Price of Anarchy in Network Games

Éva Tardos
Cornell University

What is Price of Anarchy?

- Same as efficiency loss

Papadimitriou-Koutsoupias '99

$$\frac{\text{cost of selfish solution}}{\text{“socially optimum” cost}}$$

- Closely related to approximation algorithms

$$\text{approx ratio} = \frac{\text{cost of algorithm}}{\text{“optimum” cost}}$$

Network Games?

Main topic:

Routing to minimize delay:

- Given graph. Users choose routes to send their traffic to minimize their own delay or cost

Joint work with Tim Roughgarden, and with Henry Lin, and Asher Walkover

Other Network Games?

Resource allocation games:

- Ramesh Johari - John Tsitsiklis
- Bruce Hajek - …
- …

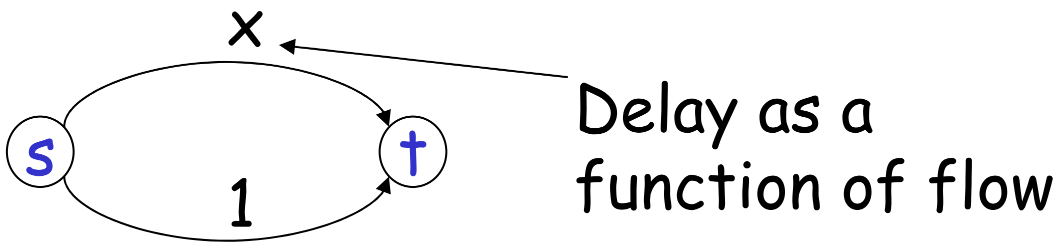
Network Design games:

- Users cooperate to build a graph that connects their terminals to minimize their own cost

Joint work with Elliot Anshelevich, Anirban Dasgupta, and Tom Wexler

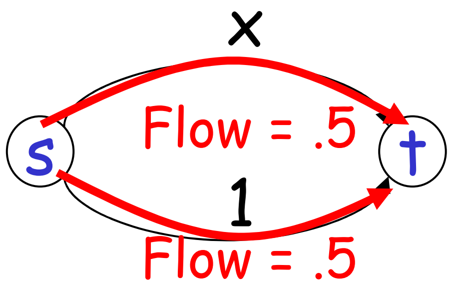
Congestion sensitive routing: Example

Congestion sensitive network



Congestion sensitive routing: Example

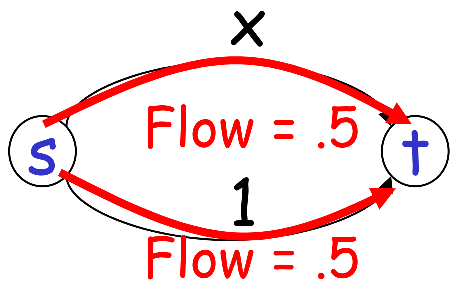
One unit of flow sent from s to t



Traffic on lower edge is envious.

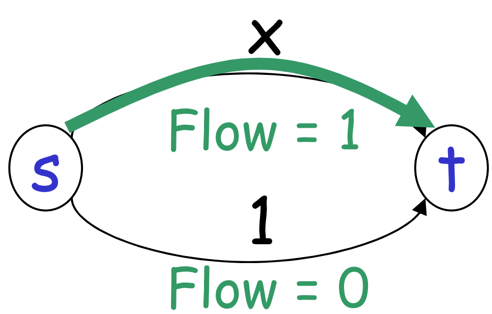
Congestion sensitive routing: Example

One unit of flow sent from s to t



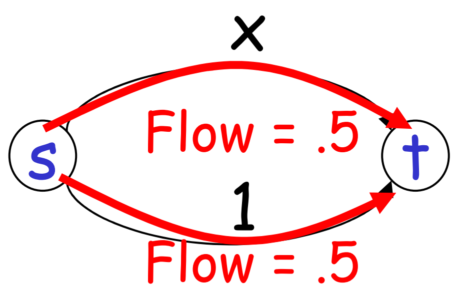
Traffic on lower edge is envious.

An envy free solution:



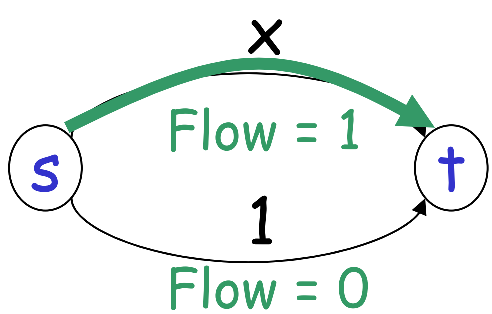
Congestion sensitive routing: Example

One unit of flow sent from s to t



Traffic on lower edge is envious.

An envy free solution:



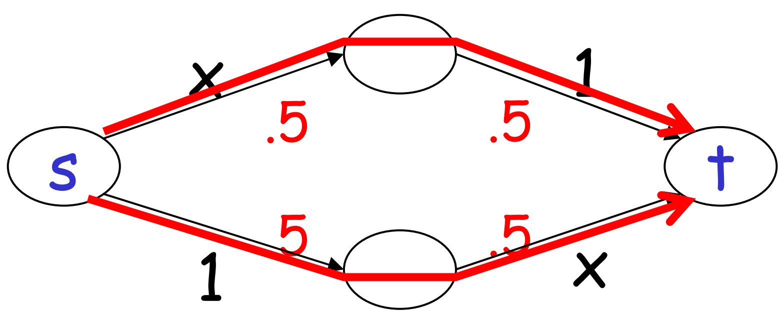
No-one is better off

Agents are selfish want to

- minimize personal latency,
- do not care about welfare of others

Braess's Paradox

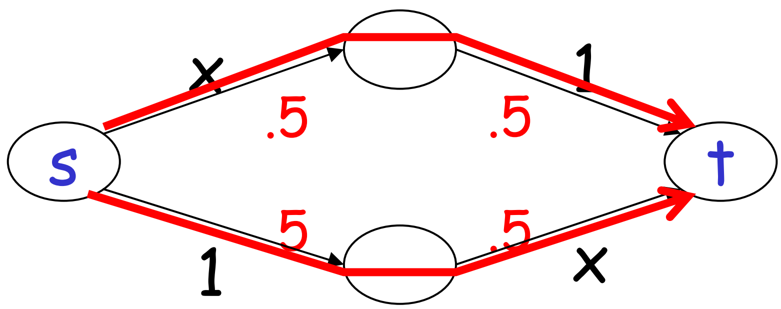
Original Network



Cost of Nash flow = 1.5

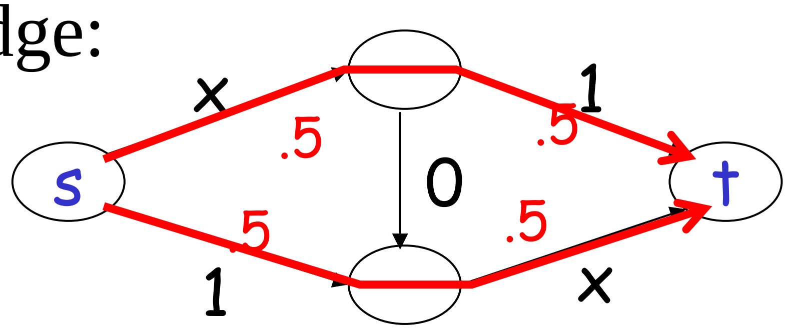
Braess's Paradox

Original Network



Cost of Nash flow = 1.5

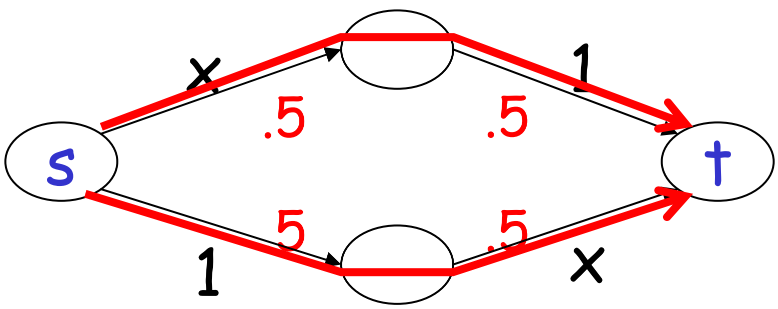
Added edge:



Effect?

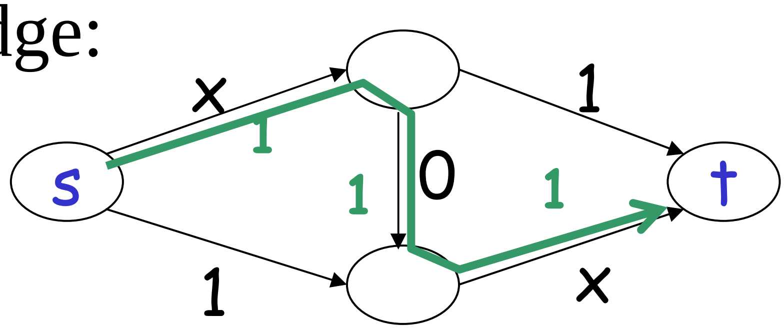
Braess's Paradox

Original Network



Cost of Nash flow = 1.5

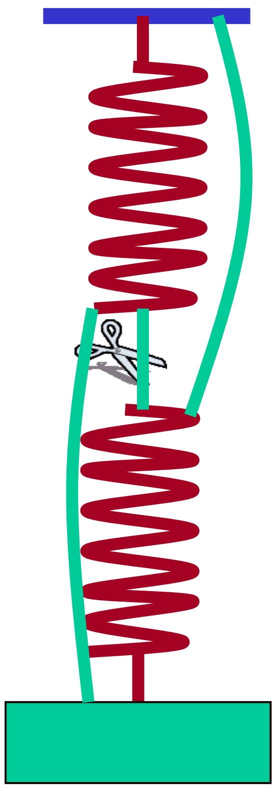
Added edge:



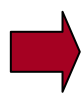
Cost of Nash flow = 2

All the flow has increased delay!

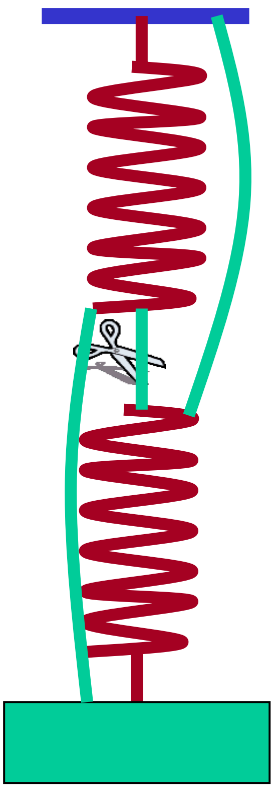
Analogous spring and power network paradox



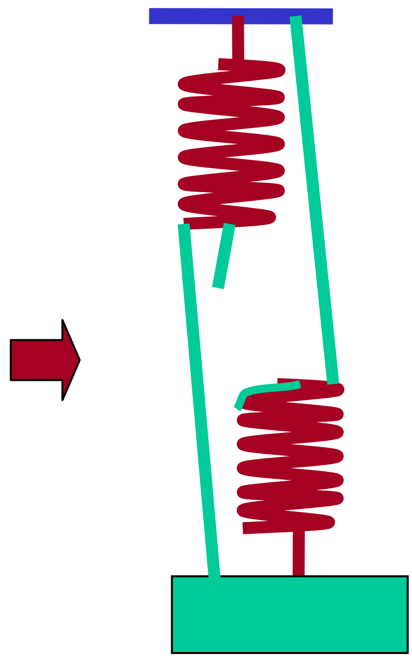
Cutting middle string



Analogous spring and power network paradox



Cutting middle string

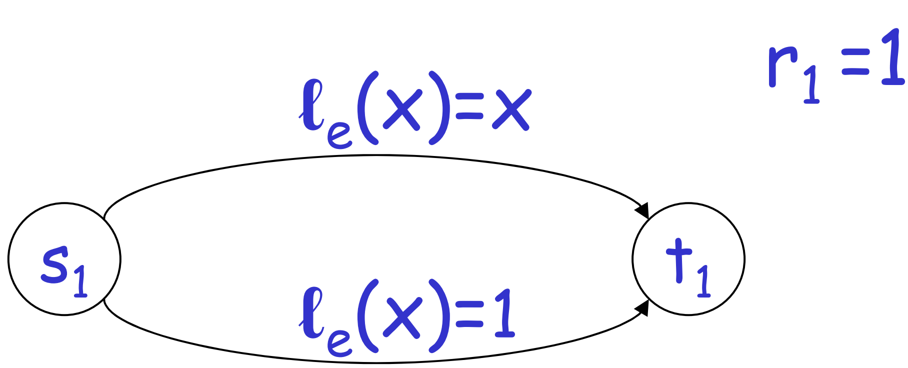


makes the weight rise

and decreases power flow along springs

Routing Traffic in Congested Networks
(Mathematical model)

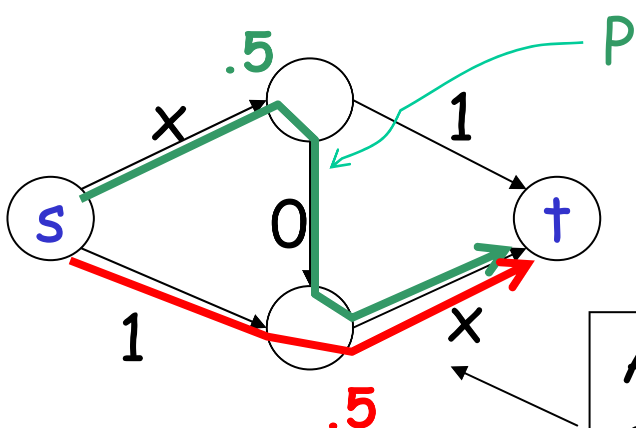
- A directed graph $G = (V,E)$
- source-sink pairs s_i,t_i for $i=1,..,k$
- rate $r_i \geq 0$ of traffic between s_i and t_i for each $i=1,..,k$
- For each edge e, a latency function $\ell_e(\cdot)$



Flows in Larger Networks

Traffic and Flows:

- f_P = amount routed on s_i-t_i path P



$\ell_P(f) = .5 + 0 + 1$

All flow on edge contributes to edge latency

The latency of a Paths

- $\ell_P(f)$ = sum of latencies of edges along P (w.r.t. flow f)

Flows at Equilibrium

A flow is at Nash equilibrium if no agent can improve its latency by changing its path

Assumptions:

- edge latency functions are continuous, and non-decreasing
- Users are very small, cannot effect latency individually

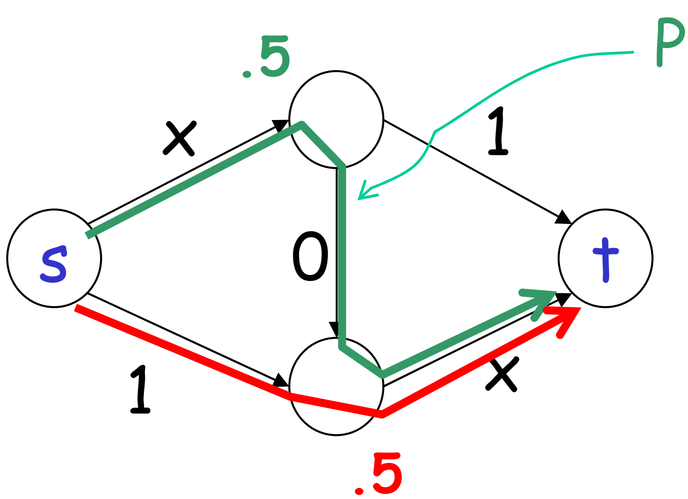
Corollary: flow f is at equilibrium if and only if all flow travels along minimum-latency paths (w.r.t. f)

Theorem: [Beckmann et al 56] The Nash equilibrium exits and is essentially unique

Social Cost of Flows (1)

C(f) = total latency of a flow f:

$$\Sigma_P f_P \cdot \ell_P(f)$$

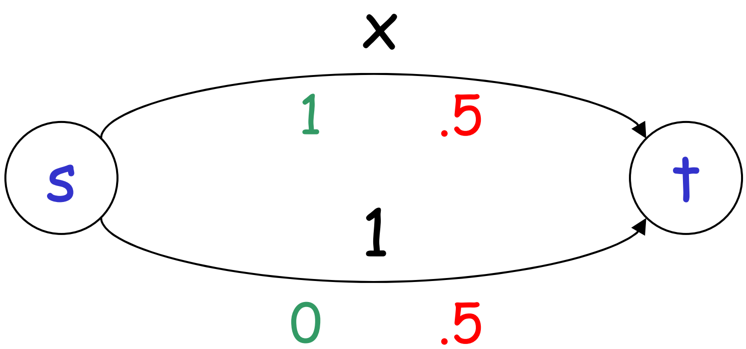


$\ell_P(f) = .5 + 0 + 1$

Question: To what extent does a Nash flow optimize social welfare?

Efficiency Loss

Cost of flow: "social welfare"

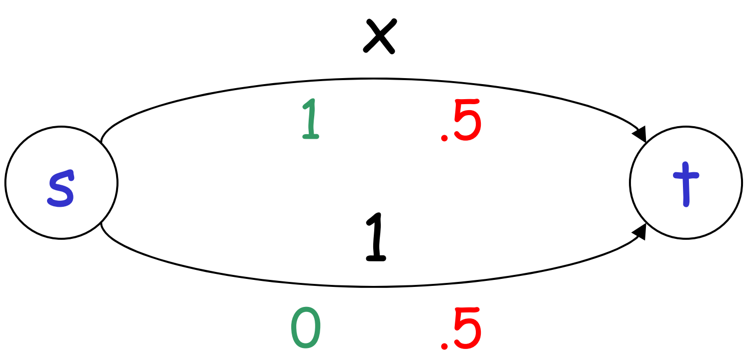


Cost of Nash flow = 1•1 + 0•1 = 1

Cost of optimal (min-cost) flow
= .5•.5 +.5•1 =.75

Cost of Selfishness

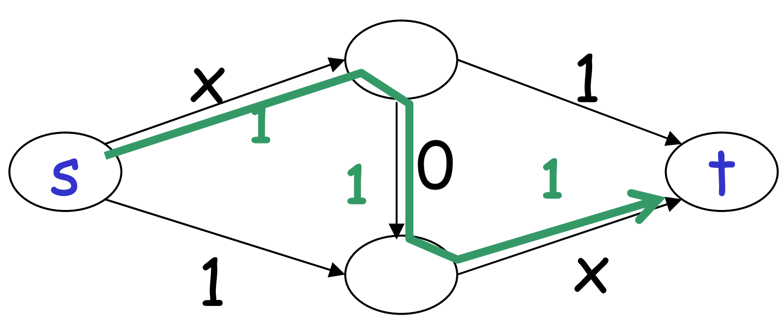
Cost of flow: "social welfare"



Cost of Nash flow = 1•1 + 0•1 = 1

Cost of optimal (min-cost) flow
= .5•.5 +.5•1 =.75

Same ration also in Braess's paradox:



Results

Theorem 1

(Roughgarden-Tardos)

In a network with linear latency functions

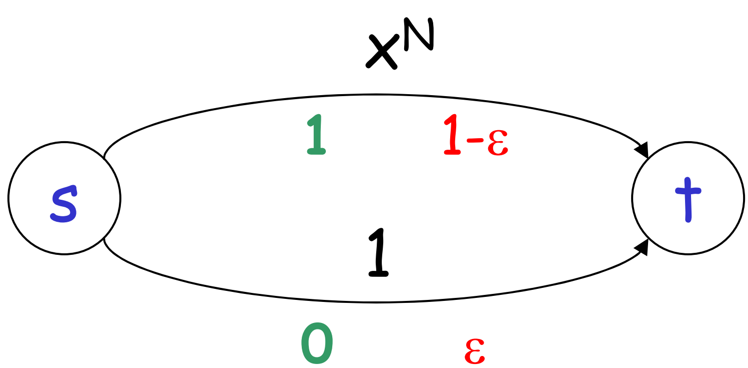
- i.e., of the form $\ell_e(x)=a_e x+b_e$

the cost of a Nash flow is at most 4/3 times that of the minimum-latency flow

General Latency Functions

Question: what about more general edge latency functions?

Bad Example: (r = 1, N large)



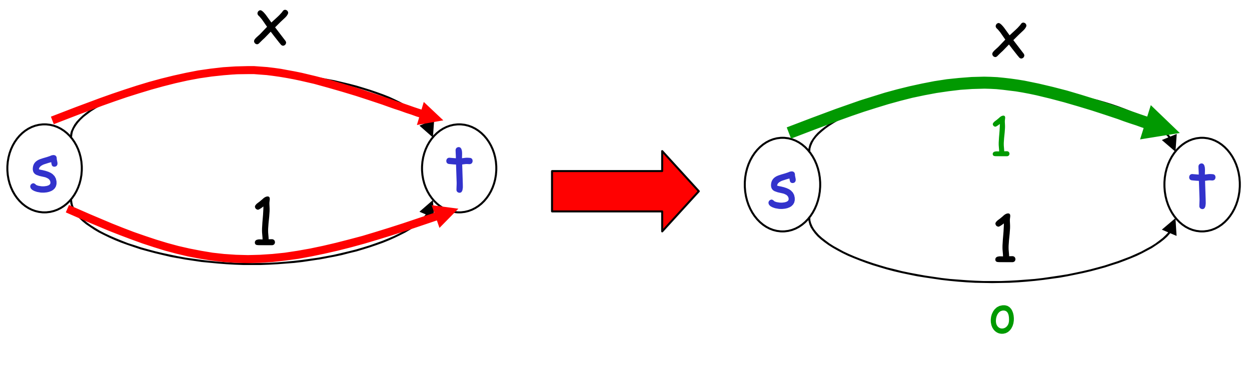
A Nash flow can cost arbitrarily more than the optimal (min-cost) flow

Results

Theorem 2 (Roughgarden):

In any network with any class of convex continuous latency functions

the worst price of anarchy is always on two edge network



Results

Theorem 3

(Roughgarden-Tardos):

In any network with continuous, nondecreasing latency functions

cost of Nash with rates r_i for all i	$\leq$	cost of opt with rates $2r_i$ for all i

Results

Theorem 3

(Roughgarden-Tardos):

In any network with continuous, nondecreasing latency functions

cost of Nash with rates r_i for all i	$\leq$	cost of opt with rates $2r_i$ for all i

Corollary: with M/M/1 delay fns: $\ell(x)=1/(u-x)$, where u=capacity

Nash w/cap. $2u \leq$ opt w/cap. u

Morale for IP versus ATM?

IP today no worse than

ATM a year from now …

- We assumed source routing, but
- IP routing would also lead to the same equilibrium, if delay is primary metric

Morale? instead of

- building central control
- build networks that support twice as much traffic

Outline

- Brief proof sketch (all three results together)
- Other results about routing with congestion
- Network design game

Characterizing the Min-Cost Flow

Min-latency flow

- for one s-t pair for simplicity

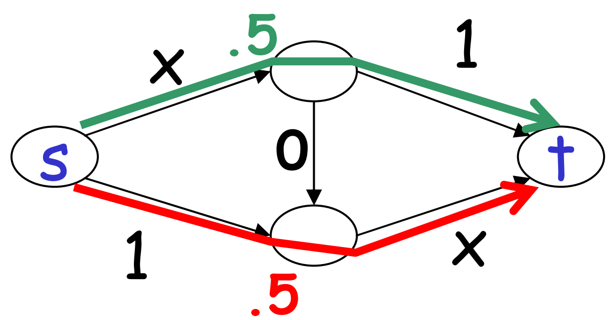
minimize $C(f) = \Sigma_e f_e \cdot \ell_e(f_e)$

subject to: f is an s-t flow carrying r units

By summing over edges rather than paths where f_e = amount of flow on edge e

Characterizing the Optimal Flow

Optimality condition: all flow travels along minimum-gradient paths



gradient is:

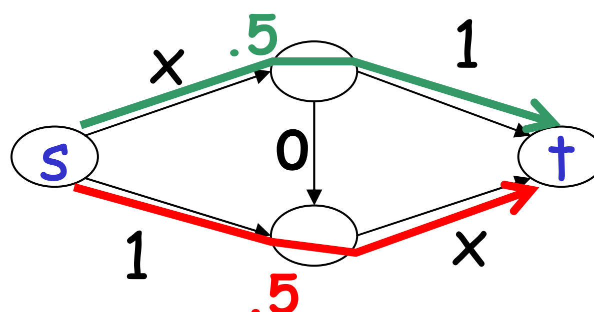
$(x\,\ell(x))'$

$= \ell(x) + x\,\ell'(x)$

Characterizing the Optimal Flow

Optimality condition: all flow travels along minimum-gradient paths

gradient is:
$(x\,\ell(x))'$
$= \ell(x)+x\,\ell'(x)$

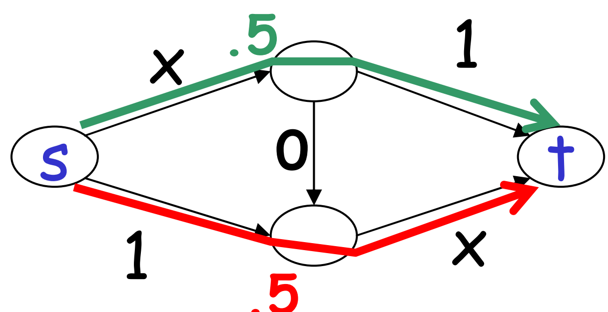


Recall: flow f is at Nash equilibrium iff all flow travels along minimum-latency paths

Characterizing the Optimal Flow

Optimality condition: all flow travels along minimum-gradient paths

gradient is:
$(x\,\ell(x))'$
$= \ell(x)+x\,\ell'(x)$

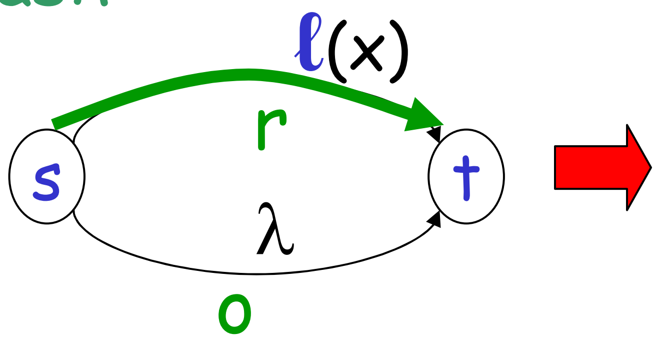


Recall: flow f is at Nash equilibrium iff all flow travels along minimum-latency paths

Corollary: Nash with "latency" $\ell(x)+x\,\ell'(x)$ is optimal flow

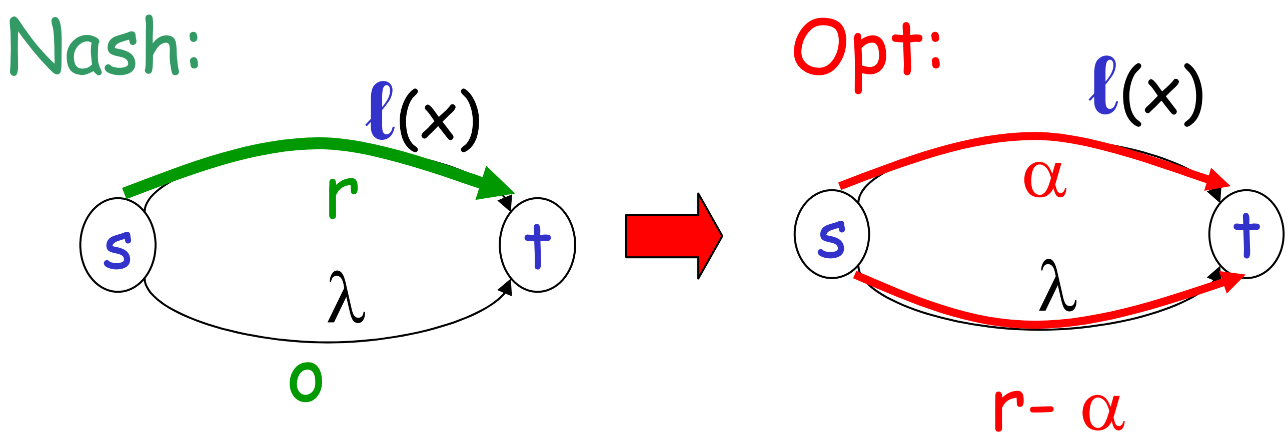
The worst case on two edge network

Nash: Opt?



With $\lambda = \ell(r)$

The worst case on two edge network



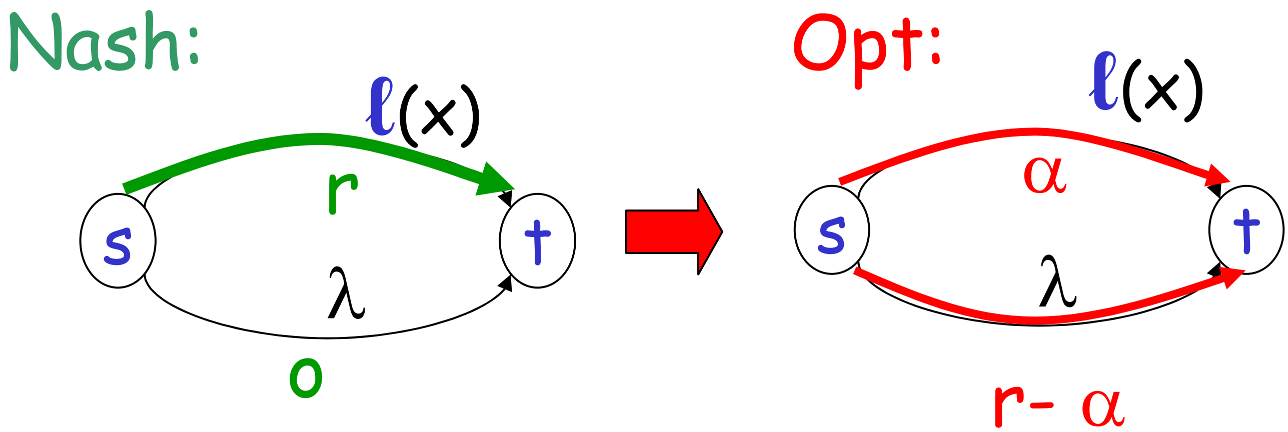
With $\lambda = \ell(r)$

To get opt select α such that

gradient in opt =delay in Nash

$$\ell(\alpha) + \alpha\, \ell'(\alpha) = \ell(r) = \lambda$$

The worst case on two edge network



With $\lambda = \ell(r)$

To get opt select α such that

gradient in opt =delay in Nash

$$\ell(\alpha) + \alpha\, \ell'(\alpha) = \ell(r) = \lambda$$

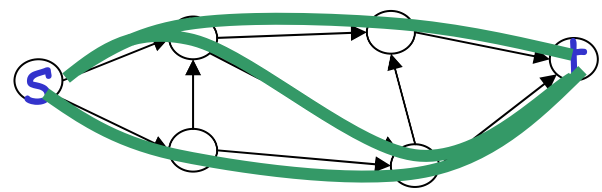
Fact: Worst case for linear 3/4

Proof idea

The worst price of anarchy

- all flow along min latency path

Nash flow f:

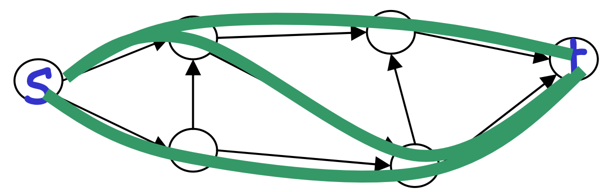


Proof idea

The worst price of anarchy

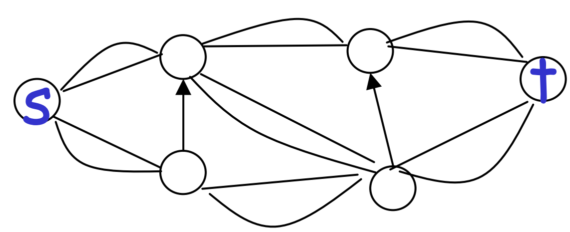
- all flow along min latency path

Nash flow f:



an augmented network:

- add new fixed latency parallel edges
- Latency $\ell_e(f_e)$

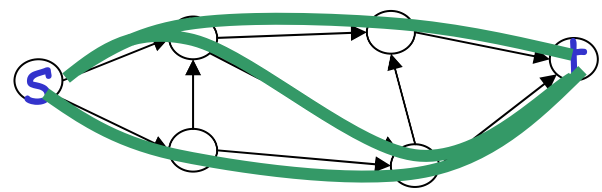


Proof idea

The worst price of anarchy

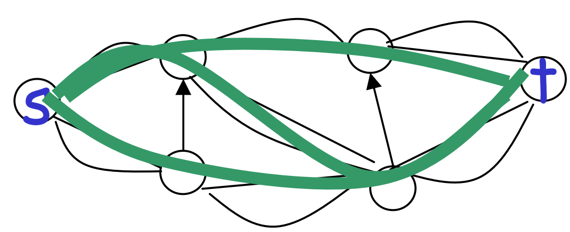
- all flow along min latency path

Nash flow f:



an augmented network:

- add a fixed latency parallel edge
- Latency $\ell_e(f_e)$



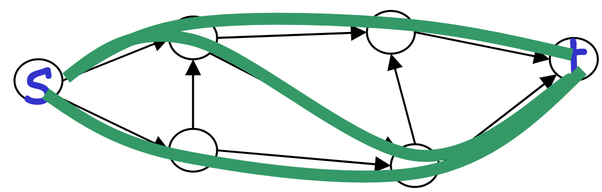
Facts:

- **Nash** not affected

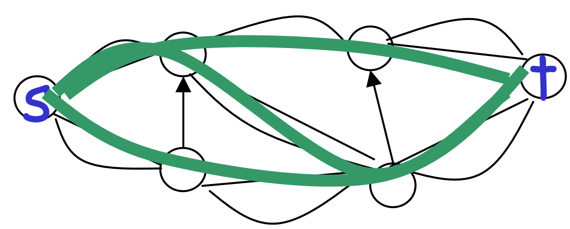
Proof idea

The worst price of anarchy
- all flow along min latency path

Nash flow f:



an augmented network:
- add a fixed latency parallel edge
- Latency $\ell_e(f_e)$

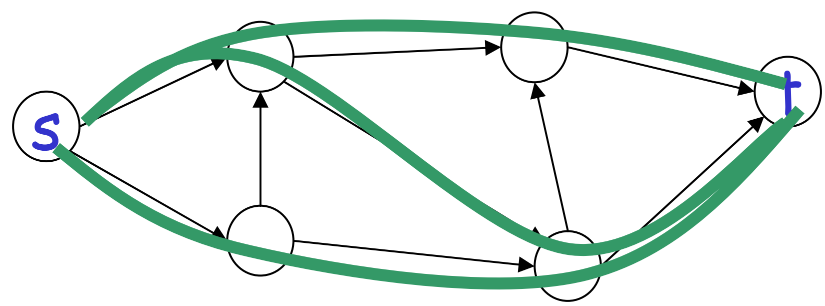


Facts:

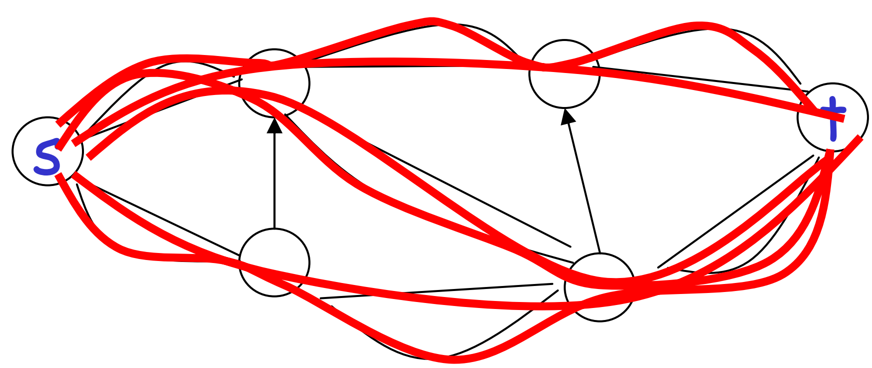
- **Nash** not affected
- **opt** can only improve

Proof idea

Nash flow:



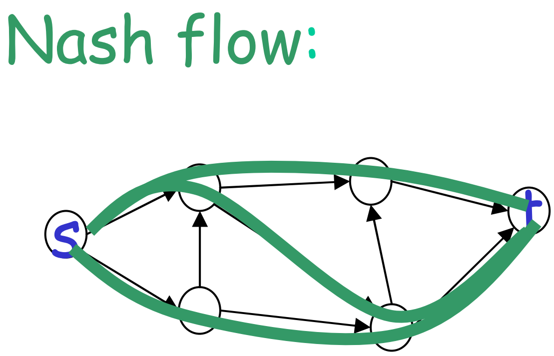
opt in an augmented network

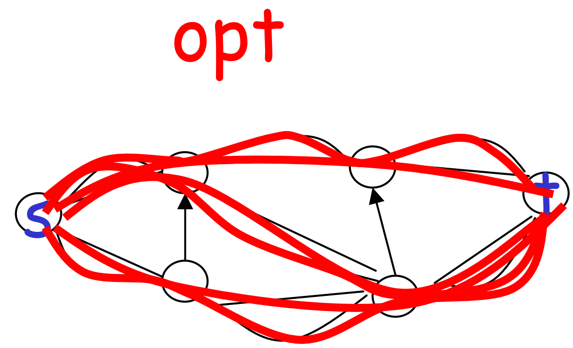


To get opt modify Nash → opt on each pair of links!

Delay in Nash = gradient in opt

Proofs



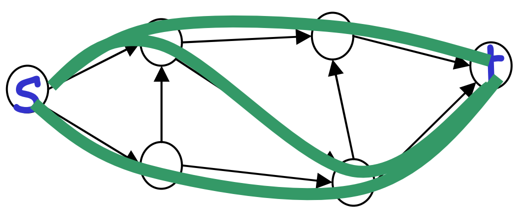


Theorem 2: the worst price of anarchy is always two edge network

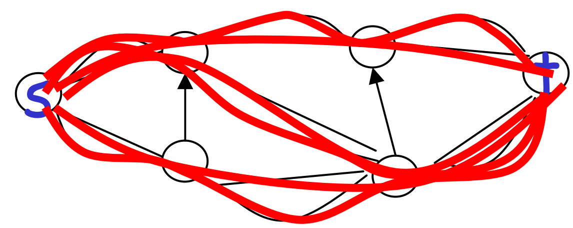
Proof: Prize of anarchy is median of ratios for the edges

Proofs

Nash flow:



opt



Theorem 3 Cost of Nash at rate r at most cost of Opt at rate 2r

Proof: Opt may cost very little, but marginal cost is as high as latency in Nash

→ Augmenting to double rate costs at least as much as Nash

Summary

Goal: prove that loss of performance in network routing due to selfishness is limited

Fact: Nash flow can cost far more than optimal

Solutions:

- Compare to optimal at an increased rate
- Restrict allowable delay functions

Results extend to nonatomic congestion games TR'02

But only work with inelastic demands

Other Models?

Minimizing maximum delay instead of average?

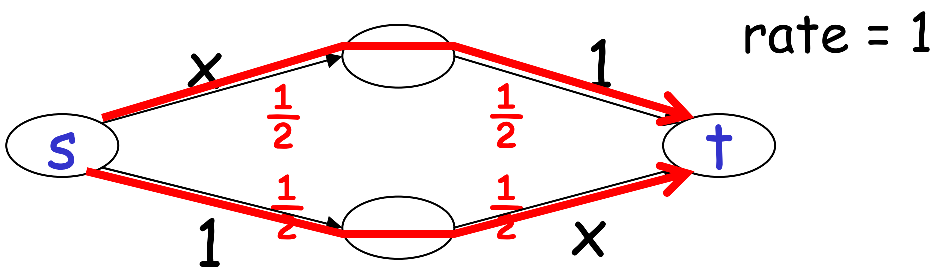
Prize of anarchy bounded independent of delay functions

- [Roughgarden'04] at most n-1 for a single commodity
- [Lin-Roughgarden-Tardos-Walkover'04]
- Exponential, but finite bound for multicommodity

Note: average delay had no bound in terms of size of graph!

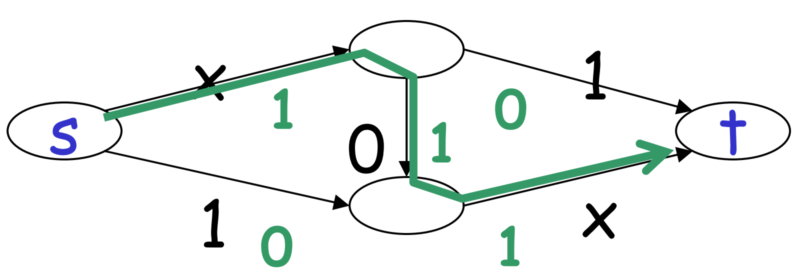
Network Design

Recall Braess's Paradox



Cost of
Nash flow
= 1.5

improved network: add one edge
worth traffic problems

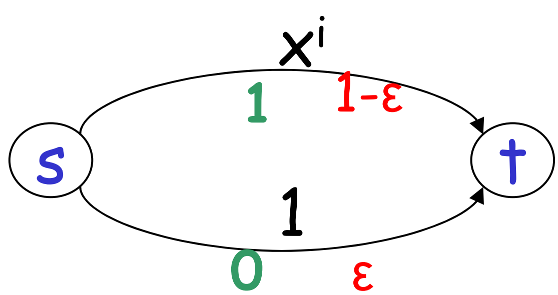


Cost of
Nash flow
= 2

How Bad is Braess' Paradox?

Given graph **G=(V,E)** and delays.
How much can we improve Nash by deleting edges

A Nash flow can cost arbitrarily more than the optimal (min-cost) flow

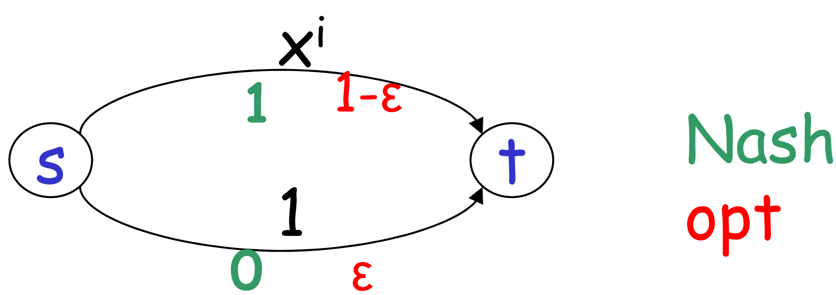


Nash
opt

How Bad is Braess Paradox?

Given graph **G=(V,E)** and delays.
How much can we improve Nash by deleting edges

A Nash flow can cost arbitrarily more than the optimal (min-cost) flow

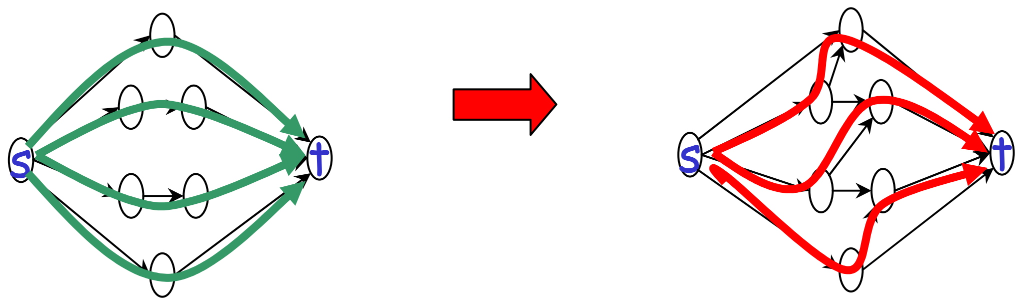


But…

All sub-graphs have the same Nash value

How Bad is Braess Paradox?

- $\leq$ factor of 2 on Braess's graph
- But there are worse graphs

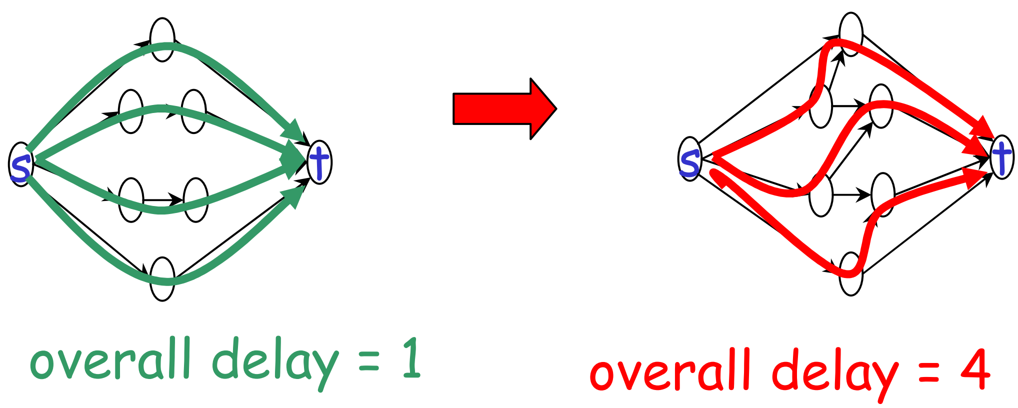


overall delay = 1

overall delay = 4

How Bad is Braess Paradox?

- ≤ factor of 2 on Braess's graph
- But there are worse graphs



Theorem For a single commodity

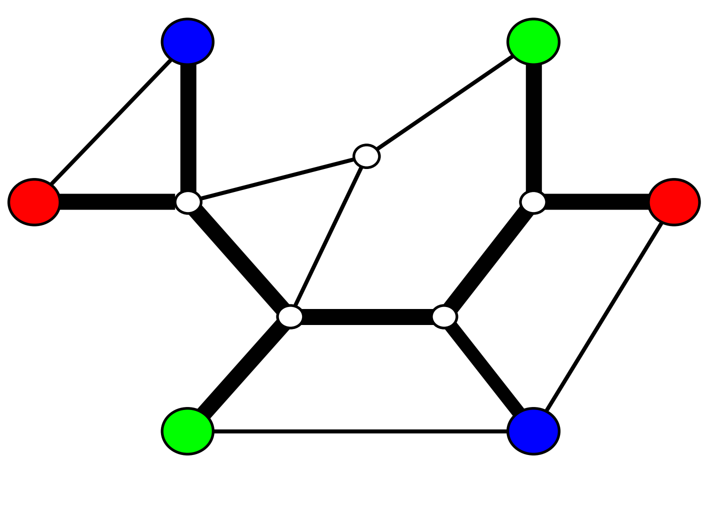
- at most n/2 [Roughgarden'01]
- at most k+1, when adding k edges [Lin, Roughgarden, Tardos'04]

Other Network Design Game

[Anshelevich-Dasgupta-T.-Wexler'03]

Building a network by many self-interested agents.

- decide on network to build and
- share cost

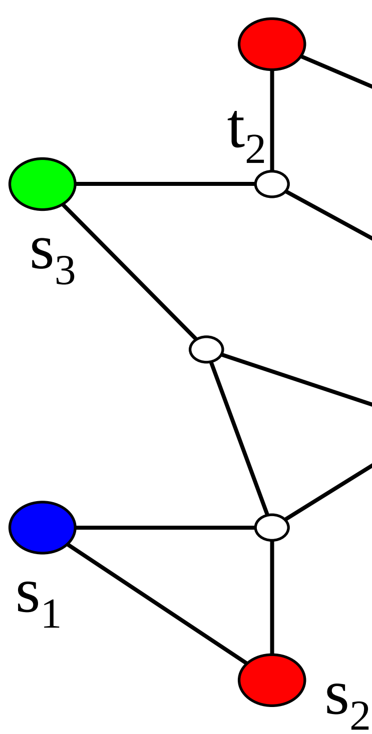


Our Model

$G = (V,E)$ is an undirected graph with edge costs $c(e)$.

There are k players.

Each player i has a source s_i and a sink t_i he wants to connect.

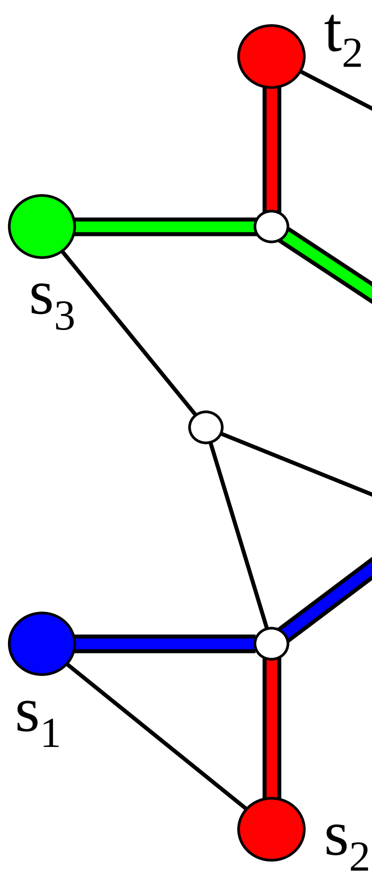


Our Model

Player i picks payments for each edge

e is bought if total payments $\geq c(e)$

Note: any player can use bought edges

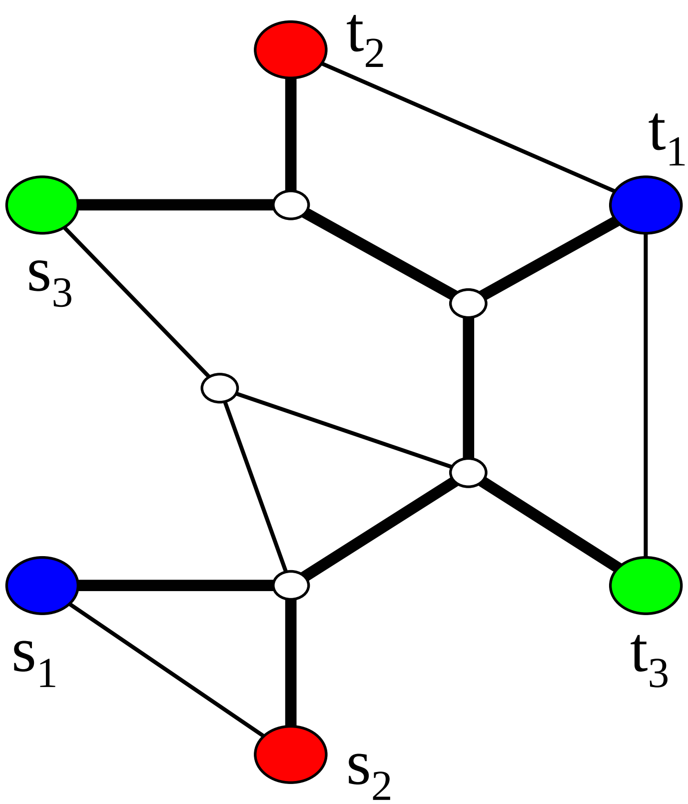


The Network Game

Each player i has only 2 concerns:

1) Must connect terminals s_i and t_i

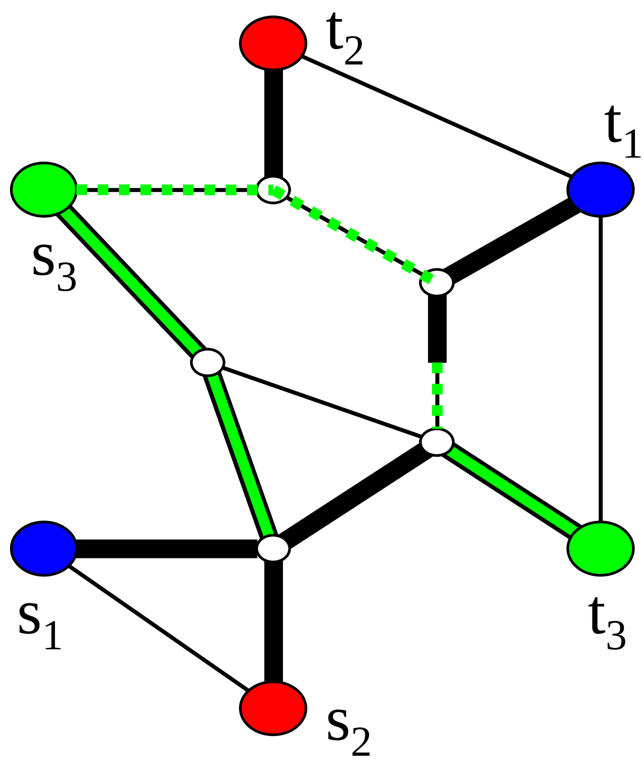
2) Wants to pay as little as possible



Nash Equilibrium

A *Nash Equilibrium* is a set of payments such that no player wants to change routing or payments.

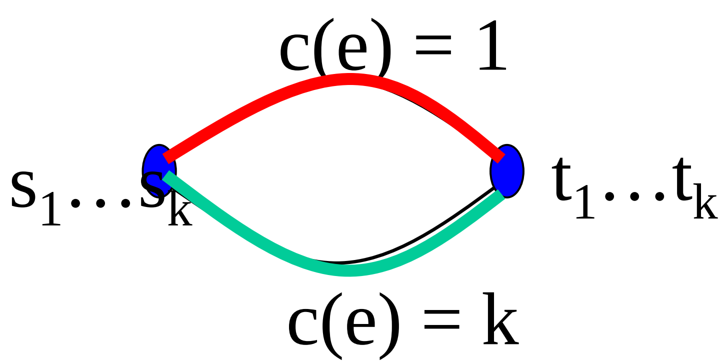
- Player i does not care whether other players connect.



Examples

Cheap Nash: Each player pays $1/k$ to top edge.

Expensive Nash: Each player pays 1 to bottom edge.



With more terminals:

may not (pure) have Nash

Our Results 2003

Anshelevich-Dasgupta-T.-Wexler'03

Single Source Game:

Theorem: Optimistic Price of Anarchy is 1: there is always a Nash that buys OPT.

Our Results 2003

Anshelevich-Dasgupta-T.-Wexler'03

Single Source Game:

Theorem: Optimistic Price of Anarchy is 1: there is always a Nash that buys OPT.

Multi Source Game:

- Nash may not exists:
- Even when it does price of anarchy can be bad… but

Theorem: There is a 3-approximate Nash that buys OPT.

Conclusion

We saw some interesting games where the outcome of selfishness near-optimal.

- Routing with simple (e.g., linear) delays.
- Routing: selfishness can be compensated by increased capacity.

- Network design: limited selfishness can lead to good outcome.

Question: Are the further interesting network games where such statements can be made?