Selfish Routing and the Price of Anarchy

Tim Roughgarden
Cornell University

Algorithms for Self-Interested Agents

Our focus: problems in which multiple agents (people, computers, etc.) interact

Motivation: the Internet

- decentralized operation and ownership

Traditional algorithmic approach:

- agents classified as obedient or adversarial
  - examples: distributed algorithms, cryptography

Algorithms and Game Theory

Recent trend: agents have own independent objectives (and act accordingly)

New goal: algorithms that account for strategic behavior by self-interested agents

Natural tool: game theory

- theory of "rational behavior" in competitive, collaborative settings
  - [von Neumann/Morgenstern 44]

Objectives

This talk: understand consequences of noncooperative behavior

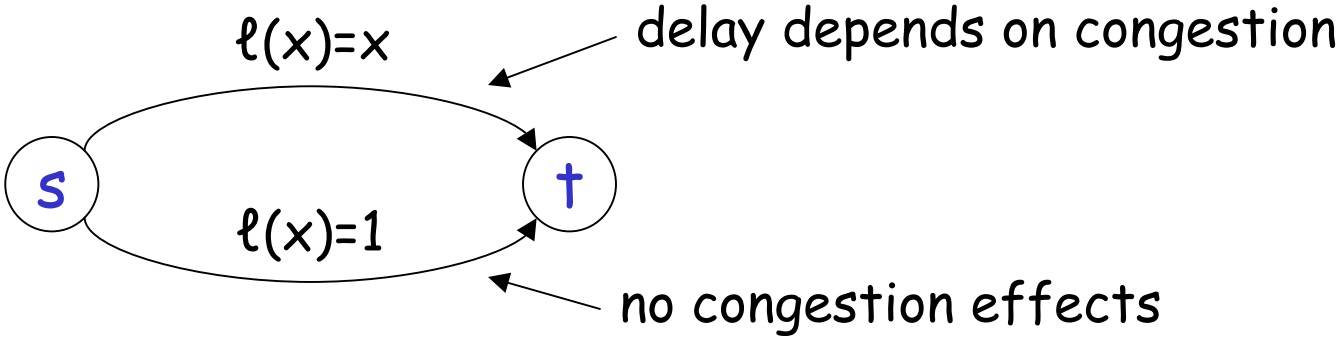
- when is the cost of selfish behavior severe?
  - the "price of anarchy" [Koutsoupias/Papadimitriou 99]
- what can we do about it?
  - design strategies, economic incentives

Our setting: routing in a congested network

- will focus on [Roughgarden/Tardos FOCS '00/JACM '02]
- and also [Roughgarden STOC '02/JCSS to appear]

Motivating Example

Example: one unit of traffic wants to go from s to t

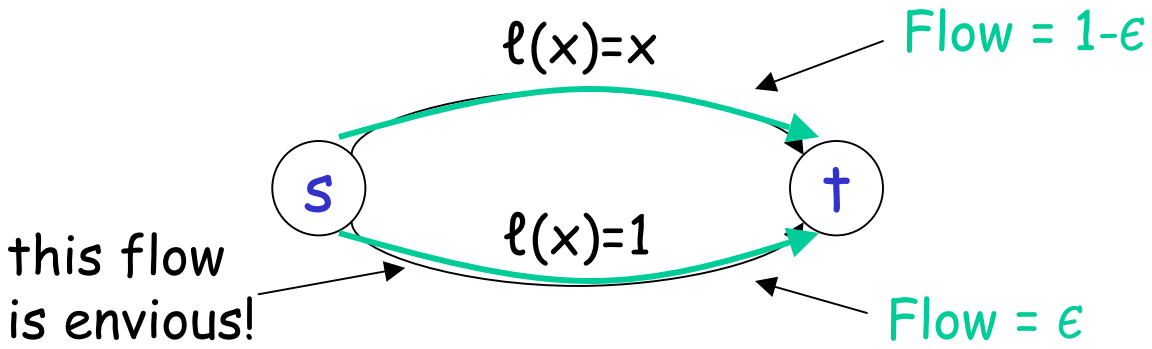


Question: what will selfish network users do?

- assume everyone wants smallest-possible delay

Motivating Example

Claim: all traffic will take the top link.

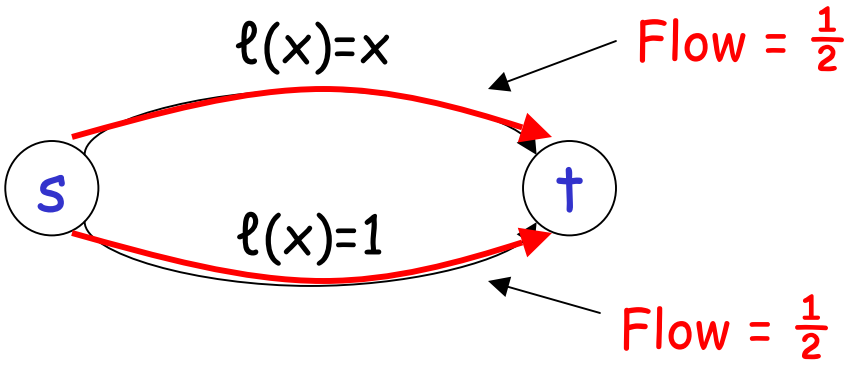


Reason:

- $\epsilon > 0 \Rightarrow$ traffic on bottom is envious
- $\epsilon = 0 \Rightarrow$ envy-free outcome
  - all traffic incurs one unit of delay

Can We Do Better?

Consider instead: traffic split equally

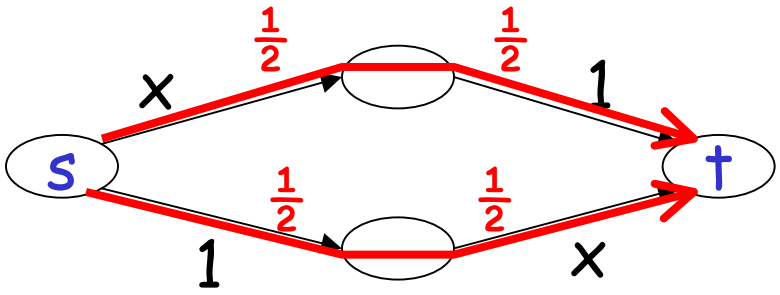


Improvement:

- half of traffic has delay 1 (same as before)
- half of traffic has delay ½ (much improved!)

Braess's Paradox

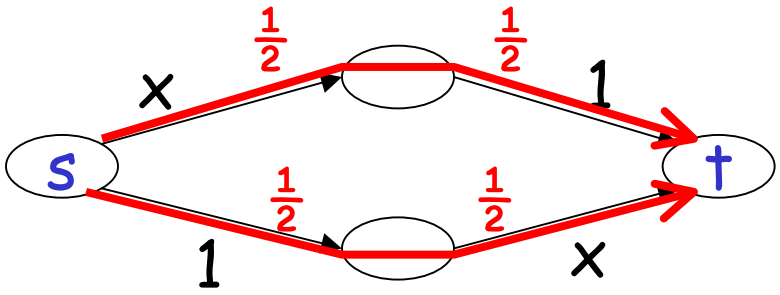
Initial Network:



Delay = 1.5

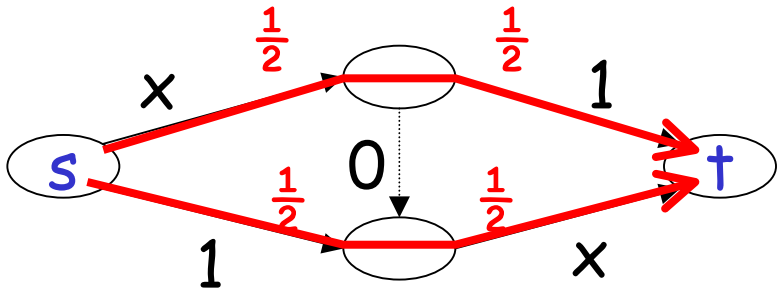
Braess's Paradox

Initial Network:



Delay = 1.5

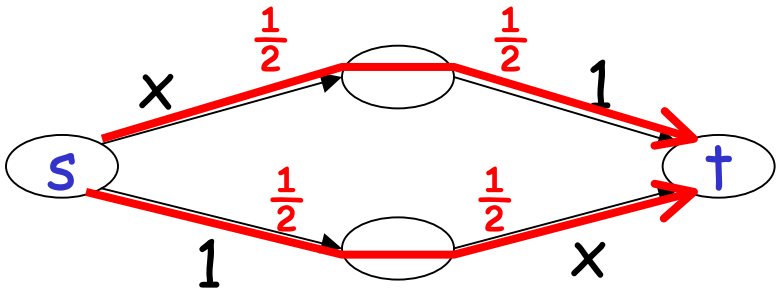
Augmented Network:



Now what?

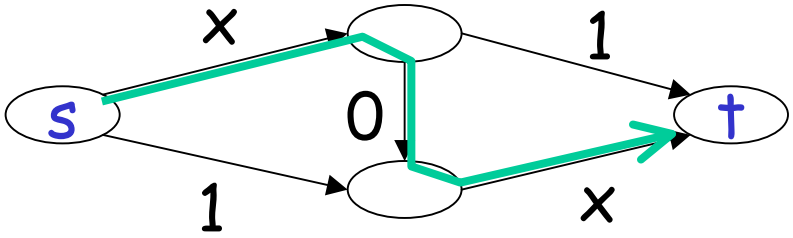
Braess's Paradox

Initial Network:



Delay = 1.5

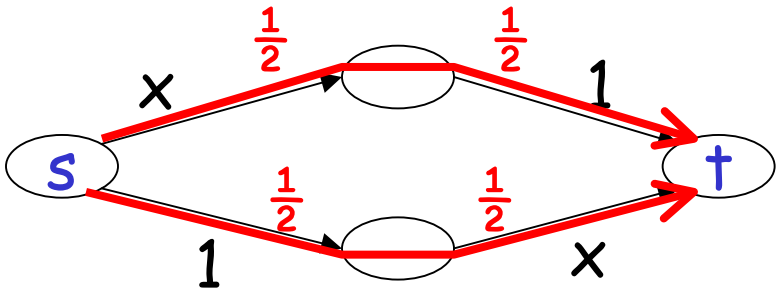
Augmented Network:



Delay = 2

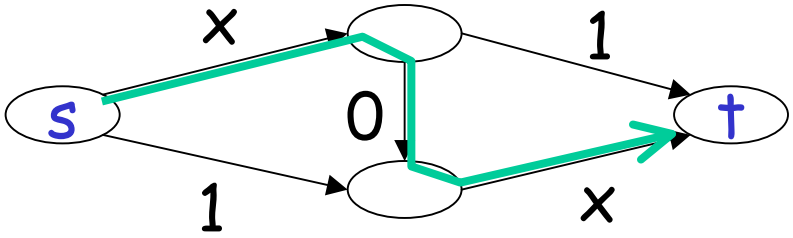
Braess's Paradox

Initial Network:



Delay = 1.5

Augmented Network:



Delay = 2

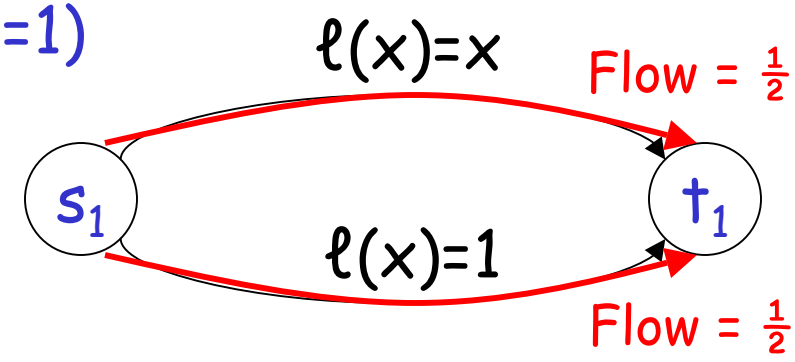
All traffic incurs more delay! [Braess 68]

- also has physical analogs [Cohen/Horowitz 91]

The Mathematical Model

- a directed graph $G = (V,E)$
- k source-destination pairs $(s_1, t_1), \ldots, (s_k, t_k)$
- a rate r_i of traffic from s_i to t_i
- for each edge e, a latency function $\ell_e(\cdot)$
  - assumed continuous and nondecreasing

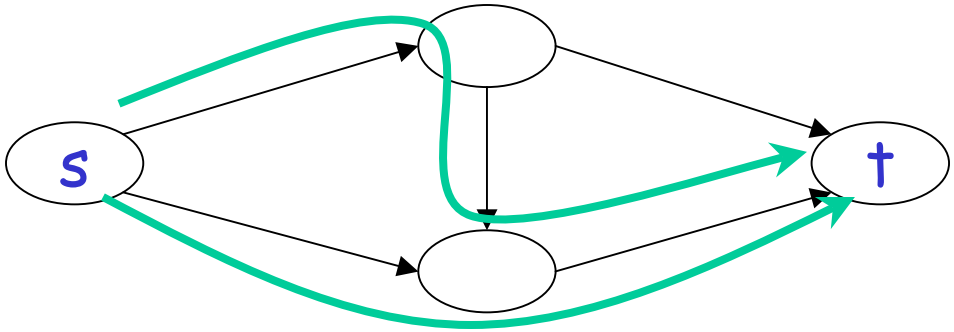
Example: $(k,r=1)$



Routings of Traffic

Traffic and Flows:

- f_P = amount of traffic routed on s_i-t_i path P
- flow vector f ⇔ routing of traffic



Selfish routing: what flows arise as the routes chosen by many noncooperative agents?

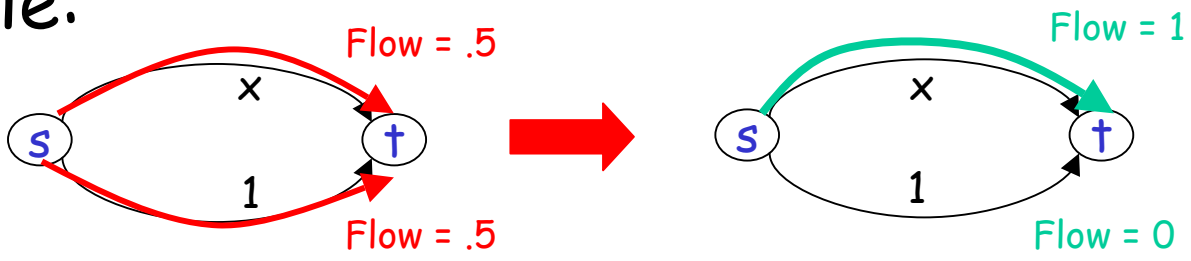
Nash Flows

Some assumptions:

- agents small relative to network
- want to minimize personal latency

Def: A flow is at Nash equilibrium (or is a Nash flow) if all flow is routed on min-latency paths [given current edge congestion]

Example:

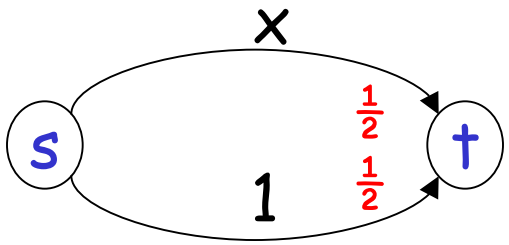


Some History

- traffic model, definition of Nash flows given by [Wardrop 52]
  - historically called user-optimal/user equilibrium
- Nash flows exist, are (essentially) unique
  - due to [Beckmann et al. 56]
- Nash flows also arise via distributed shortest-path protocols (e.g., OSPF, BGP)
  - as long as latency used for edge weights
  - convergence studied in [Tsitsiklis/Bertsekas 86]

The Cost of a Flow

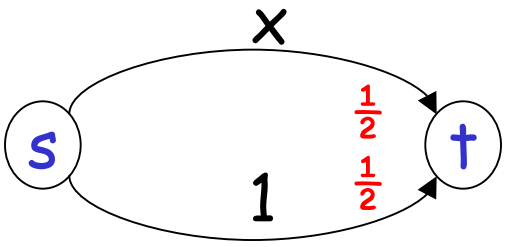
Def: the cost C(f) of flow f = sum of all delays incurred by traffic (aka total latency)



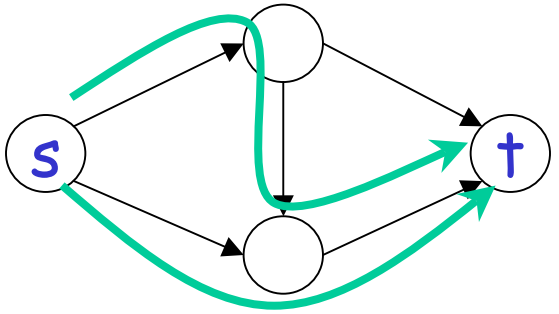
Cost = $\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot 1 = \frac{3}{4}$

The Cost of a Flow

Def: the cost C(f) of flow f = sum of all delays incurred by traffic (aka total latency)



Cost = $\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot 1 = \frac{3}{4}$



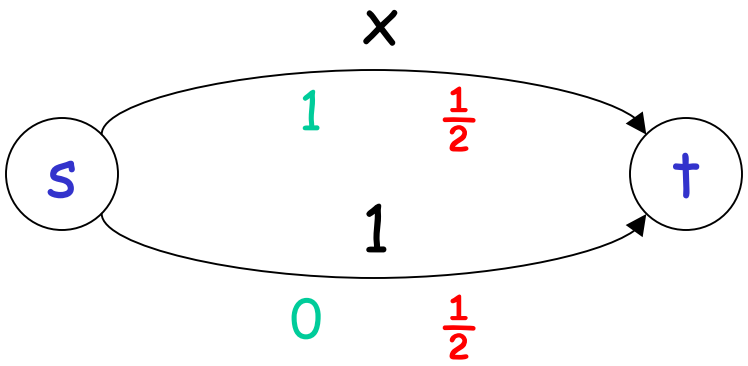
Formally: if $\ell_P(f)$ = sum of latencies of edges of P (w.r.t. the flow f), then:

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

Inefficiency of Nash Flows

Note: Nash flows do not minimize total latency

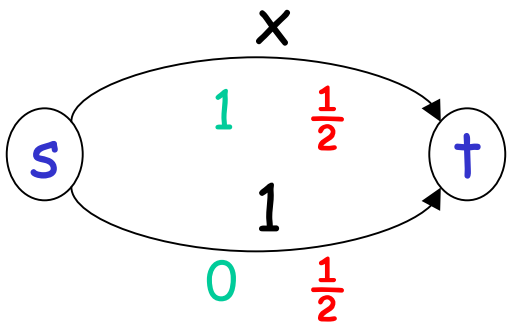
- observed informally by [Pigou 1920]
- lack of coordination leads to inefficiency



- Cost of Nash flow = 1·1 + 0·1 = 1
- Cost of optimal (min-cost) flow = $\frac{1}{2}\cdot\frac{1}{2} + \frac{1}{2}\cdot 1 = \frac{3}{4}$

How Bad Is Selfish Routing?

Pigou's example is simple…



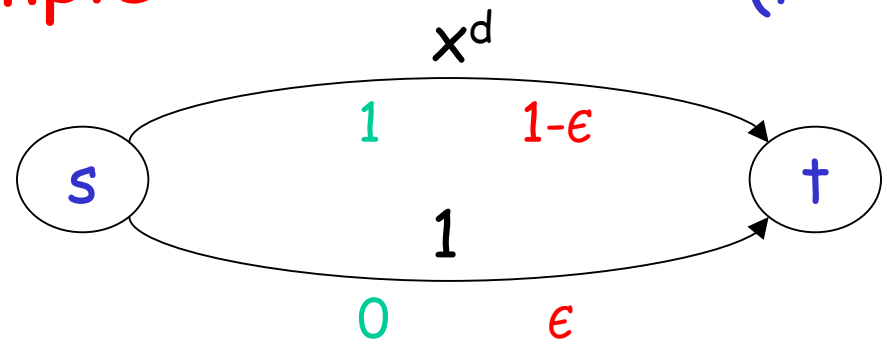
Central question: How inefficient are Nash flows in more realistic networks?

Goal: prove that Nash flows are near-optimal

- want laissez-faire approach to managing networks

The Bad News

Bad Example: (r = 1, d large)



Nash flow has cost 1, min cost ≈ 0

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

- even if latency functions are polynomials

Hardware Offsets Selfishness

Approach #1: use different type of guarantee

Theorem: [Roughgarden/Tardos 00] for every network:

Nash cost at rate r ≤ opt cost at rate 2r

Hardware Offsets Selfishness

Approach #1: use different type of guarantee

Theorem: [Roughgarden/Tardos 00] for every network:

Nash cost at rate r $\leq$ opt cost at rate $2r$

Also: M/M/1 fns ($\ell(x)=1/(u-x)$, u = capacity) $\Rightarrow$

Nash w/capacities $2u$ $\leq$ opt w/capacities u

Linear Latency Functions

Approach #2: restrict class of allowable latency functions

Def: linear latency fn is of form $\ell_e(x)=a_e x+b_e$

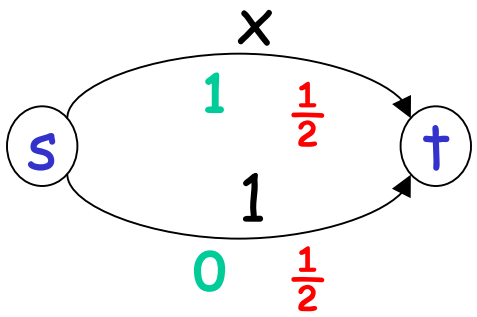
Theorem: [Roughgarden/Tardos 00] for every network with linear latency fns:

$$\text{cost of Nash flow} \leq 4/3 \times \text{cost of opt flow}$$

Sources of Inefficiency

Corollary of previous Theorem:

- For linear latency fns, worst Nash/OPT ratio is realized in a two-link network!



- Cost of Nash = 1
- Cost of OPT = $\frac{3}{4}$

- simple explanation for worst inefficiency
  - confronted w/two routes, selfish users overcongest one of them

Simple Worst-Case Networks

Theorem: [Roughgarden 02] fix any class of latency fns, and the worst Nash/OPT ratio occurs in a two-node, two-link network.

- under mild assumptions (convexity, richness)
- inefficiency of Nash flows always has simple explanation; simple networks are worst examples

Simple Worst-Case Networks

Theorem: [Roughgarden 02] fix any class of latency fns, and the worst Nash/OPT ratio occurs in a two-node, two-link network.

- under mild assumptions (convexity, richness)
- inefficiency of Nash flows always has simple explanation; simple networks are worst examples

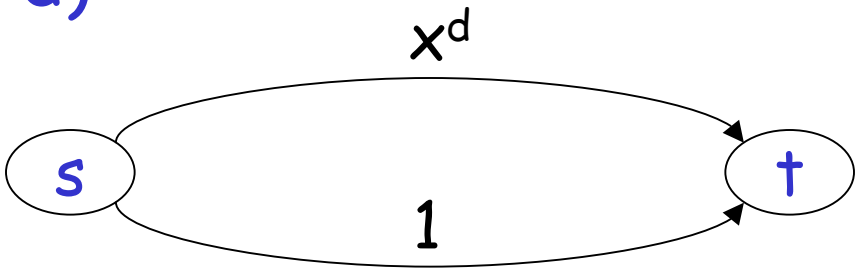
Proof Idea: Nash flows solve a certain minimization problem

- not quite total latency, but close
- electrical current is physical analog

Computing the Price of Anarchy

Application: worst-case examples simple ⇒ worst-case ratio is easy to calculate

Example: polynomials with degree ≤ d, nonnegative coeffs ⇒ price of anarchy $\Theta(d/\log d)$



Hardware Offsets Selfishness

Theorem: [Roughgarden/Tardos 00] for every network:

Nash cost at rate r $\leq$ opt cost at rate $2r$

Corollary: networks with M/M/1 delay fns $\Rightarrow$

Nash w/capacities $2u$ $\leq$ opt w/capacities u

Key Difficulty

Suppose f a Nash flow, f^* an opt flow at twice the rate. Want to show that $C(f^*) \geq C(f)$.

Note: cost of f can be written as

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

Similarly: $C(f^*) = \Sigma_e f_e^* \cdot \ell_e(f_e^*)$

Problem: what is the relation between $\ell_e(f_e)$ and $\ell_e(f_e^*)$?

Key Trick

Idea: lower bound cost of f^* using a different set of latency fns c such that:

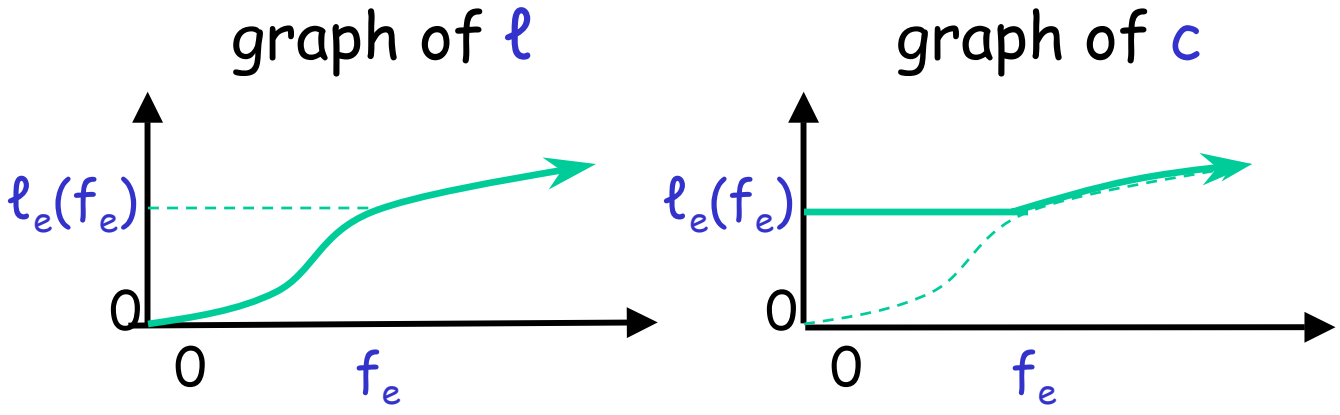
- easy to lower bound cost of f^* w.r.t. latency fns c
- cost of f^* w.r.t. fns c $\approx$ cost of f^* w.r.t. fns ℓ

Key Trick

Idea: lower bound cost of f^* using a different set of latency fns c such that:

- easy to lower bound cost of f^* w.r.t. latency fns c
- cost of f^* w.r.t. fns c $\approx$ cost of f^* w.r.t. fns ℓ

The construction:



Lower Bounding OPT

Assume for simplicity: only one commodity.

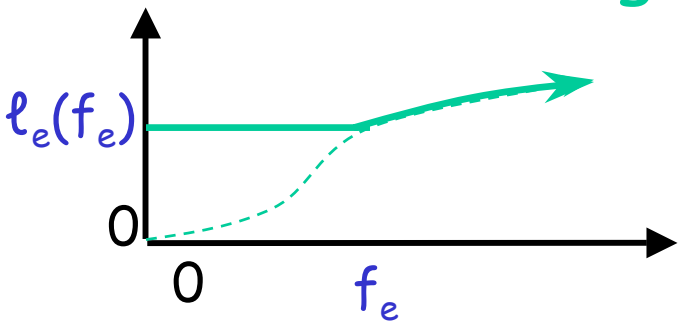
- all traffic in Nash flow has same latency, say L
- cost of Nash flow easy to compute: $C(f) = rL$

Lower Bounding OPT

Assume for simplicity: only one commodity.

- all traffic in Nash flow has same latency, say L
- cost of Nash flow easy to compute: $C(f) = rL$

Key observation: latency of path P w.r.t. latency fns c with no congestion is $\ell_P(f)$



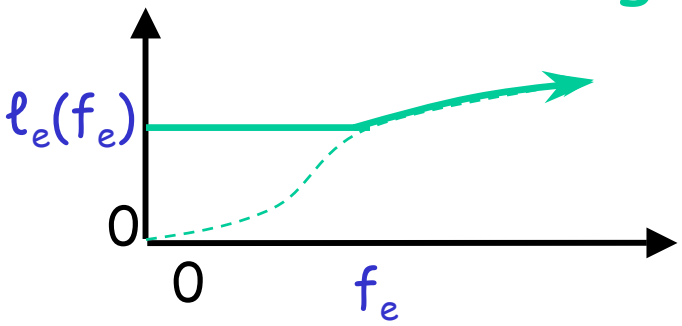
path latency in Nash flow

Lower Bounding OPT

Assume for simplicity: only one commodity.

- all traffic in Nash flow has same latency, say L
- cost of Nash flow easy to compute: $C(f) = rL$

Key observation: latency of path P w.r.t. latency fns c with no congestion is $\ell_P(f)$



path latency in Nash flow

$\Rightarrow$ cost of f^* w.r.t. c is at least $2rL = 2C(f)$

Bounding the Overestimate

So far: cost of f^* w.r.t. c is $\geq 2C(f)$.

Claim: (will finish proof of Thm)

[cost of f^* w.r.t. c] - $C(f^*) \leq C(f)$.

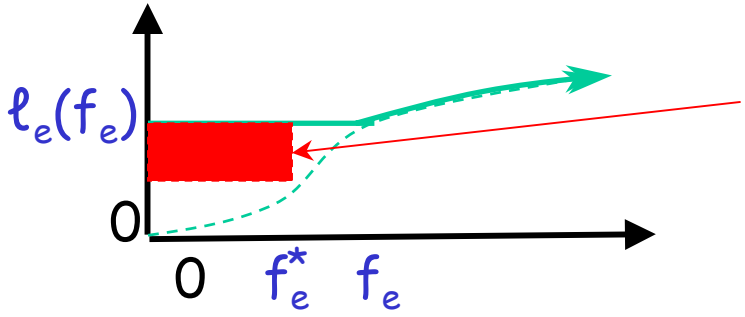
Bounding the Overestimate

So far: cost of f^* w.r.t. c is $\geq 2C(f)$.

Claim: (will finish proof of Thm)

[cost of f^* w.r.t. c] - $C(f^*)$ $\leq$ $C(f)$.

Reason: difference in costs on edge e is



typical value of

$c_e(f_e^*)f_e^* - \ell_e(f_e^*)f_e^*$

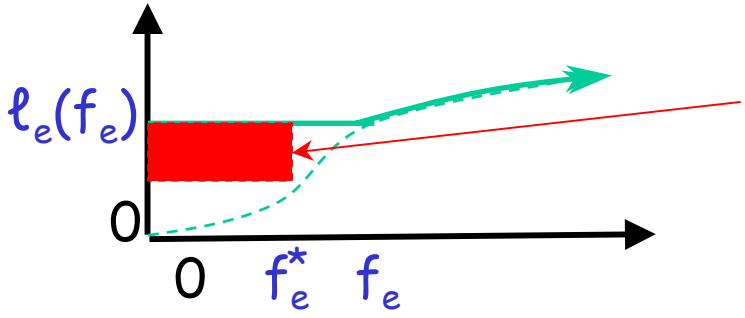
Bounding the Overestimate

So far: cost of f^* w.r.t. c is $\geq 2C(f)$.

Claim: (will finish proof of Thm)

[cost of f^* w.r.t. c] - $C(f^*)$ $\leq$ $C(f)$.

Reason: difference in costs on edge e is



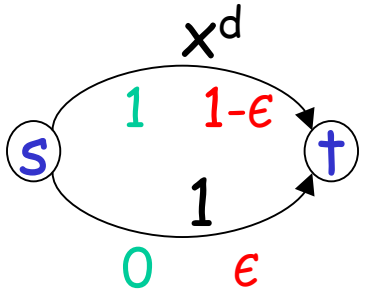
typical value of
$c_e(f_e^*)f_e^* - \ell_e(f_e^*)f_e^*$

$\Rightarrow$ $c_e(f_e^*)f_e^* - \ell_e(f_e^*)f_e^* \leq \ell_e(f_e)f_e$ ← sum over edges to get Claim

Summary

Goal: prove that loss in network performance due to selfish routing is not too large.

Problem: a Nash flow can cost far more than an optimal flow.



Solutions:

- compare Nash to opt flow with extra traffic
- restrict class of allowable edge latency functions, obtain bounded price of anarchy

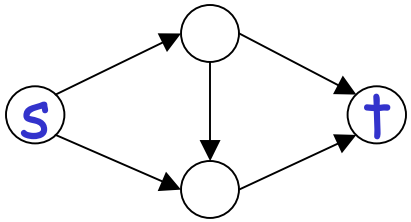
Coping with Selfishness

Goal: design/manage networks so that selfish routing "not too bad"

$\Rightarrow$ adds algorithmic dimension

Approach #1: Network design

- want to avoid Braess's Paradox



Results: [Roughgarden FOCS '01]

- Braess's Paradox can be arbitrarily severe in larger networks, hard to efficiently detect
- also [Lin/Roughgarden/Tardos, in prep]

Coping with Selfishness

Approach #2: Stackelberg routing

- some traffic routed centrally, selfish users react to congestion accordingly
- [Roughgarden STOC '01]: Stackelberg routing can dramatically improve over the Nash flow

Approach #3: Edge pricing

- use economic incentives (taxes) to influence selfish behavior
- [Cole/Dodis/Roughgarden EC '03 + STOC '03]: explore this idea for selfish routing

Future Research

- Explore other game-theoretic environments using an approximation framework
  - [Czumaj/Krysta/Voecking STOC '02], [Vetta FOCS '02], etc.
- Approximation algorithms for network design
  - also interesting without game-theoretic constraints
  - [Kumar/Gupta/Roughgarden FOCS '02]
  - [Gupta/Kumar/Roughgarden STOC '03]
- Algorithms for key game-theoretic concepts
  - Nash/market equilibria (e.g., [Devanur et al FOCS '02])

Extensions

Fact: positive results continue to hold for:

- approximate Nash flows [RT00]
  - users route on approximately min-latency paths
- finitely many agents, splittable flow [RT00]
  - weakens assumption that agents are small
- "nonatomic congestion games", games without combinatorial structure of a network [RT02]