

---

# Network Transport Layer: Network Resource Allocation Framework

Qiao Xiang

<https://qiaoxiang.me/courses/cnns-xmuf22/index.shtml>

11/10/2022

# Admin.

---

- Lab assignment 3 due date extended to 8pm Nov. 13
- DO NOT procrastinate future lab assignments any more!
  
- Exam this afternoon
  - Do not forget to bring your 1-page cheat sheet ☺

# Outline

---

- Admin and recap
- Transport congestion control
  - what is congestion (cost of congestion)
  - basic congestion control alg.
  - TCP/Reno congestion control
  - TCP Cubic
  - TCP/Vegas
  - network wide resource allocation
    - general framework
    - objective function: axiom derivation of network-wide objective function
    - algorithm: general distributed algorithm framework
    - application: TCP/Reno TCP/Vegas revisited

## Recap: TCP/Reno Throughput Modeling

---

$$\Delta W = \begin{cases} \frac{1}{W} & \text{if the packet is not lost} \\ -\frac{W}{2} & \text{if packet is lost} \end{cases}$$

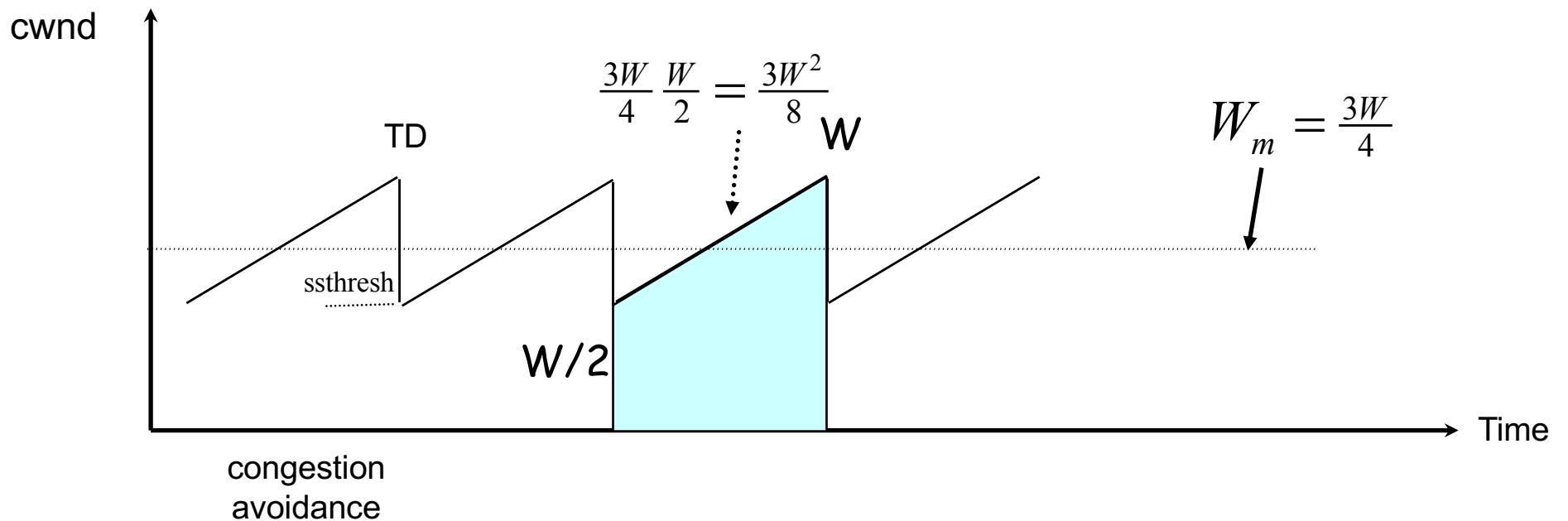
$$\text{mean of } \Delta W = (1-p)\frac{1}{W} + p(-\frac{W}{2}) = 0$$

$$\Rightarrow \text{mean of } W = \sqrt{\frac{2(1-p)}{p}} \approx \frac{1.4}{\sqrt{p}}, \text{ when } p \text{ is small}$$

$$\Rightarrow \text{throughput} \approx \frac{1.4S}{RTT\sqrt{p}}, \text{ when } p \text{ is small}$$

- 4 This is called the TCP throughput sqrt of loss rate law.

# Recap: TCP/Reno Throughput Modeling



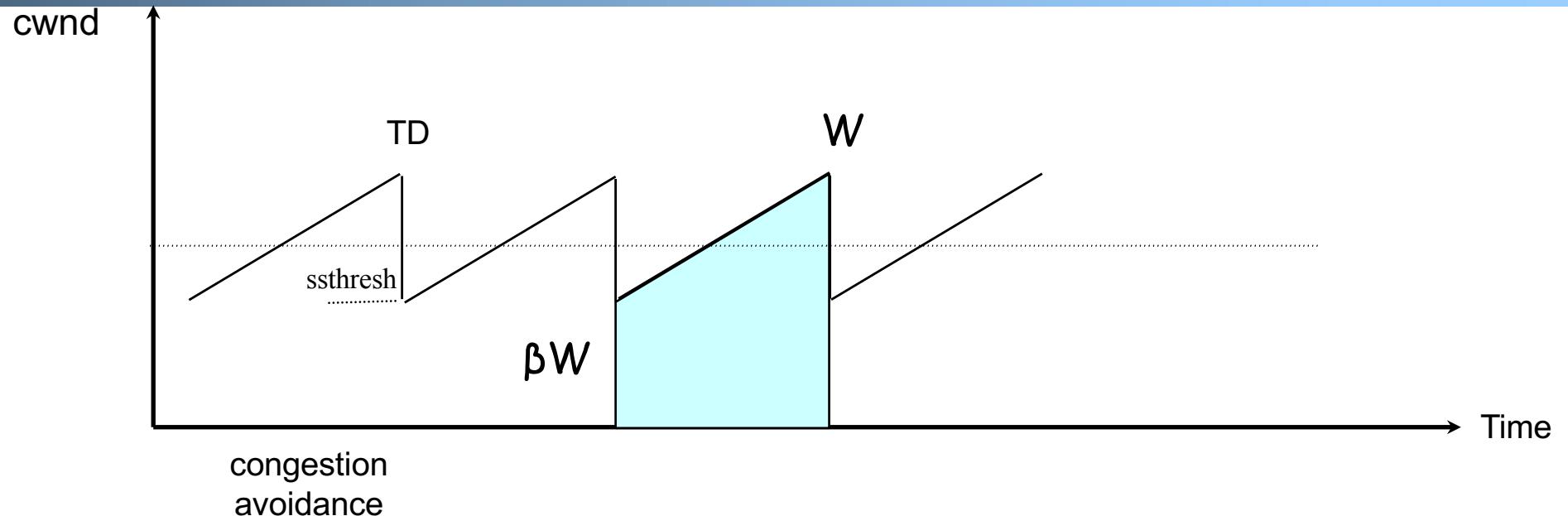
$$\text{Total packets sent per cycle} = (W/2 + W)/2 * W/2 = 3W^2/8$$

$$\text{Assume one loss per cycle} \Rightarrow p = 1/(3W^2/8) = 8/(3W^2)$$

$$\Rightarrow W = \frac{\sqrt{8/3}}{\sqrt{p}} = \frac{1.6}{\sqrt{p}}$$

$$\Rightarrow \text{throughput} = \frac{S}{RTT} \frac{3}{4} \frac{1.6}{\sqrt{p}} = \boxed{\frac{1.2S}{RTT \sqrt{p}}} \quad 5$$

## Recap: Generic AIMD and TCP Friendliness



$$\text{Total packets sent per cycle} = \frac{\beta W + W}{2} \cdot \frac{(1-\beta)W}{\alpha} = \frac{(1-\beta)(1+\beta)}{2\alpha} W^2$$

$$\text{Assume one loss per cycle } p = \frac{2\alpha}{(1-\beta)(1+\beta)W^2} \quad W = \sqrt{\frac{2\alpha}{(1-\beta)(1+\beta)p}}$$

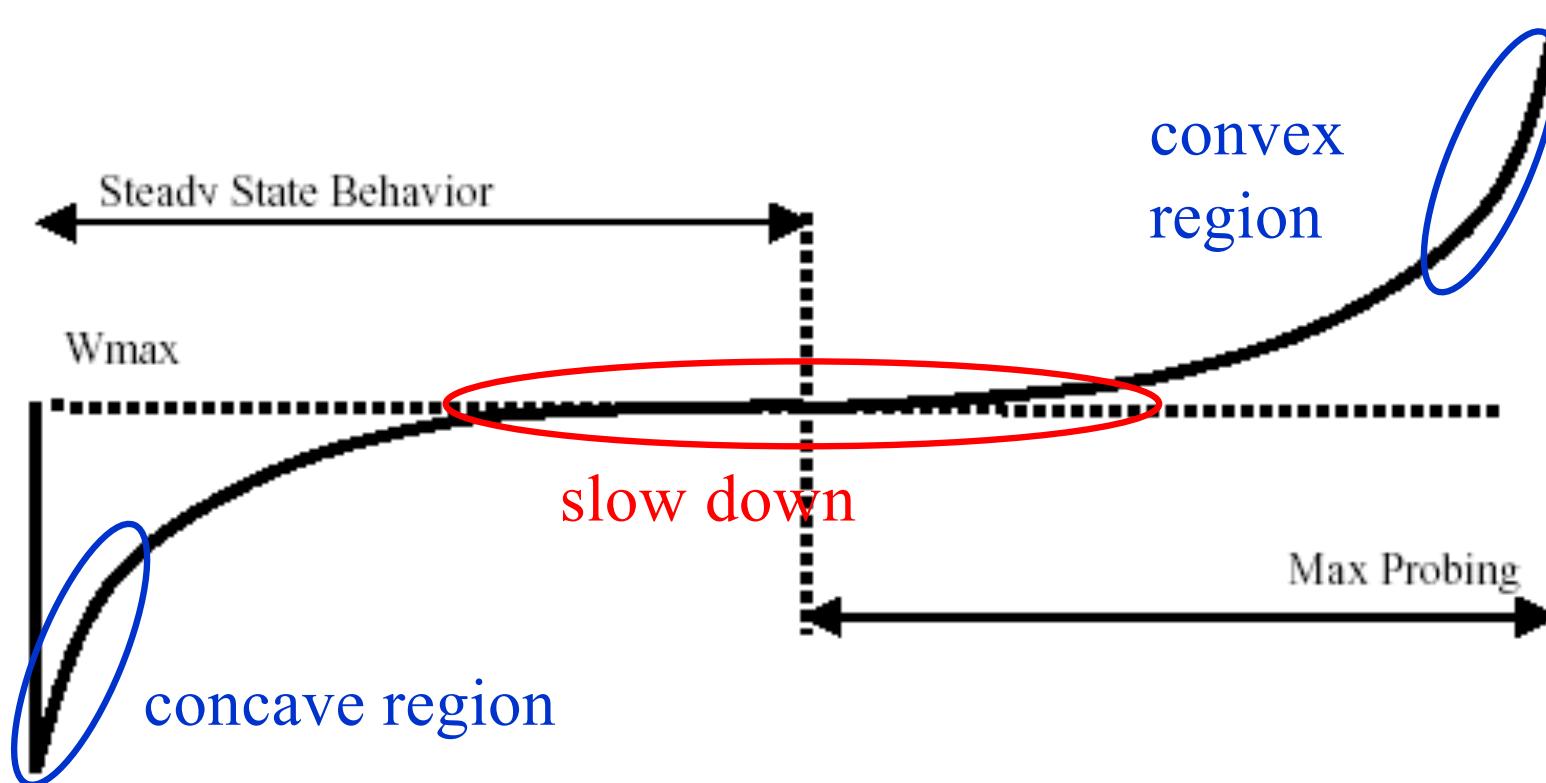
$$\text{tput} = \frac{W_m S}{RTT} = \frac{S}{RTT} \cdot \frac{(1+\beta)W}{2} = \frac{S}{RTT} \sqrt{\frac{\alpha(1+\beta)}{2(1-\beta)p}}$$

$$\text{TCP friendly} \Rightarrow \alpha = 3 \frac{1-\beta}{1+\beta}$$

$$\beta' = 1 - \beta$$

## Recap: TCP Cubic

$$W_{tcp(t)} = W_{max} \beta' + 3 \frac{1-\beta'}{1+\beta'} \frac{t}{RTT}$$



$$W_{cubic} = C(t - K)^3 + W_{max} \quad K = \sqrt[3]{W_{max} \beta / C}$$

where  $C$  is a scaling factor,  $t$  is the elapsed time from the last window reduction, and  $\beta$  is a constant multiplication decrease factor

# Outline

---

- Admin and recap
- Transport congestion control
  - what is congestion (cost of congestion)
  - basic congestion control alg.
  - TCP/Reno congestion control
  - TCP Cubic
  - TCP/Vegas
  - network wide resource allocation
    - general framework

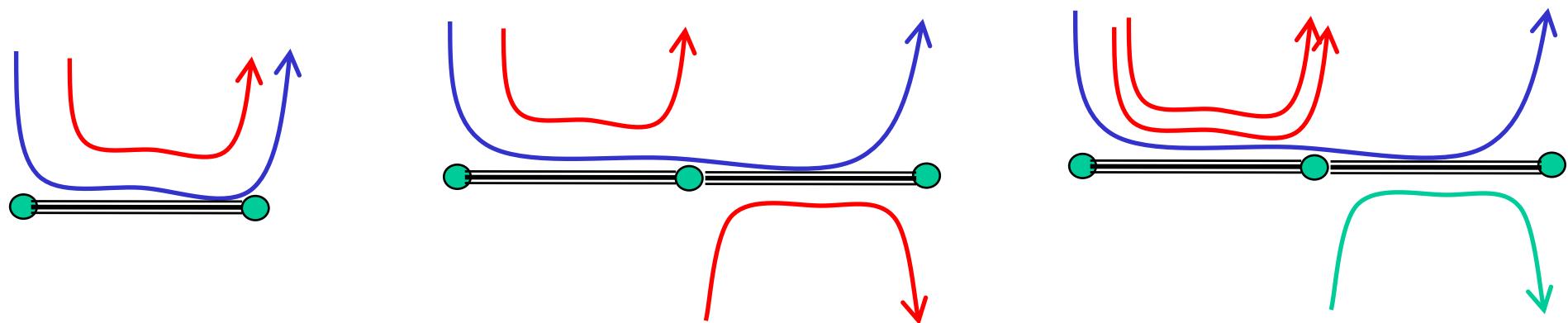
# Motivation

---

- So far our discussion is implicitly on a network with a single bottleneck link; this simplifies design and analysis:
  - efficiency/optimality (high utilization)
    - fully utilize the bandwidth of the link
  - fairness (resource sharing)
    - each flow receives an *equal* share of the link's bandwidth

# Network Resource Allocation

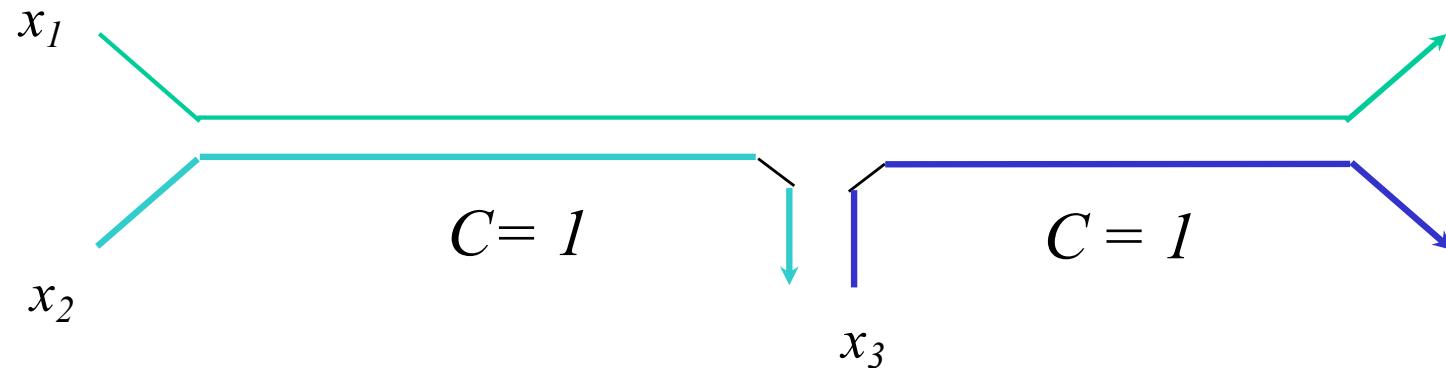
- It is important to understand and design protocols for a general network topology
  - how **will** TCP allocate resource in a **general topology**?
  - how **should** resource be allocated in a **general topology**?



# Example: TCP/Reno Rates

■ Rates:

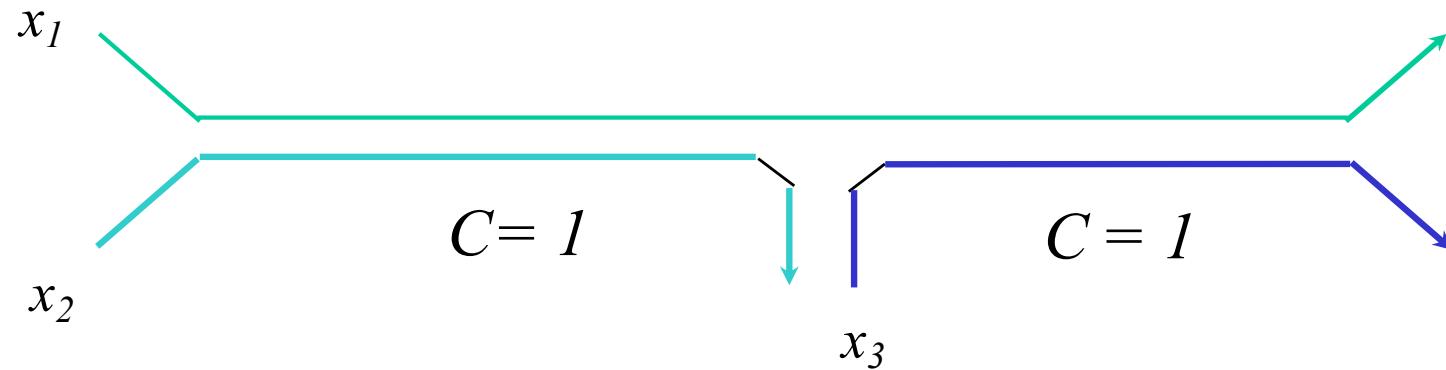
$$x_1 = \frac{1}{1+2\sqrt{2}} = 0.26$$
$$x_2 = x_3 = 0.74$$



# Example: TCP/Vegas Rates

---

■ Rates :  $x_1 = 1/3$   
 $x_2 = x_3 = 2/3$



## Example: Max-min Fairness



- Max-min fairness: maximizes the throughput of the flow receiving the minimum (of resources)
  - Justification: John Rawls, *A Theory of Justice* (1971)
    - [http://en.wikipedia.org/wiki/John\\_Rawls](http://en.wikipedia.org/wiki/John_Rawls)
  - This is a resource allocation scheme used in ATM and some other network resource allocation proposals

# Example: Max-Min

$$\max_{x_f \geq 0}$$

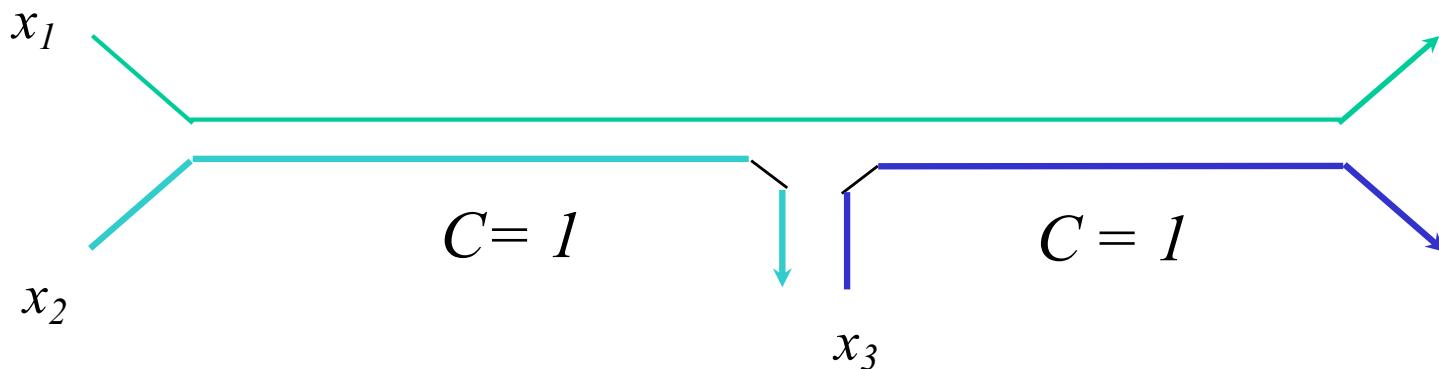
$$\min\{x_f\}$$

subject to

$$x_1 + x_2 \leq 1$$

$$x_1 + x_3 \leq 1$$

■ Rates:  $x_1 = x_2 = x_3 = 1/2$



## Framework: Network Resource Allocation Using Utility Functions

- A set of flows  $F$
- Let  $x_f$  be the rate of flow  $f$ , and the utility to flow  $f$  is  $U_f(x_f)$ .
- Maximize aggregate utility, subject to capacity constraints

$$\begin{array}{ll}\max & \sum_{f \in F} U_f(x_f) \\ \text{subject to} & \sum_{f: f \text{ uses link } l} x_f \leq c_l \text{ for any link } l \\ \text{over} & x \geq 0\end{array}$$

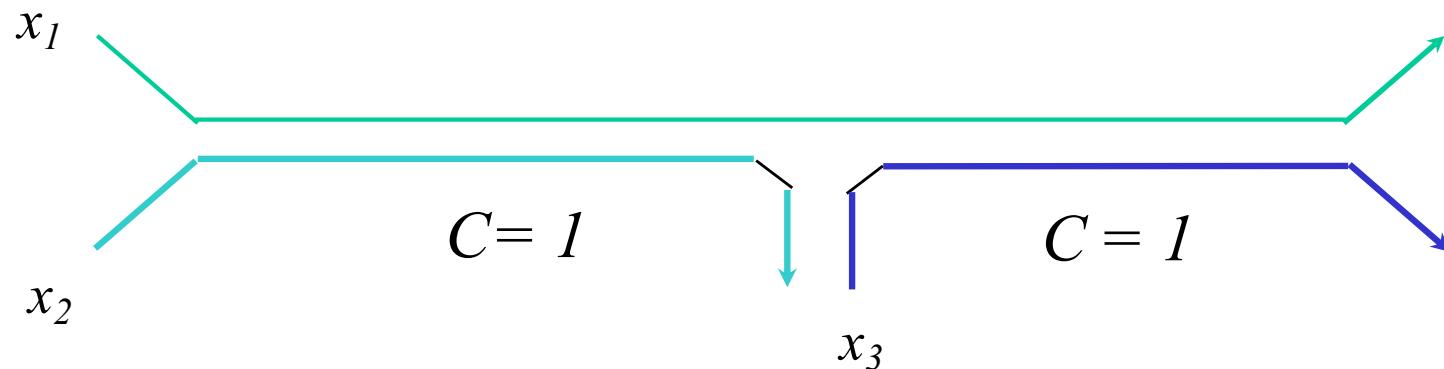
# Example: Maximize Throughput

$$\max_{x_f \geq 0} \quad \sum_f x_f \quad U_f(x_f) = xf$$

subject to

$$x_1 + x_2 \leq 1$$
$$x_1 + x_3 \leq 1$$

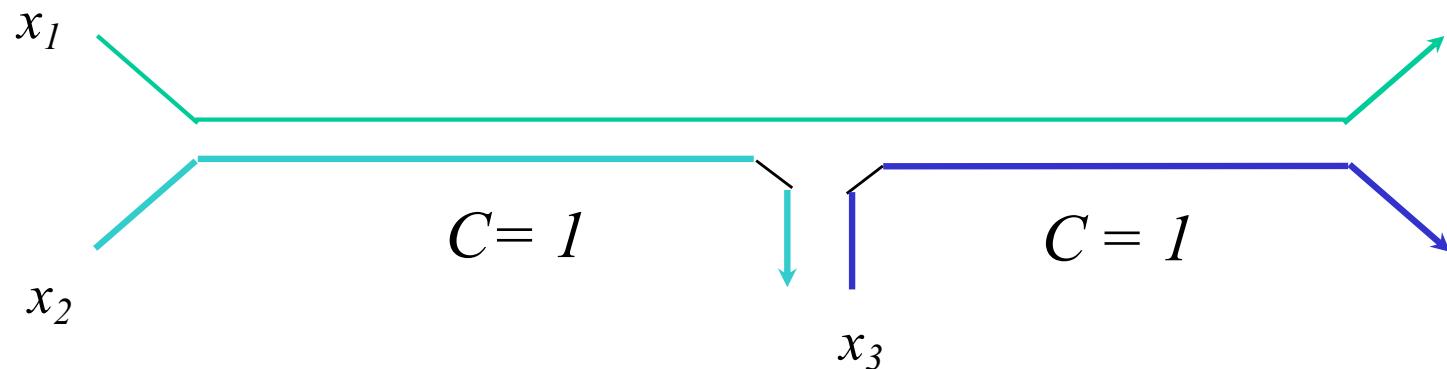
■ Optimal:  $x_1 = 0$   
 $x_2 = x_3 = 1$



# Example: Proportional Fairness

$$\begin{array}{ll} \max_{x_f \geq 0} & \sum_f \log x_f \\ \text{subject to} & x_1 + x_2 \leq 1 \\ & x_1 + x_3 \leq 1 \end{array}$$
$$U_f(x_f) = \log(x_f)$$

■ Optimal:  $x_1 = 1/3$   
 $x_2 = x_3 = 2/3$



## Example 3: a “Funny” Utility Function

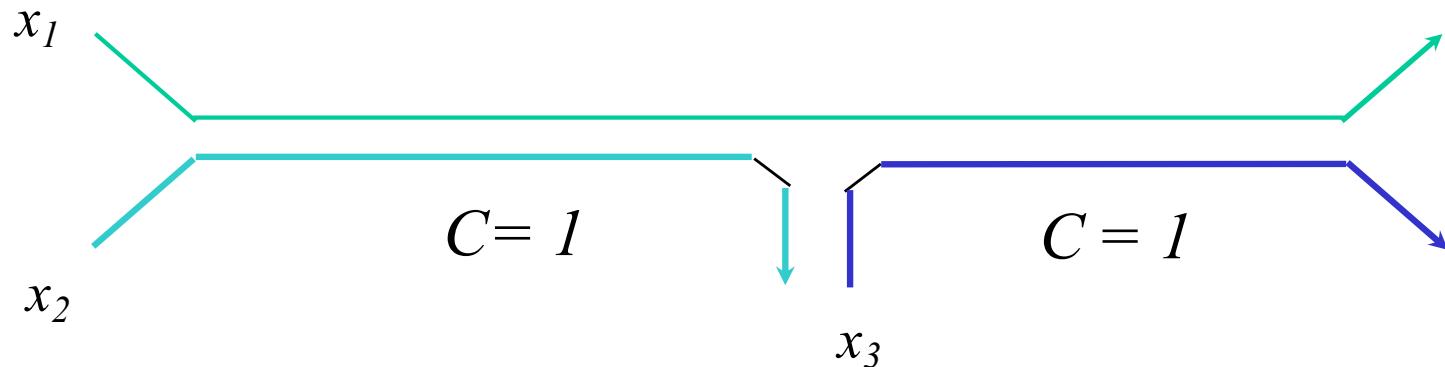
$$\max_{x_f \geq 0} -\frac{1}{4x_1} - \frac{1}{x_2} - \frac{1}{x_3}$$

subject to  $x_1 + x_2 \leq 1$

$$x_1 + x_3 \leq 1$$

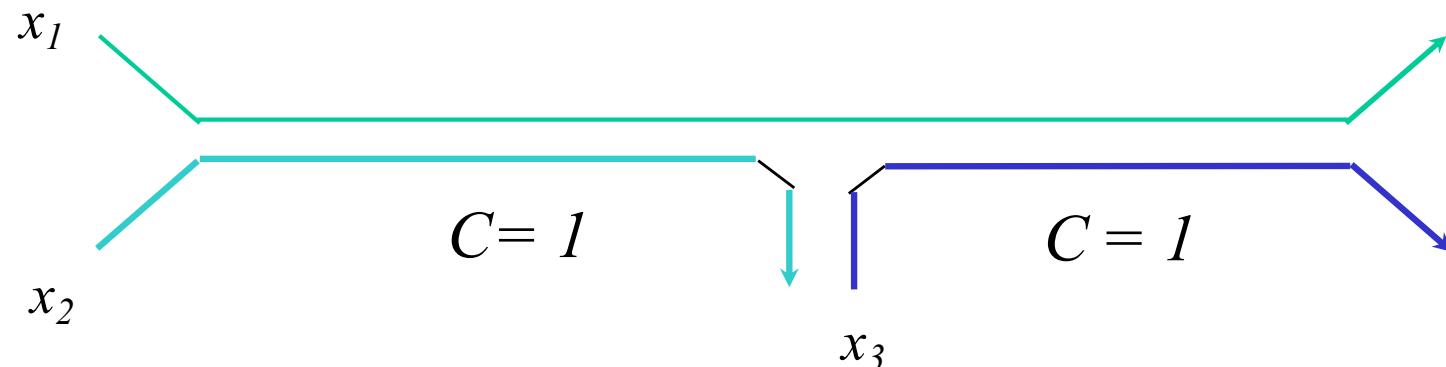
$$U_f(x_f) = -\frac{1}{RTT^2 x_f}$$

■ Optimal:  $x_1 = \frac{1}{1+2\sqrt{2}} = 0.26$   
 $x_2 = x_3 = 0.74$



# Summary: Allocations

Objective	Allocation ( $x_1, x_2, x_3$ )		
TCP/Reno	0.26	0.74	0.74
TCP/Vegas	1/3	2/3	2/3
Max Throughput	0	1	1
Max-min	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Max sum $\log(x)$	1/3	2/3	2/3
Max sum of $-1/(\text{RTT}^2 \cdot x)$	0.26	0.74	0.74



# Questions

$$\begin{aligned}
 & \max && \sum_{f \in F} U_f(x_f) \\
 & \text{subject to} && \sum_{\substack{f: f \text{ uses link } l}} x_f \leq c_l \text{ for any link } l \\
 & \text{over} && x \geq 0
 \end{aligned}$$

- Forward engineering: systematically
  - design objective function
  - design distributed alg to achieve objective
- Science/reverse engineering: what do TCP/Reno, TCP/Vegas achieve?

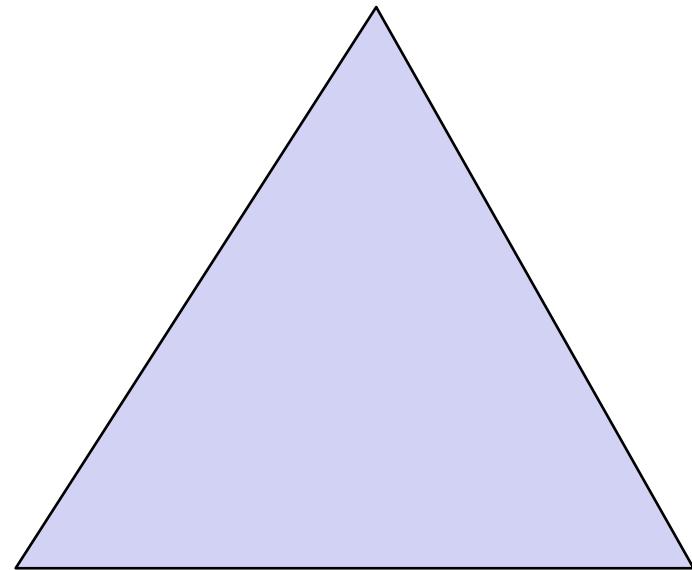
Objective	Allocation ( $x_1, x_2, x_3$ )		
TCP/Reno	0.26	0.74	0.74
TCP/Vegas	1/3	2/3	2/3
Max throughput	0	1	1
Max-min	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Max sum log( $x$ )	1/3	2/3	2/3
Max sum of $-1/(\text{RTT}^2 x)$	0.26	0.74	0.74

# Outline

---

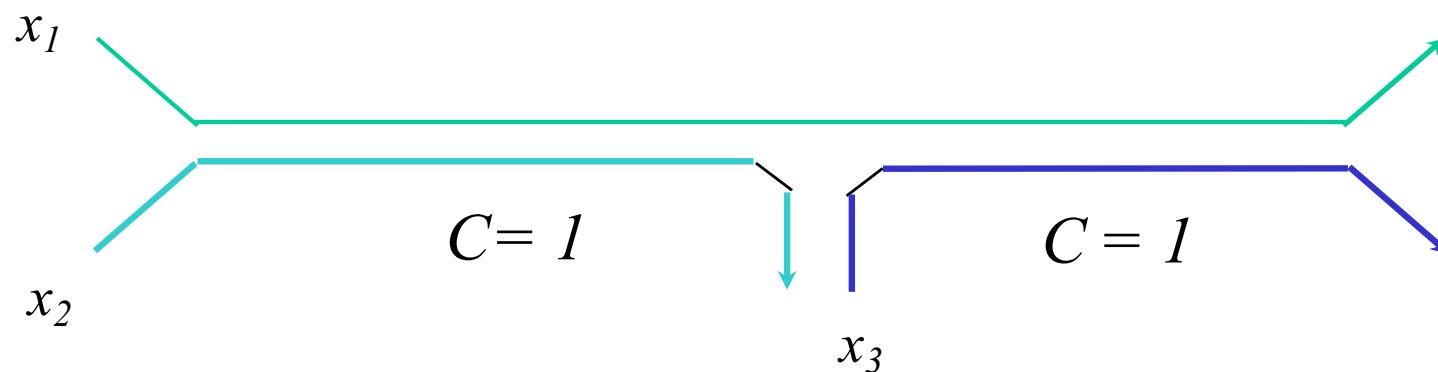
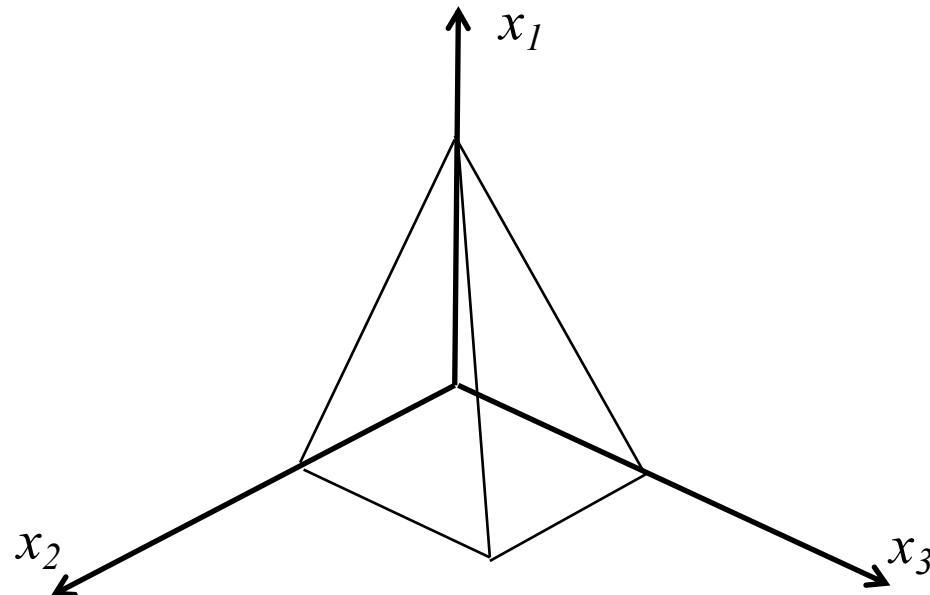
- Admin and recap
- Transport congestion control
  - what is congestion (cost of congestion)
  - basic congestion control alg.
  - TCP/Reno congestion control
  - TCP Cubic
  - TCP/Vegas
  - network wide resource allocation
    - general framework
    - objective function: an example of an axiom derivation of network-wide objective function

# Network Bandwidth Allocation Using Nash Bargain Solution (NBS)



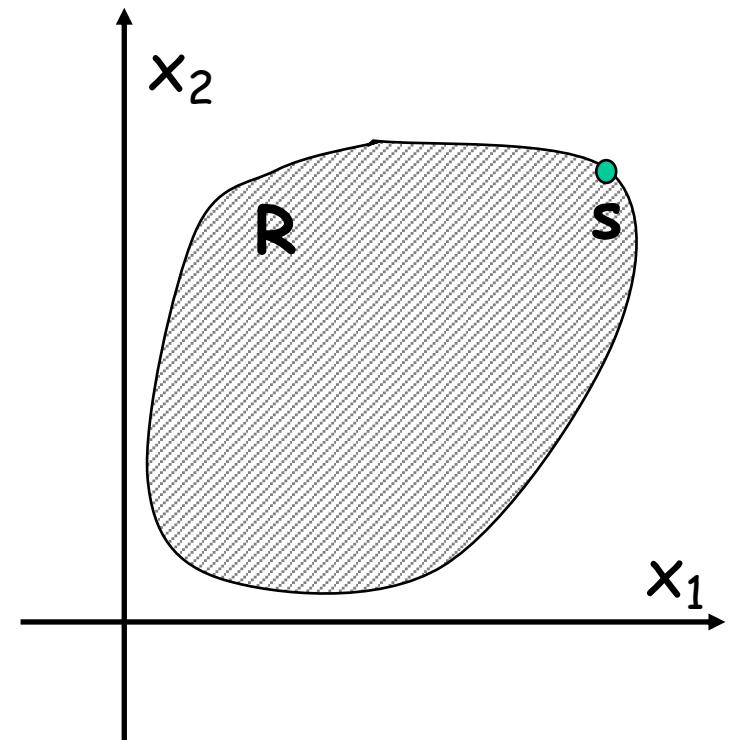
- High level picture
  - given the feasible set of bandwidth allocation, we want to pick an allocation point that is efficient and fair
- The determination of the allocation point should be based on "first principles" (axioms)

# Network Bandwidth Allocation: Feasible Region



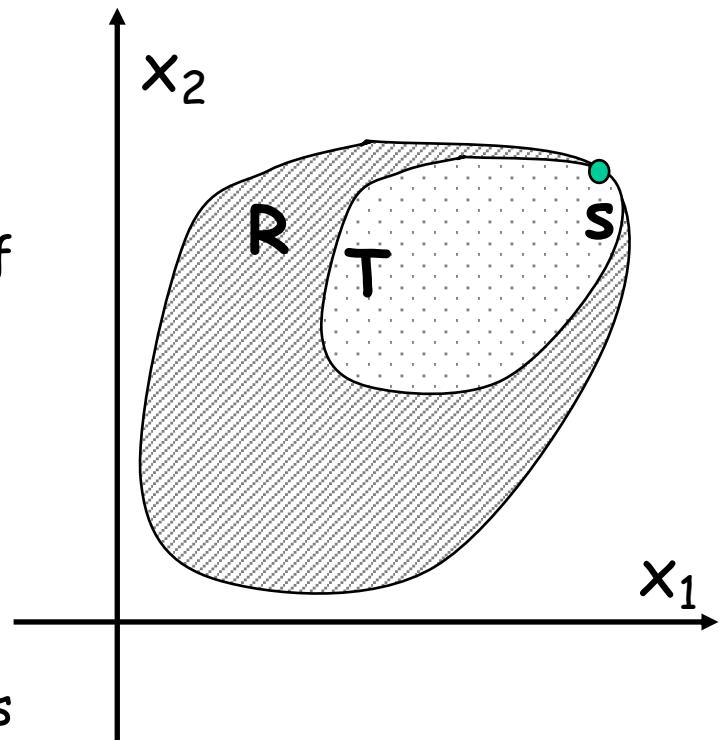
# Nash Bargain Solution (NBS)

- Assume a finite, convex feasible set in the first quadrant
- Axioms



# Nash Bargain Solution (NBS)

- Assume a finite, convex feasible set in the first quadrant
- Axioms
  - Pareto optimality
    - impossibility of increasing the rate of one user without decreasing the rate of another
  - symmetry
    - a symmetric feasible set yields a symmetric outcome
  - invariance of linear transformation
    - the allocation must be invariant to linear transformations of users' rates
  - independence of irrelevant alternatives
    - assume  $s$  is an allocation when feasible set is  $R$ ,  $s \in T \subset R$ , then  $s$  is also an allocation when the feasible set is  $T$



# Nash Bargain Solution (NBS)

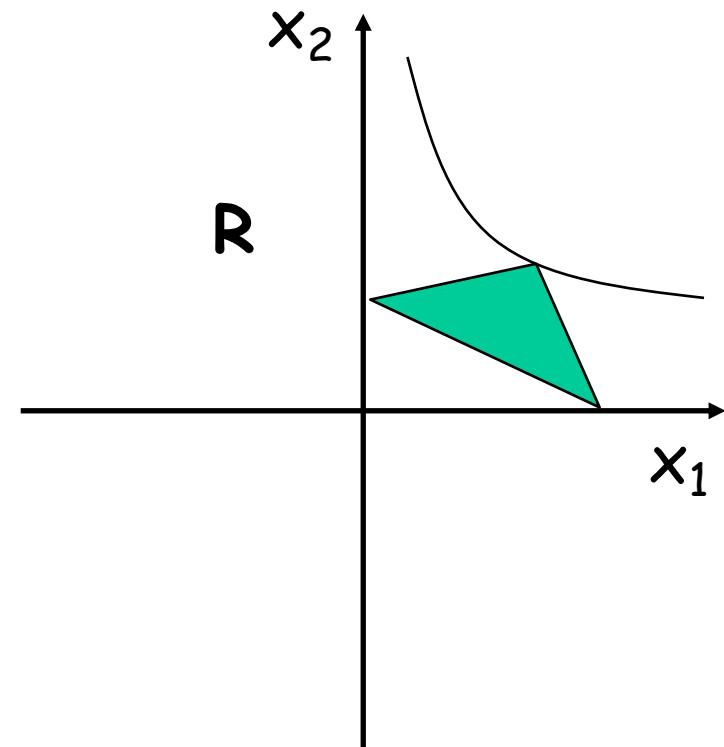
- Surprising result by John Nash (1951)

- the rate allocation point is the feasible point which maximizes

$$x_1 x_2 \cdots x_F$$

- This is equivalent to maximize

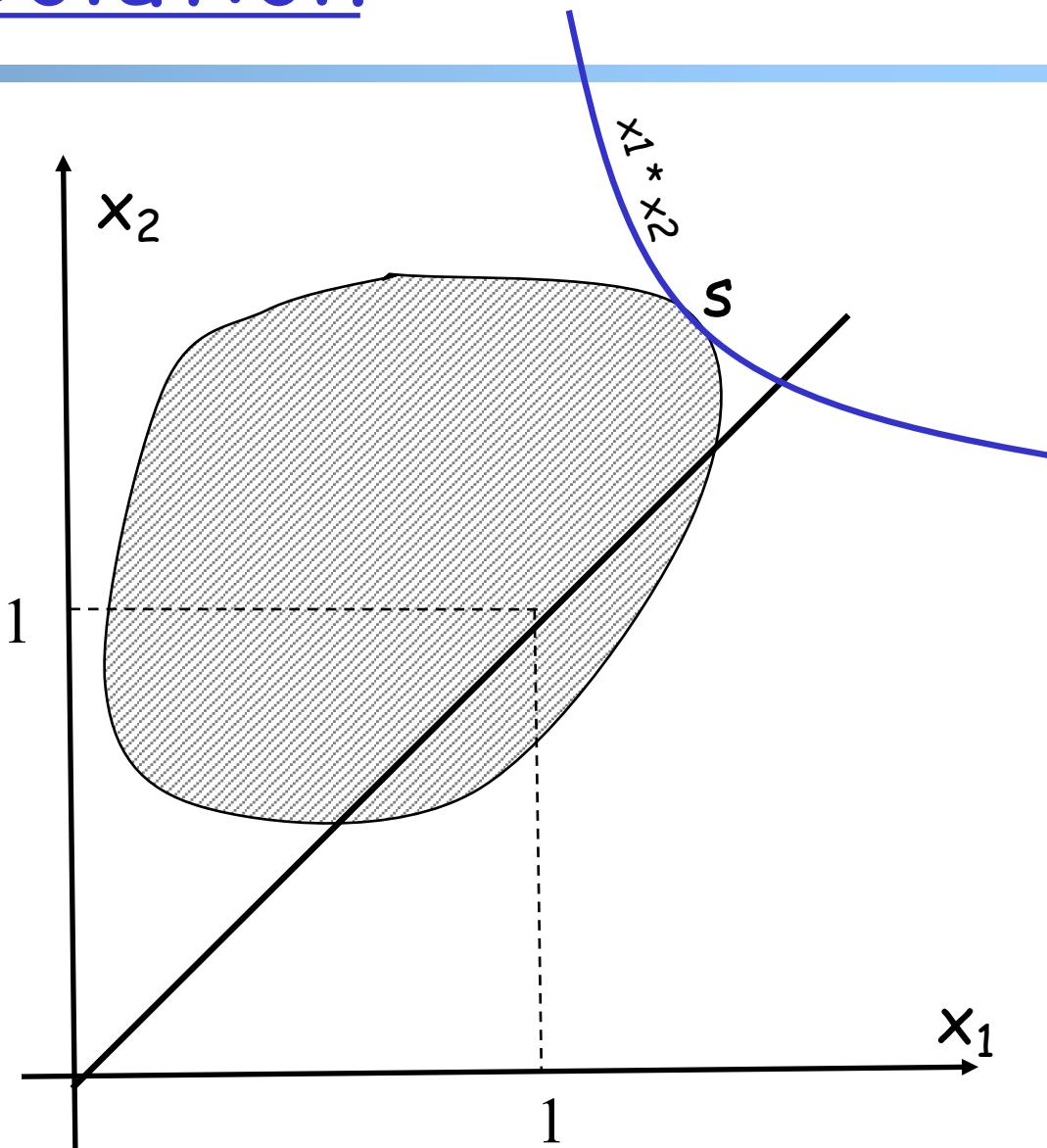
$$\sum_f \log(x_f)$$



- In other words, assume each flow  $f$  has utility function  $\log(x_f)$
- I will give a proof for  $F = 2$ 
  - think about  $F > 2$

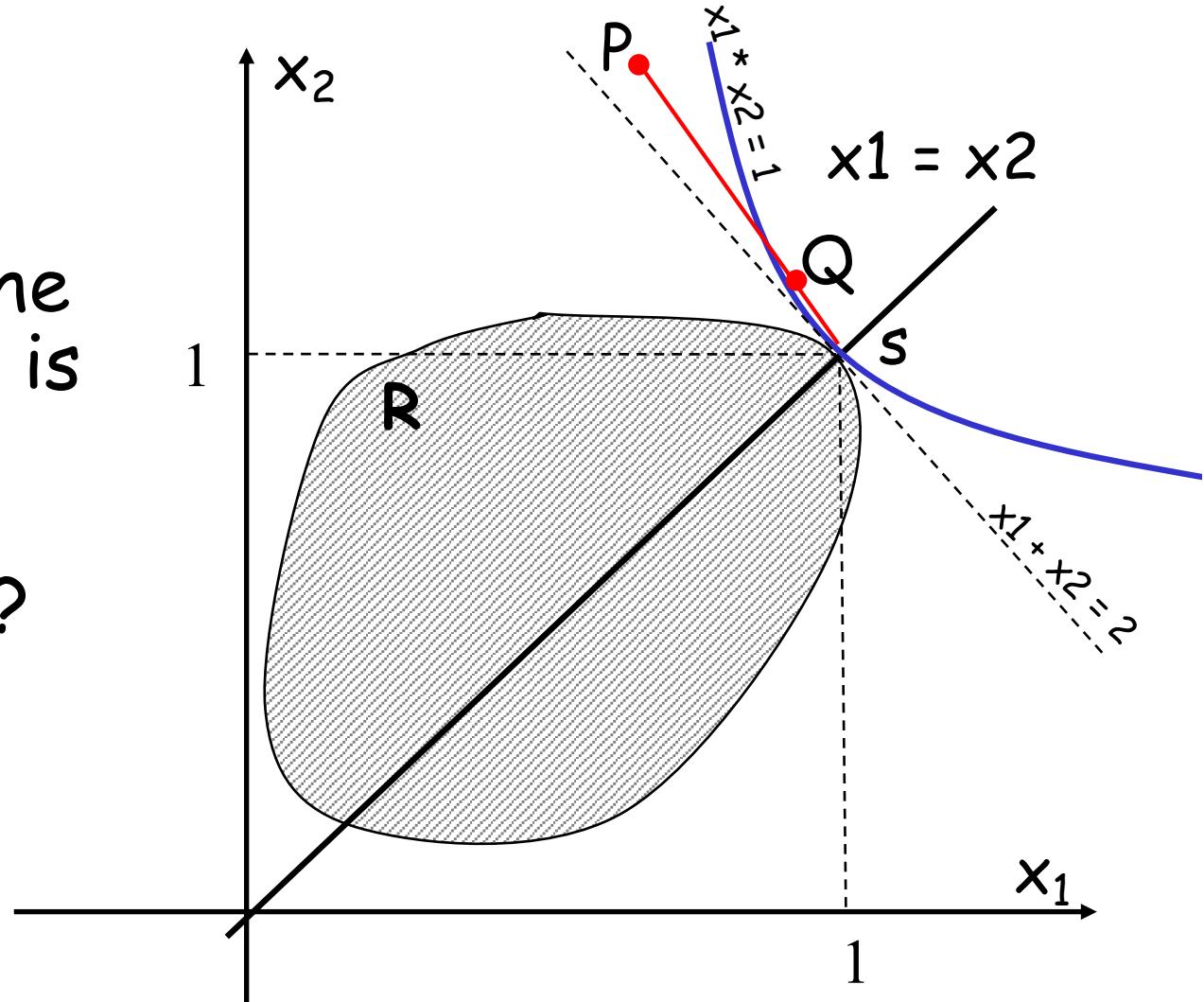
# Nash Bargain Solution

- Assume  $s$  is the feasible point which maximizes  $x_1 * x_2$
  
- Scale the feasible set so that  $s$  is at  $(1, 1)$ 
  - how?



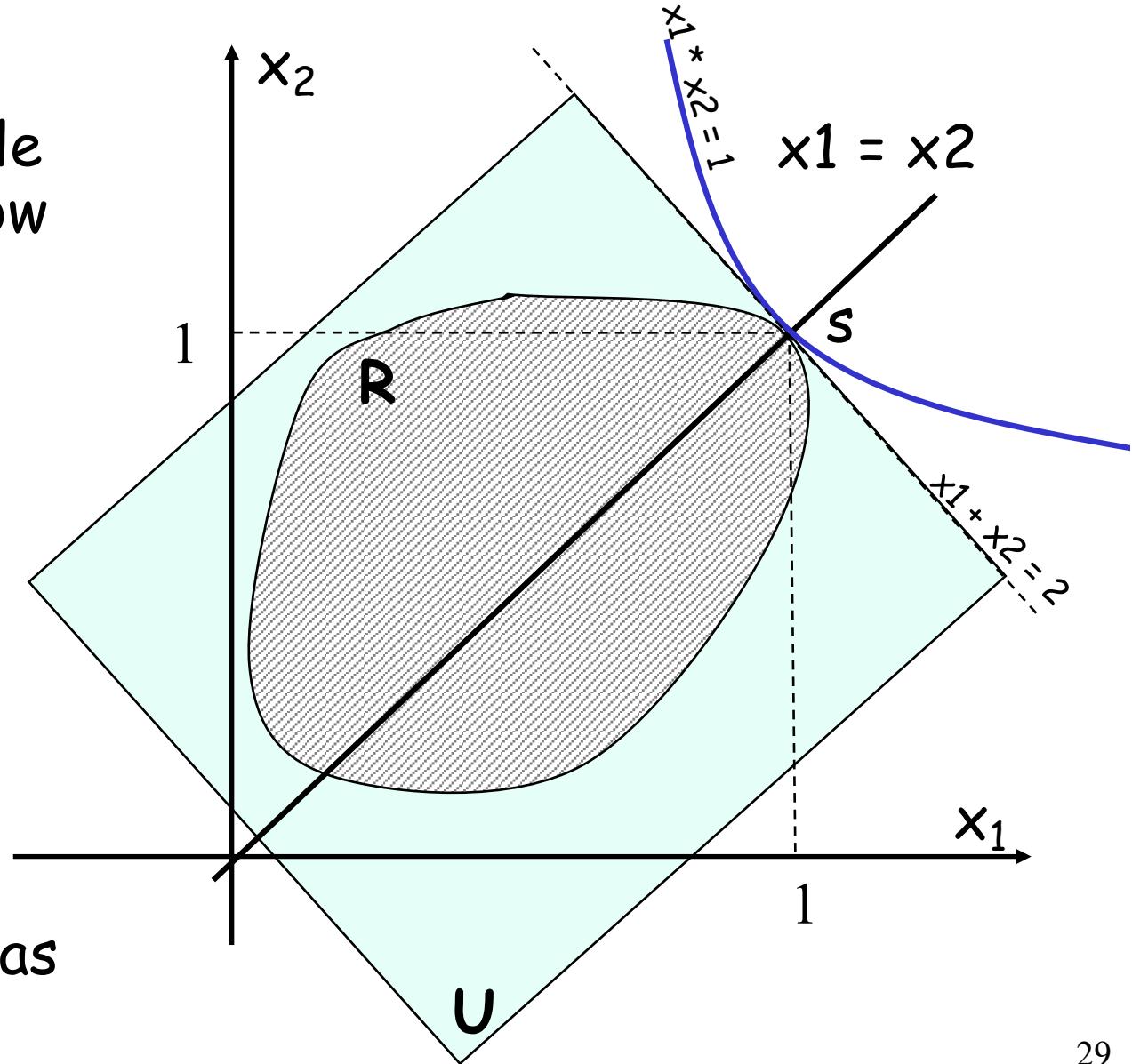
# Nash Bargain Solution

Question: after the transformation, is there any feasible point with  $x_1 + x_2 > 2$ ?



# Nash Bargain Solution

- Consider the symmetric rectangle  $U$  containing the now feasible set
  - > According to symmetry and Pareto,  $s$  is the allocation when feasible set is  $U$
- According to independence of irrelevant alternatives, the allocation of  $R$  is  $s$  as well.



## NBS $\Leftrightarrow$ Proportional Fairness

---

- Allocation is proportionally fair if for any other allocation, aggregate of proportional changes is non-positive, e.g. if  $x_f$  is a proportional-fair allocation, and  $y_f$  is any other feasible allocation, then require

$$\sum_f \frac{y_f - x_f}{x_f} \leq 0$$

# Questions to Think

---

- Vary the axioms and see if you can derive any objective functions

# Outline

---

- Admin and recap
- Transport congestion control
  - what is congestion (cost of congestion)
  - basic congestion control alg.
  - TCP/Reno congestion control
  - TCP Cubic
  - TCP/Vegas
  - network wide resource allocation
    - general framework
    - objective function: an example axiom derivation of network-wide objective function
    - algorithm: general distributed algorithm framework
    - application: TCP/Reno TCP/Vegas revisited

## Recall: Resource Allocation Framework

### □ The Resource-Allocation Problem:

$$\begin{array}{ll}\max & \sum_{f \in F} U_f(x_f) \\ \text{subject to} & \sum_{\substack{f: f \text{ uses link } l}} x_f \leq c_l \text{ for any link } l \\ \text{over} & x \geq 0\end{array}$$

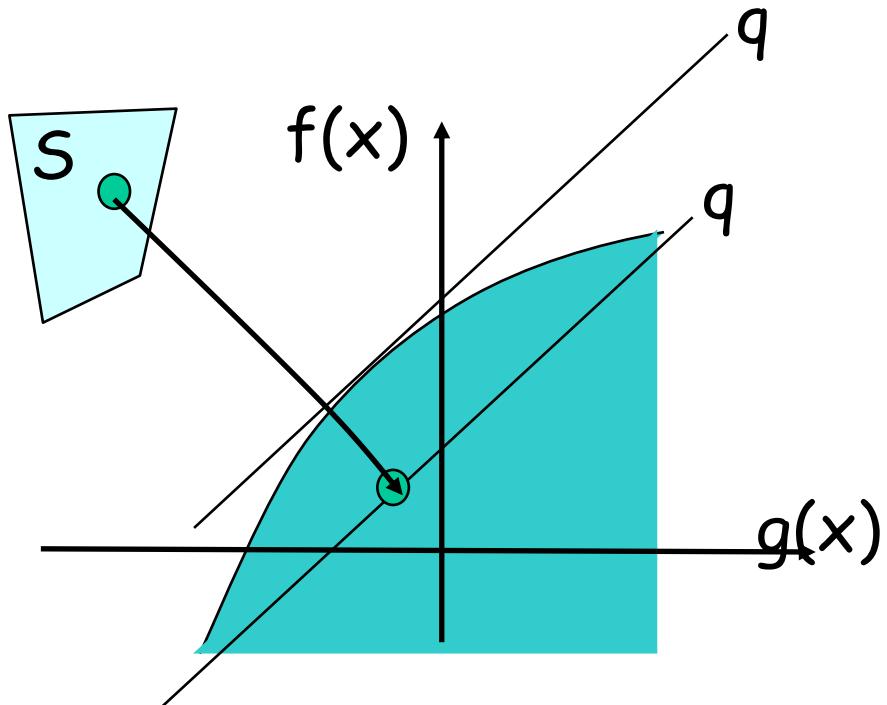
- Goal: Design a distributed alg to solve the problem.
- Discussion:
  - What are typical approaches to solve optimization, e.g.,?  
 $\max U(x)$
  - Why is the Resource-Allocation problem hard to solve by a distributed algorithm?

## A Two-Slide Summary of Constrained Convex Optimization Theory

$$\begin{array}{ll}
 \max & \sum_{f \in F} U_f(x_f) \\
 \text{subject to} & Ax \leq C \\
 \text{over} & x \geq 0
 \end{array}$$

$$\begin{array}{ll}
 \max & f(x) \\
 \text{subject to} & g(x) \leq 0 \\
 \text{over} & x \in S
 \end{array}$$

$f(x)$  concave  
 $g(x)$  linear  
 $S$  is a convex set



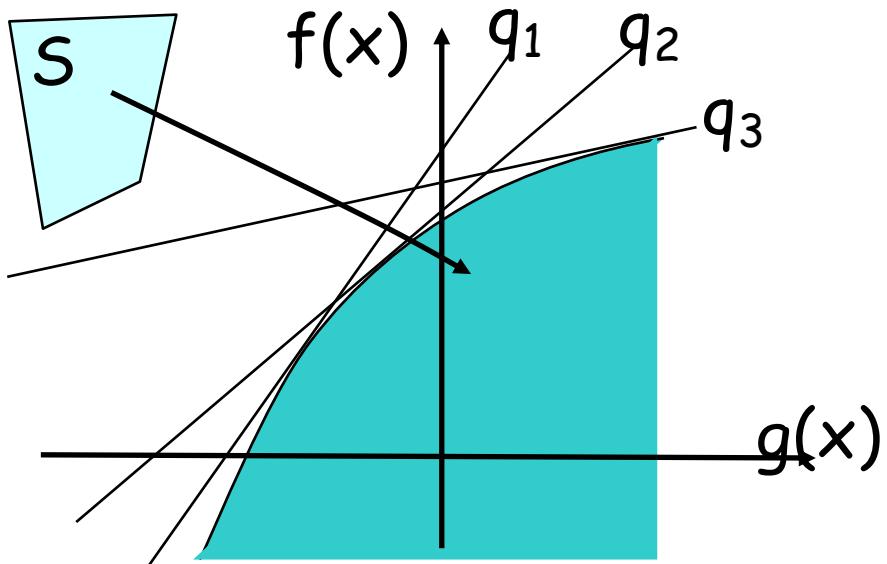
- Map each  $x$  in  $S$ , to  $[g(x), f(x)]$
- Top contour of map is concave
- Easy to read solution from contour
- For each slope  $q (>= 0)$ , computes  $f(x) - q g(x)$  of all mapped  $[f(x), g(x)]$

$$D(q) = \max_{x \in S} (f(x) - q g(x))$$

# A Two-Slide Summary of Constrained Convex Optimization Theory

$$\begin{array}{ll} \max & f(x) \\ \text{subject to} & g(x) \leq 0 \\ \text{over} & x \in S \end{array}$$

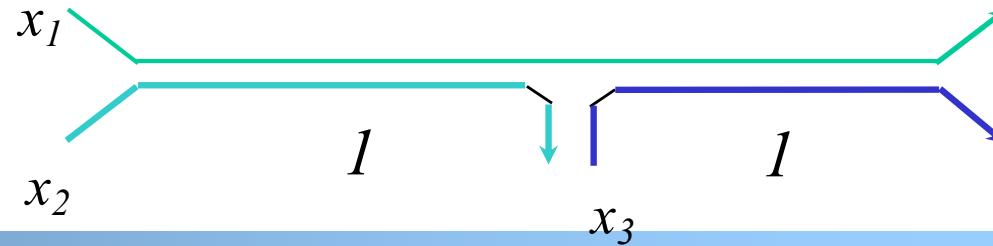
$f(x)$  concave  
 $g(x)$  linear  
 $S$  is a convex set



$$D(q) = \max_{x \in S} (f(x) - qg(x))$$

- $D(q)$  is called the dual;
- $q$  ( $>= 0$ ) are called prices in economics
- $D(q)$  provides an upper bound on obj.
- According to optimization theory:  
when  $D(q)$  achieves minimum over all  $q$  ( $>= 0$ ), then the optimization objective is achieved.

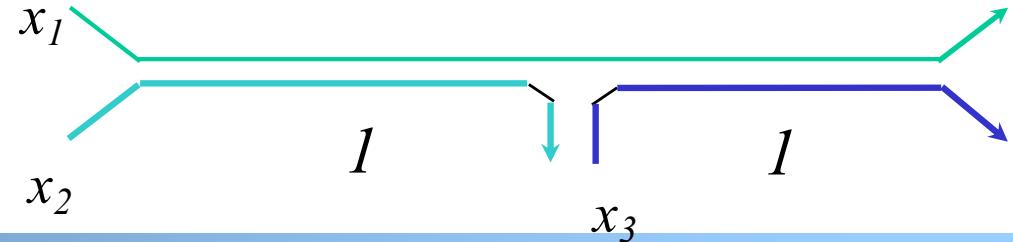
## Dual of the Primal



$$\begin{array}{ll}\max & \sum_{f \in F} U_f(x_f) \\ \text{subject to} & \sum_{f: f \text{ uses link } l} x_f \leq c_l \text{ for any link } l \\ \text{over} & x \geq 0\end{array}$$

$$D(q) = \max_{x_f \geq 0} \left( \sum_f U_f(x_f) - \sum_l q_l \left( \sum_{f: \text{uses } l} x_f - c_l \right) \right)$$

## Dual of the Primal



$$\begin{aligned} D(q) &= \max_{x_f \geq 0} \left( \sum_f U_f(x_f) - \sum_l q_l \left( \sum_{f: \text{uses } l} x_f - c_l \right) \right) \\ &= \max_{x_f \geq 0} \sum_f \left( U_f(x_f) - x_f \sum_{l: f \text{ uses } l} q_l \right) + \sum_l q_l c_l \\ &= \sum_f \max_{x_f \geq 0} \left( U_f(x_f) - x_f \sum_{l: f \text{ uses } l} q_l \right) + \sum_l q_l c_l \end{aligned}$$

## Distributed Optimization: User Problem

---

- Given  $p_f$  (=sum of dual var  $q_i$  along the path)  
flow  $f$  chooses rate  $x_f$  to maximize:

$$\begin{aligned} \max_{x_f} \quad & U_f(x_f) - x_f p_f \\ \text{over} \quad & x_f \geq 0 \end{aligned}$$

- Using the price signals, the optimization problem of each user is independent of each other!

## Distributed Optimization: User Problem

$$\begin{array}{ll} \max_{x_f} & U_f(x_f) - x_f p_f \\ \text{over} & x_f \geq 0 \end{array}$$

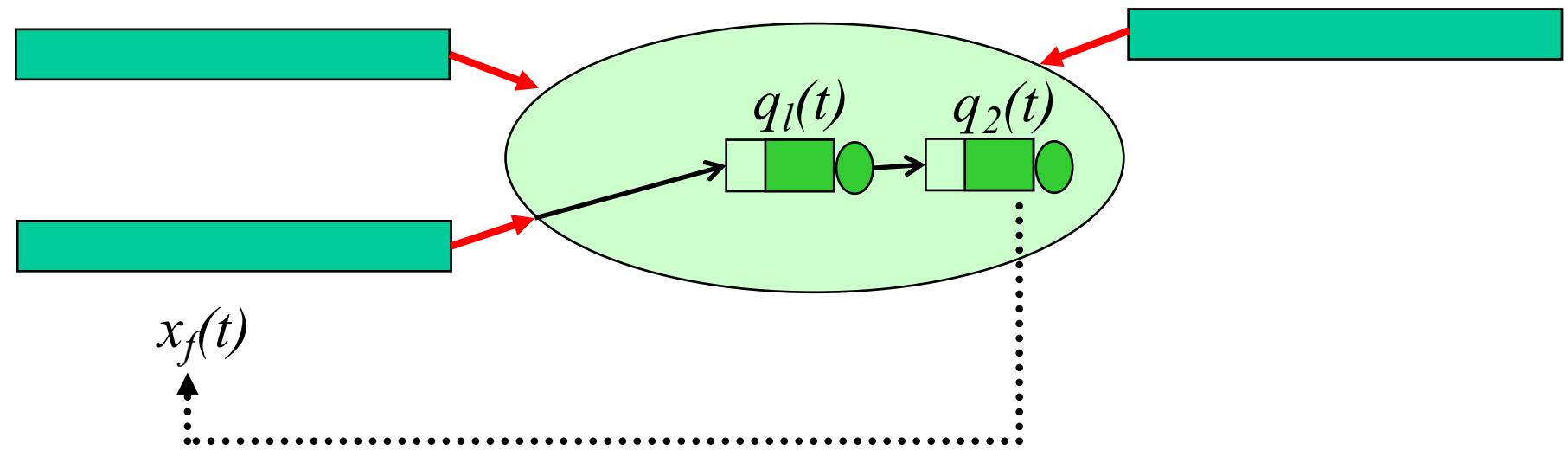
How should flow f adjust  $x_f$  locally?

$$\Delta x_f \propto U'_f(x_f) - p_f$$

At equilibrium (i.e., at optimal),  $x_f$  satisfies:

$$U'_f(x_f) - p_f = 0$$

# Interpreting Congestion Measure



$$p_f = \sum_{f \text{ uses } l} q_l$$

$$\Delta x_f \propto U'_f(x_f) - p_f$$

## Distributed Optimization: Network Problem

$$D(q) = \sum_f \max_{x_f \geq 0} \left( U_f(x_f) - x_f \sum_{l: f \text{ uses } l} q_l \right) + \sum_l q_l c_l$$

The network (i.e., link l) adjusts the link signals  $q_l$  (assume after all flows have picked their optimal rates given congestion signal)

$$\min_{q \geq 0} \tilde{D}(q) = \sum_l q_l \left( c_l - \sum_{f: f \text{ uses } l} x_f \right)$$

## Distributed Optimization: Network Problem

$$\min_{q \geq 0} D(q) = \sum_l q_l (c_l - \sum_{f: f \text{ uses } l} x_f)$$

how should link  $l$  adjust  $q_l$  locally?

$$\Delta q_l \propto -\frac{\partial D(q)}{q_l}$$

$$\frac{\partial}{\partial q_l} D(q) = c_l - \sum_{f: \text{uses } l} x_f$$

$$\Delta q_l \propto \sum_{f: \text{uses } l} x_f - c_l$$

# System Architecture

## □ SYSTEM(U):

$$\begin{array}{ll}\max & \sum_{f \in F} U_f(x_f) \\ \text{subject to} & \sum_{\substack{f: f \text{ uses link } l}} x_f \leq c_l \text{ for any link } l \\ \text{over} & x \geq 0\end{array}$$

## □ USER<sub>f</sub>:

$$\Delta x_f \propto U'_f(x_f) - p_f$$

$$\begin{array}{ll}\max_{x_f} & U_f(x_f) - x_f p_f \\ \text{over} & x_f \geq 0\end{array}$$

## □ NETWORK:

$$\Delta q_l \propto -\frac{\partial D(q)}{q_l}$$

$$\min_{q \geq 0} \tilde{D}(q) = \sum_l q_l (c_l - \sum_{\substack{f: f \text{ uses } l}} x_f)$$

# Decomposition Theorem

---

- There exist vectors  $p$ ,  $w$  and  $x$  such that
  1.  $w_f = p_f x_f$  for  $f \in F$
  2.  $w_f$  solves  $\text{USER}_f(U_f; p_f)$
  3.  $x$  solves  $\text{NETWORK}(w)$
- The vector  $x$  then also solves  $\text{SYSTEM}(U)$ .

# Outline

---

- Admin and recap
- Transport congestion control
  - what is congestion (cost of congestion)
  - basic congestion control alg.
  - TCP/Reno congestion control
  - TCP Cubic
  - TCP/Vegas
  - network wide resource allocation
    - general framework
    - objective function: an example axiom derivation of network-wide objective function
    - algorithm: a general distributed algorithm framework
    - application: TCP/Reno and TCP/Vegas revisited

# TCP/Reno Dynamics

---

$$\Delta x_f \propto U'_f(x_f) - p_f$$

$$\Delta W_{pkt} = (1-p)\frac{1}{W} - p\frac{W}{2}$$

$$\Delta W_{RTT} = \Delta W_{pkt} W = (1-p) - p\frac{W^2}{2} \cong 1 - p\frac{W^2}{2}$$

$$\Delta x = \frac{\Delta W_{RTT}}{RTT} = \frac{1}{RTT} - \frac{RTT}{2} p x^2$$

$$= \frac{RTT}{2} x^2 (\frac{2}{x^2 RTT^2} - p)$$

## TCP/Reno Dynamics

$$\Delta x_f \propto U'_f(x_f) - p_f$$

$$\Delta x = \frac{RTT}{2} x^2 \left( \frac{2}{x^2 RTT^2} - p \right)$$

$$U'_f(x_f) - p_f$$

$$\Rightarrow U'_f(x_f) = \left( \frac{\sqrt{2}}{x_f RTT} \right)^2 \Rightarrow U_f(x_f) = -\frac{2}{RTT^2 x_f}$$

## TCP/Vegas Dynamics

---

$$\Delta x_f \propto U'_f(x_f) - p_f$$

$$\Delta W_{RTT} \approx -(w - xRTT_{\min} - \alpha)$$

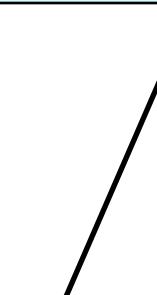
$$\begin{aligned}
 \Delta x &= \frac{\Delta W R_{TT}}{RTT} = -\left(\frac{w}{RTT} - \frac{x}{RTT} RTT_{\min} - \frac{\alpha}{RTT}\right) \\
 &= -\frac{w}{RTT} + \frac{x}{RTT} RTT_{\min} + \frac{\alpha}{RTT} \\
 &= -x + \frac{x}{RTT} RTT_{\min} + \frac{\alpha}{RTT} \\
 &= \frac{x}{RTT} (-RTT + RTT_{\min} + \frac{\alpha}{x}) \\
 &= \frac{x}{RTT} \left( \frac{\alpha}{x} - (RTT - RTT_{\min}) \right)
 \end{aligned}$$

$$\begin{aligned}
 \Delta W &\simeq \alpha - (W - \frac{RTT_{\min}}{RTT} W) \\
 &\simeq \alpha - (W - \frac{RTT_{\min}}{RTT} xRTT) \\
 &\simeq -(W - xRTT_{\min} - \alpha)
 \end{aligned}$$

## TCP/Vegas Dynamics

$$\Delta x_f \propto U'_f(x_f) - p_f$$

$$\Delta x = \frac{x}{RTT} (\frac{\alpha}{x} - (RTT - RTT_{min}))$$

$$U'_f(x_f) - p_f$$


$$\Rightarrow U'_f(x_f) = \frac{\alpha}{x}$$

$$\Rightarrow U_f(x_f) = \alpha \log(x_f)$$

## Summary: TCP/Vegas and TCP/Reno

---

- Pricing signal is queueing delay  $T_{queueing}$
- Pricing signal is loss rate  $p$

$$x_f = \frac{\alpha}{T_{queueing}}$$

$$U'_f(x_f) = T_{queueing}$$

$$\Rightarrow U'_f(x_f) = \frac{\alpha}{x_f}$$

$$\Rightarrow U_f(x_f) = \alpha \log(x_f)$$

$$x_f = \frac{\alpha}{RTT \sqrt{p}}$$

$$U'_f(x_f) = p$$

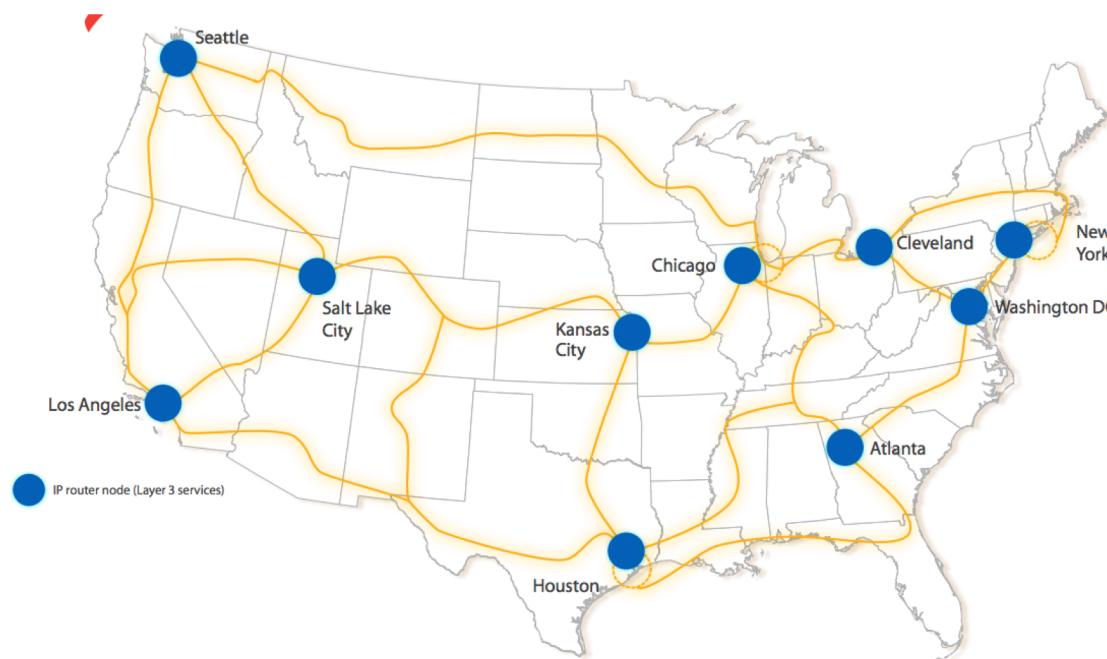
$$\Rightarrow U'_f(x_f) = \left( \frac{\alpha}{x_f RTT} \right)^2$$

$$\Rightarrow U_f(x_f) = -\frac{\alpha'}{RTT^2 x_f}$$

# Discussion

---

- Assume that you are given a set of flows deployed at a given network topology.
- What is a simple way to predict TCP rate allocation?



## Summary: Resource Allocation Frameworks

### □ Forward (design) engineering:

- how to determine objective functions
- given objective, how to design effective alg

$$\begin{array}{ll}\max & \sum_{f \in F} U_f(x_f) \\ \text{subject to} & Ax \leq C \\ \text{over} & x \geq 0\end{array}$$

### □ Reverse (understand) engineering:

- understand current protocols (what are the objectives of TCP/Reno, TCP/Vegas?)

### □ Additional pointers:

- <http://www.statslab.cam.ac.uk/~frank/pf/>