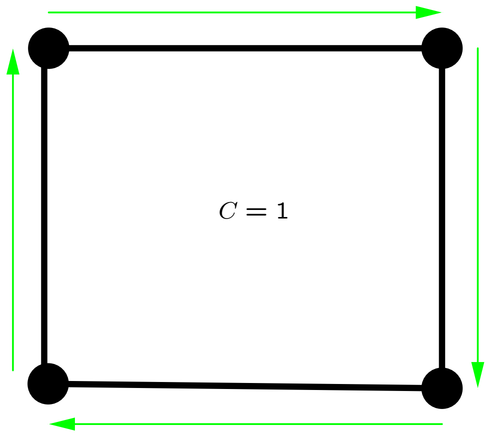
Adding a link can cause everyone to suffer!

Atomic Routing Games

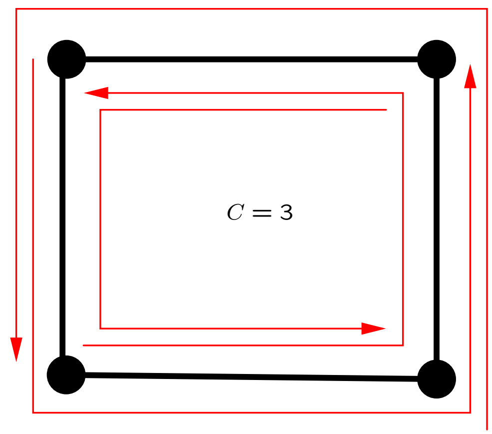
Atomic routing games: traffic is not a flow; it is non-splittable.

Delay $\sim$ max. congestion.

Nash-routings



Optimal & Nash-routing
Congestion: 1
$PoS = 1$



Worst Nash-routing
Congestion: 3
$PoA = 3$

Can generalize to N players on an N-cycle:

$$PoS = 1, \qquad PoA = N - 1.$$

Price of Anarchy

Routing games with max. player/social costs on general networks. [Busch, M. 2006]

Theorem 2 $PoA < 2(L + \log n)$.

L upper bounds path lengths in the strategy sets.
L can be small (eg. Hypercubes).

Theorem 3 $\kappa_e - 1 \leq PoA \leq c(\kappa_e^2 + \log^2 n)$.

$\kappa_e(G)$ is the length of the longest cycle.
PoA is bounded by topological properties of the network.

PoA is bad.

Price of Stability

Routing games with max. player/social costs on general networks. [Busch, M. 2006]

Theorem 1
(i) $PoS = 1$;
(ii) All best response dynamics converge to a Nash-routing

$$SC(\mathbf{p}_{final}) \leq SC(\mathbf{p}_{start}).$$

– There exist good Nash-routing.
– Starting at any good routing, selfish players can only improve!

Good Initial Routings

Randomized Oblivious Routing.

(Small Stretch)

Wrap Up

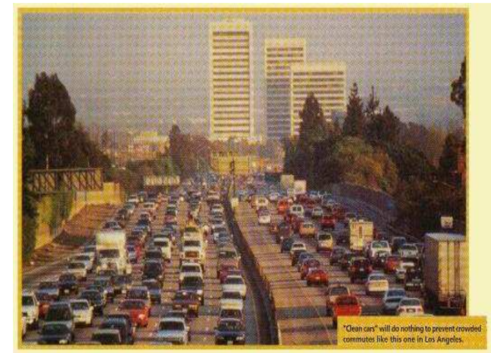
Randomized oblivious routing

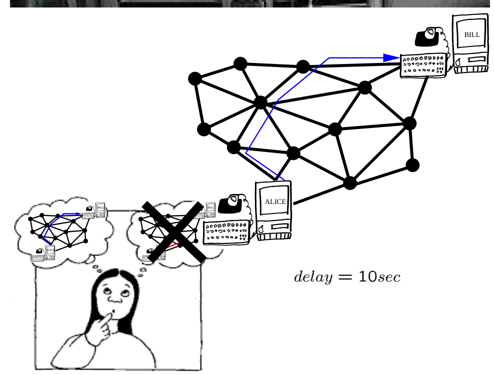
good starting routes.

$PoS = 1$ and selfish moves improves routing

greed is good.

There are bad Nash-routings: $PoA \sim \min\{\kappa_e, N, L\}$.





Thank You!

http://www.cs.rpi.edu/~magdon

Proof Sketch: $PoS = 1$

Establish a total order $\leq_c, <_c$ among routings with:

Lemma 1 There exists a minimum routing $\mathbf{p}^*$. [Compactness of routings.]

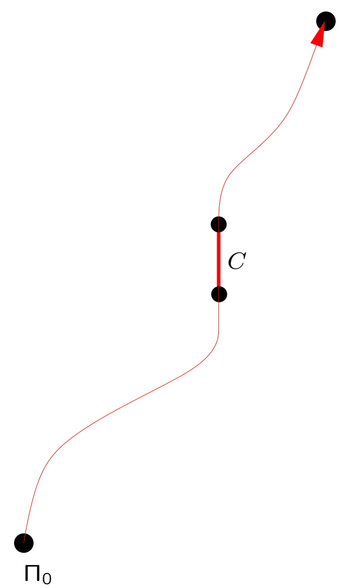
Lemma 2 $SC(\mathbf{p}) \leq SC(\mathbf{p}')$ iff $\mathbf{p} \leq_c \mathbf{p}'$.

Lemma 3 If $\mathbf{p} \rightarrow \mathbf{p}'$ in a selfish move, then $\mathbf{p}' <_c \mathbf{p} \implies SC(\mathbf{p}') < SC(\mathbf{p})$.

Corollary Minimum routings $\mathbf{p}^*$ are a Nash-routings. Best response dynamics converge to better Nash-routing.

(Note: cf. potential function methods.)

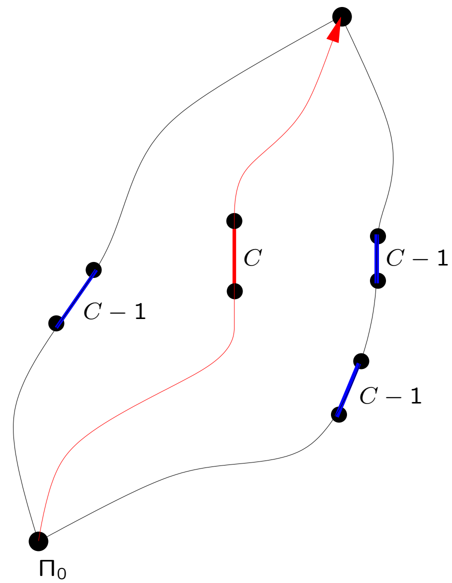
Proof Sketch: $PoA \leq 2(\ell + \log n)$



E_0: Edges of congestion C.

Π_0: Players using edges in E_0.

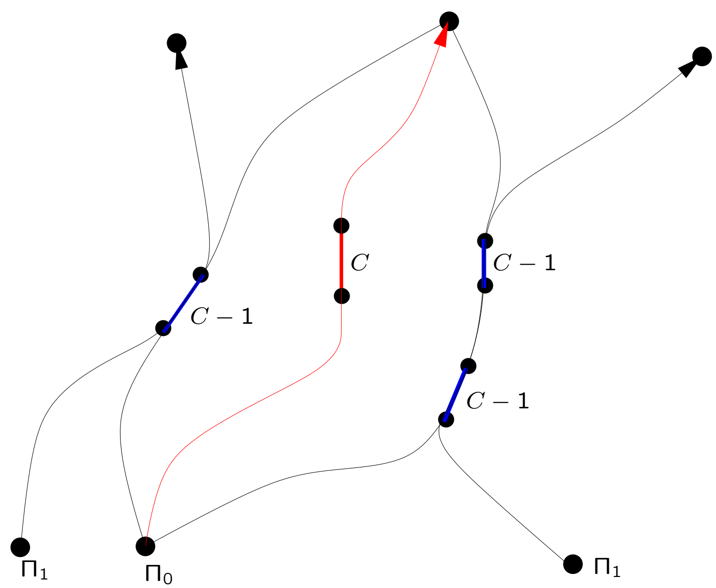
Proof Sketch: $PoA \leq 2(\ell + \log n)$



Alternative paths for players in Π_0 must all have at least one edge with congestion at least $C - 1$.

$$\left(\begin{array}{l} E_0\text{: Edges of congestion } C. \\ \Pi_0\text{: Players using edges in } E_0. \end{array} \right)$$

Proof Sketch: $PoA \leq 2(\ell + \log n)$



E_1: All these edges of congestion $\geq C-1$.

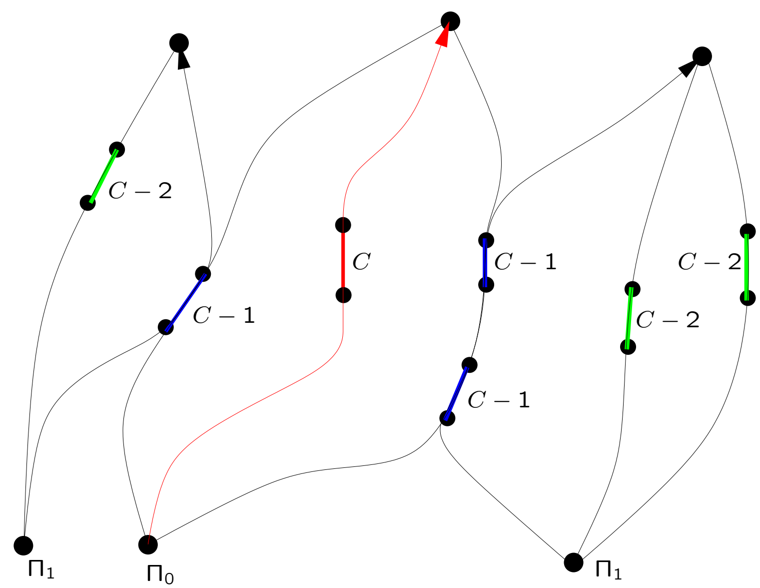
Π_1: Players using edges in E_1.

if $|E_1| \leq 2|E_0|$, stop, else continue **Edge Expansion Process**

$(|E_0| = 1, |E_1| = 4)$

(E_1 is formed from all possible paths of players in Π_0)

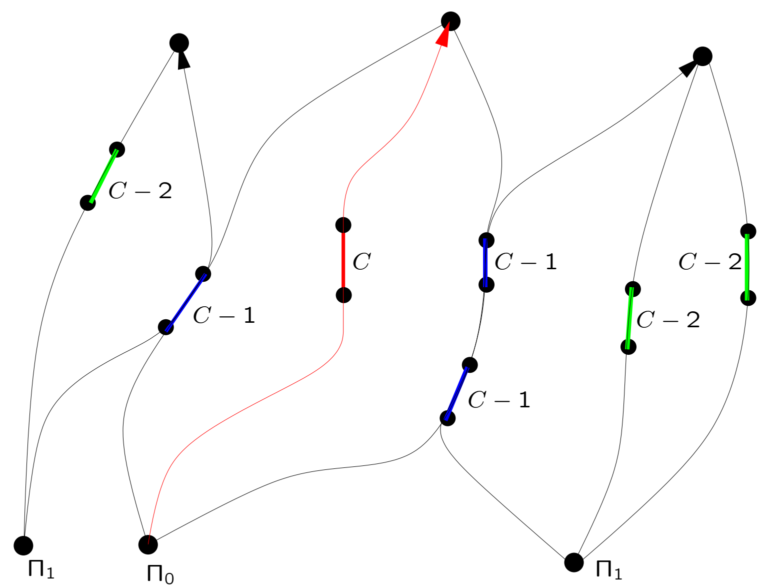
Proof Sketch: $PoA \le 2(\ell + \log n)$



Alternative paths for players in Π_1 must all have at least one edge with congestion at least $C-2$.

$$\left(\begin{array}{l} E_1\text{: Edges of congestion at least } C-1. \\ \Pi_1\text{: Players using edges in } E_1. \end{array} \right)$$

Proof Sketch: $PoA \leq 2(\ell + \log n)$



E_2: All these edges of congestion $\geq C-2$.

if $|E_2| \leq 2|E_1|$, stop.

$(|E_1| = 4, |E_2| = 7)$

(E_2 is formed from all possible paths of players in Π_1)

Proof Sketch: $PoA \le 2(\ell + \log n)$

$E_0 \quad E_1 \quad \ldots \quad E_{s-1} \quad E_s$

$\Pi_0 \quad \Pi_1 \quad \ldots \quad \Pi_{s-1}$

$s \le \log n$

(Each step doubles the size of E_i.)

Max. # times edges used by packets in Π_{s-1}

Min. # times edges in E_{s-1} used (only packets in Π_{s-1} use edges in E_{s-1})

$$|\Pi_{s-1}| \cdot \ell \ge (C - (s-1)) \cdot |E_{s-1}|$$

$$C_{opt} \ge \frac{|\Pi_{s-1}|}{|E_s|} \ge \frac{|\Pi_{s-1}|}{2|E_{s-1}|}$$

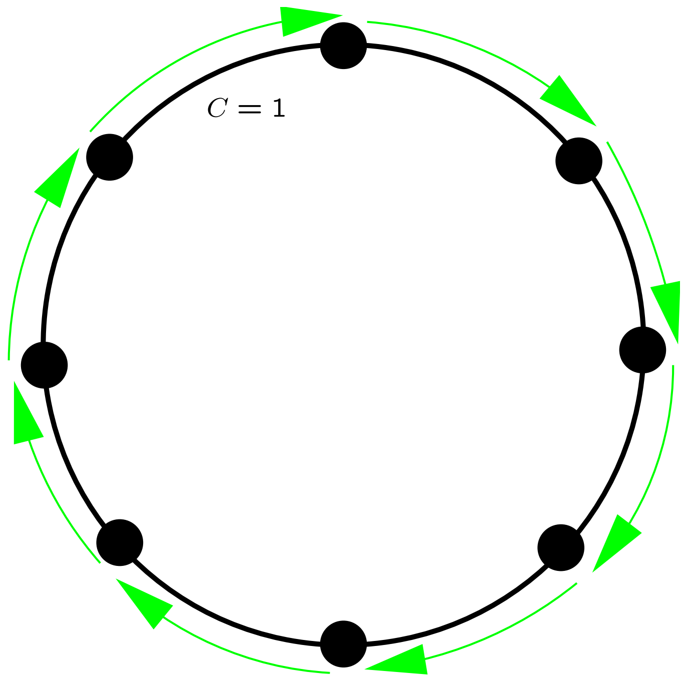
Optimal C

Every packet in Π_{s-1} must use at least one edge in E_s

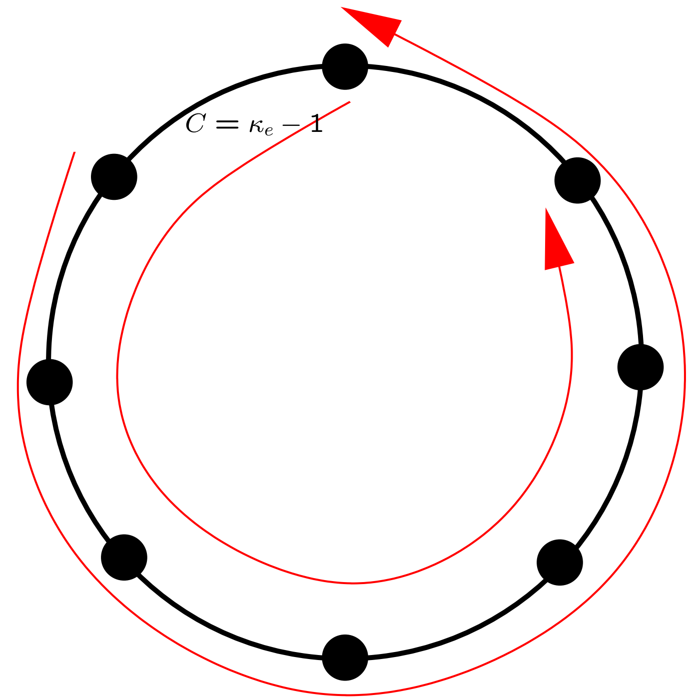
$|E_s| \le 2|E_{s-1}|$

$$\Rightarrow PoA = \frac{C}{C_{opt}} \le 2\ell + s - 1.$$

Proof Sketch: $\kappa_e - 1 \leq PoA \leq c(\kappa_e^2 + \log n)$



Optimal Nash-routing
(Players use shortest paths)
$C = 1$



Worst Case Nash-routing
(Players use longest paths)
$C = n - 1 = \kappa_e - 1$

If network is not a cycle, use the largest cycle in the network.

Proof Sketch: $\kappa_e - 1 \leq PoA \leq c(\kappa_e^2 + \log n)$

Combinatorial Lemma If G is 2-connected, then $\kappa_e(G) \geq \sqrt{2\ell} - \frac{3}{2}$.

2-connected Networks:

$\ell = O(\kappa_e^2)$, so

$PoA \leq 2(\ell + \log n) \implies PoA = O(\kappa_e^2 + \log^2 n)$.

General Networks:

Step 1: Decompose G: tree of 2-conected and acyclic components.

Step 2: Many players satisfied in some 2-connected component;

Step 3: Extend $PoA \leq 2(\ell + \log n)$ to **Partial Nash-routing**.

Step 4: Use 2-connected and Partial Nash-routing results.