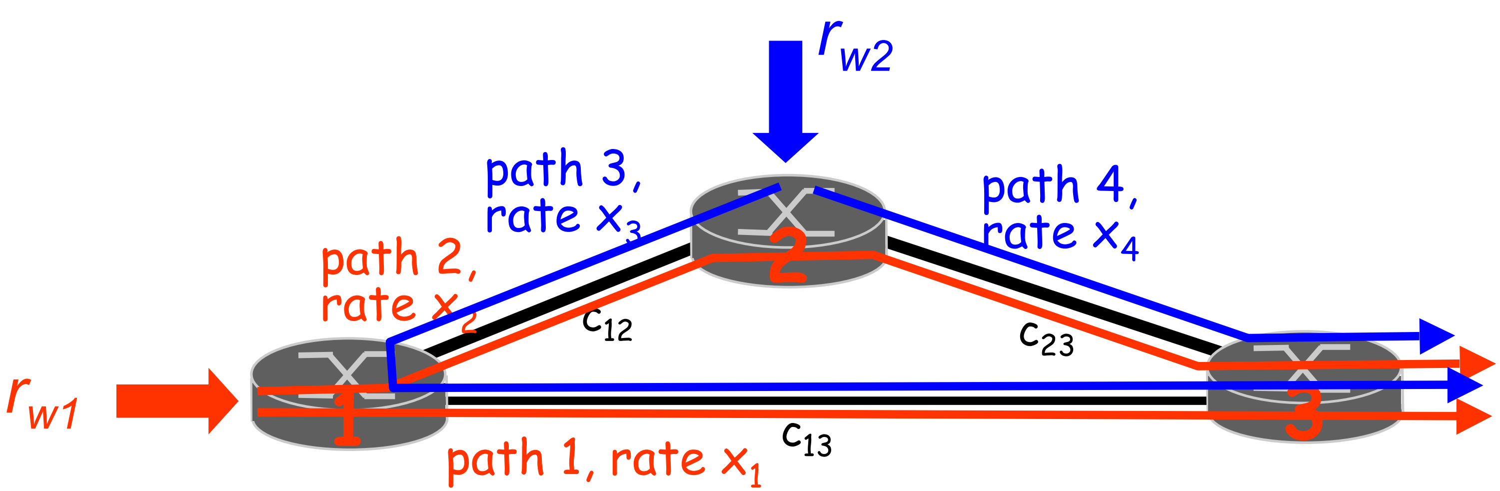
Optimization-based routing and congestion control

Routing, congestion control as optimization problems:

- how to route flows, set flow rates to optimize an objective (cost) function
- *routing and congestion control protocols as distributed asynchronous implementations of optimization algorithms*
  - systematic approach towards protocol design
  - e.g., TCP as distributed rate optimization

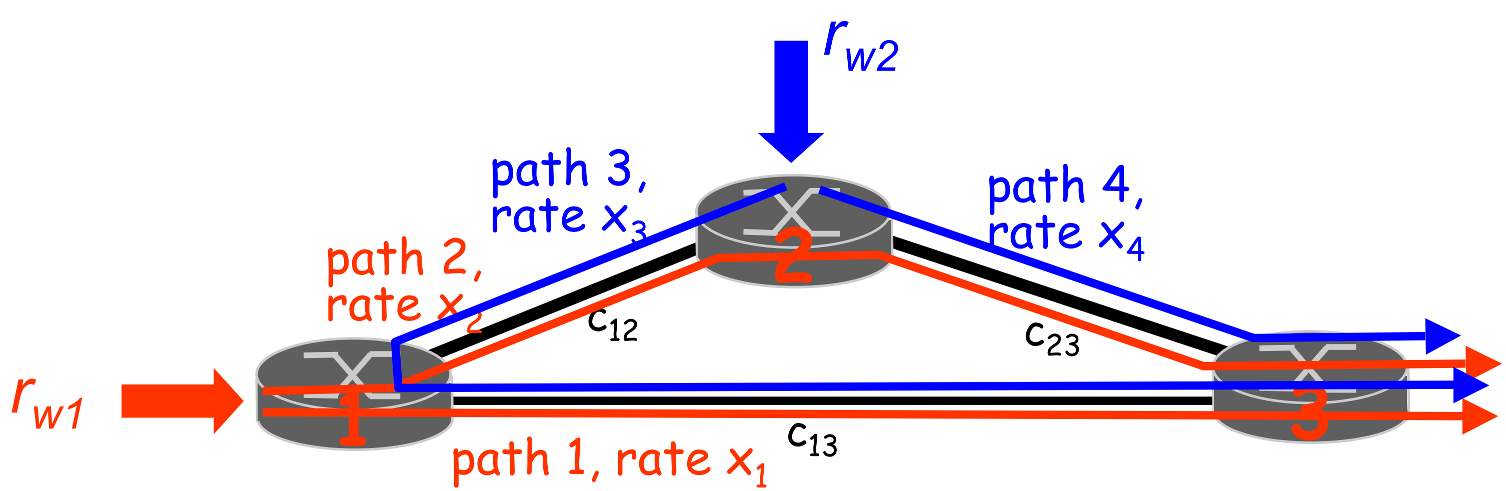
Optimization Framework

- W: set of source-destination (sd) pairs
- r_w: rate of sd pair w
- P_w: set of paths between sd pair w
- x_p: packet flow rate ("fluid") on path p
- c_{ij} link capacity of link i,j, (assume same as c_{ji})



Optimization Framework

- *routing problem:* r_w (rate of sd pair w) typically *given*
  - *question:* what rate x_p on each path p
- *rate control problem:* r_w (rate of sd pair w) *variable*
  - *question:* what rate x_p on each *given* path p
  - single path or multiple paths between sd pair w



Optimization Framework

Key question: how to set rates on paths? Input rates may be fixed (routing) or variable (rate control)

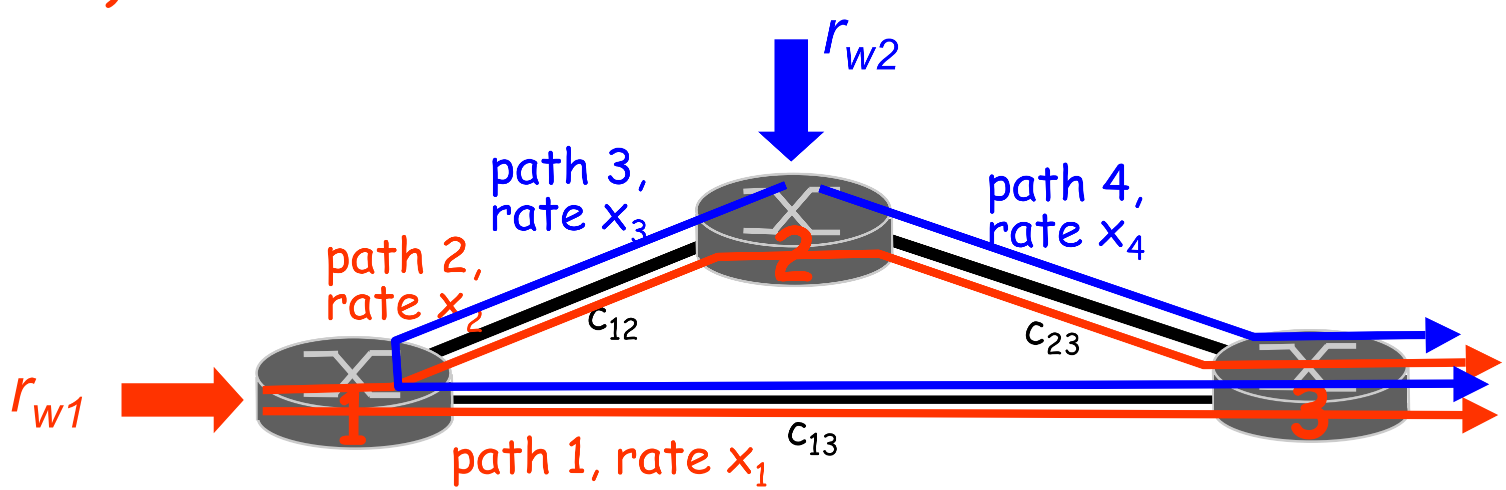
(rate of sd pair w) typically *fixed*

what rate x_p on each path p

problem: r_w (rate of sd pair w) *variable*

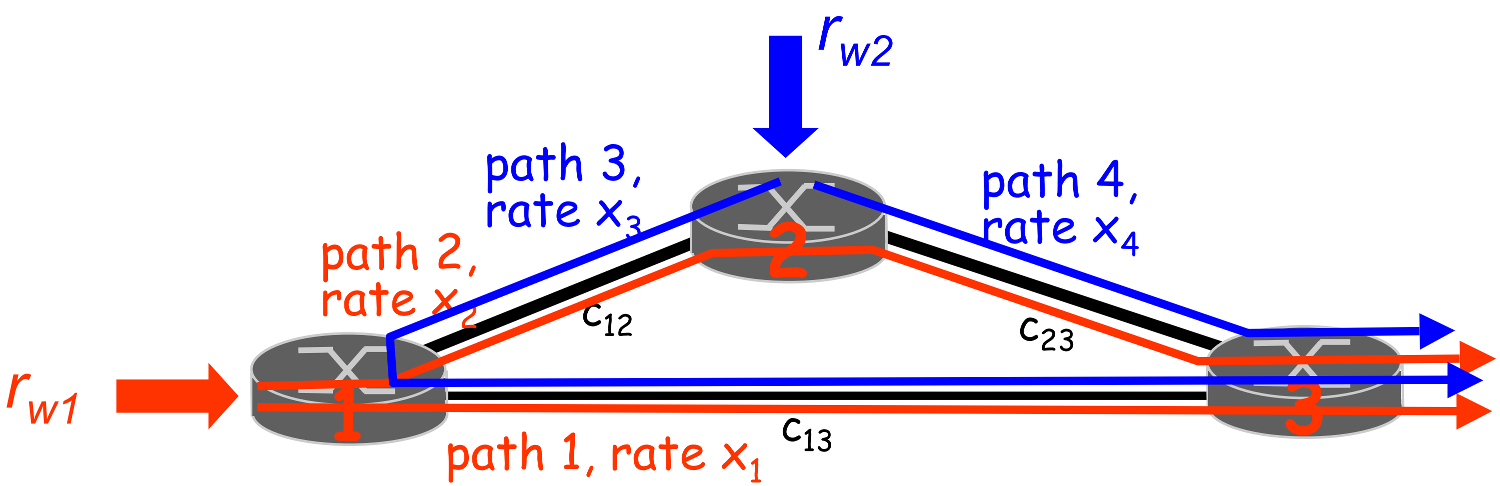
what rate x_p on each given path p

or multiple paths between sd pair w



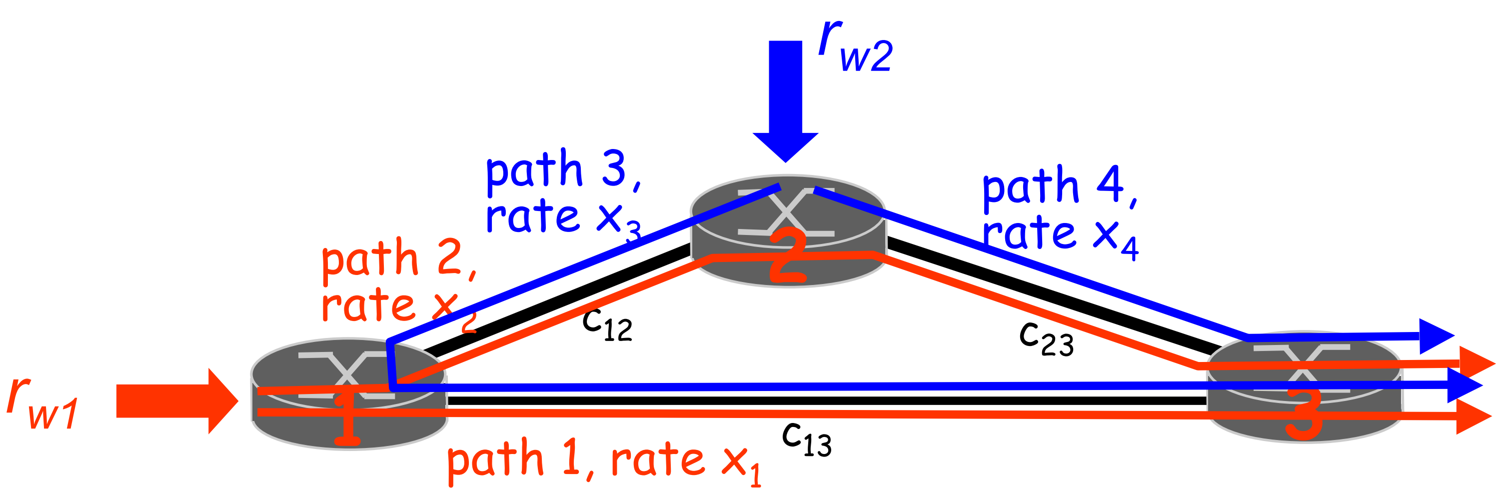
Optimization Framework

- lookout for: where are path rates set
  - *centrally:* global computation
  - *at endpoints:* distributed algorithm with multiple endpoints at network edge
  - *at routers:* distributed algorithm with multplie routers within network



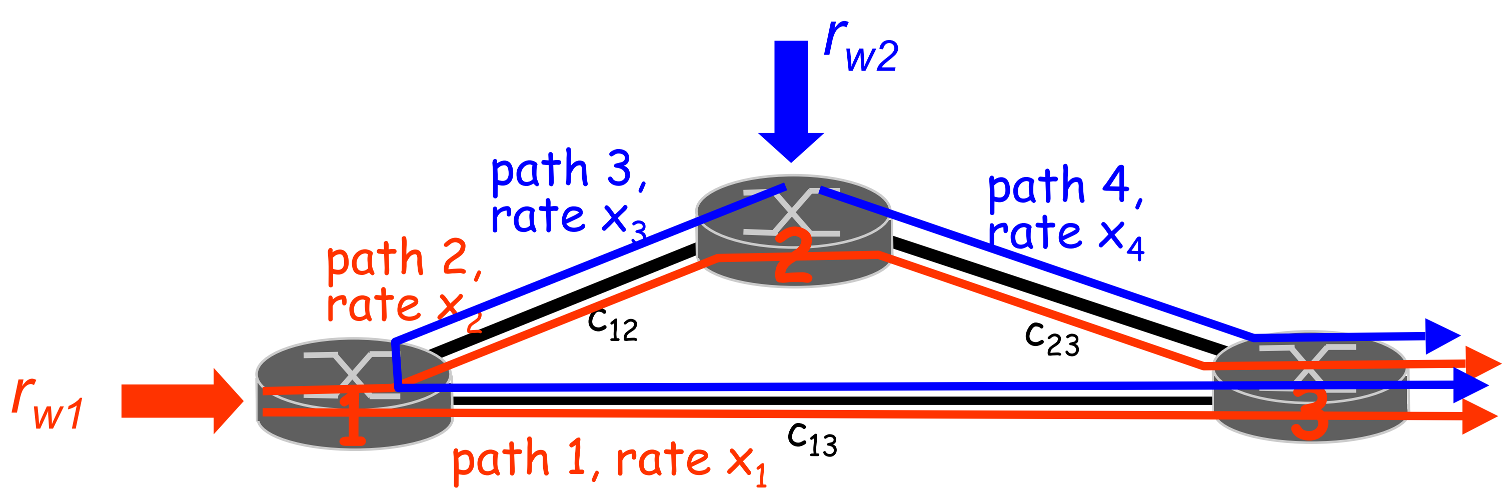
Optimization Framework

- lookout for: what cost function is being optimized? Typically:
  - *minimize* system-wide delay with variable routing given fixed sd traffic rates $\{r_w\}$
  - *maximize* system-wide utility, $\Sigma_w U_w(r_w)$, with variable traffic rates $\{r_w\}$ given fixed paths



Optimization Framework

- lookout for: how are capacity constraints taken into account



Optimization-based routing

- read: Gallagher 1992 [sec 5.4 intro, 5.5, 5.6], [Gallagher 1977].

W: set of source-destination (sd) pairs
r_w: *fixed* rate of sd pair w (traffic to be routed)
P_w: set of paths between sd pair w
x_p: packet flow rate ("fluid") on path p
c_{ij} link capacity of link i,j, (assume same as c_{ji})

$$F_{ij} = \sum_{\substack{\text{all paths } p \\ \text{crossing link } i,j}} x_p$$

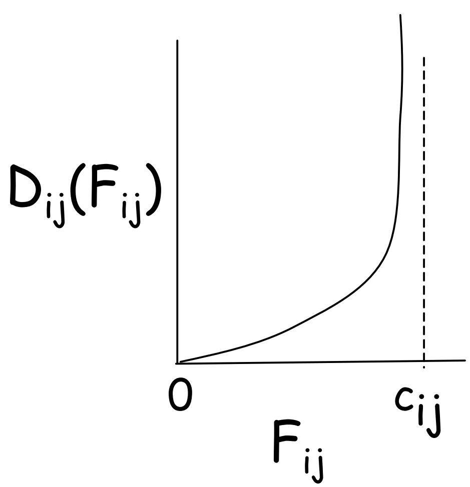
Optimization-based routing

minimize: $\sum_{\substack{\text{all links}\\ ij}} D_{ij}(F_{ij}) = \sum_{\substack{\text{all links}\\ ij}} \frac{F_{ij}}{c_{ij} - F_{ij}}$

subject to: $\sum_{\substack{\text{all paths}\\ p \text{ in } P_w}} x_p = r_w$ for all w in W

$x_p \geq 0$ for all p in P_w, w in W

all incoming flow (fixed) must be routed to dest.



as F_{ij} approaches c_{ij}, $D_{ij}(F_{ij})$ approaches infinity

Optimization-based routing

equivalently minimize: $\sum_{\substack{\text{all links} \\ ij}} D_{ij}\left(\sum_{\substack{\text{all paths } p \\ \text{with link } ij}} x_p\right)$

subject to: $\sum_{\substack{\text{all paths} \\ p \text{ in } P_w}} x_p = r_w$ for all w in W

$x_p \geq 0$ for all p in P_w, w in W

$\left.\right\}$ *all incoming flow (fixed) must be routed to dest.*

This is the routing optimization problem: how to choose $\{x_p\}$ to minimize cost function above.

How to solve optimal routing problem?

Insight: looking at properties of optimal routing solution suggests algorithms (protocols) for finding optimum!

minimize:

$$\sum_{\substack{\text{all links} \\ ij}} D_{ij}\Big(\sum_{\substack{\text{all paths } p \\ \text{with link } ij}} x_p\Big)$$

consider:
$$\frac{\partial D(x)}{\partial x_p} = \sum_{ij} \frac{\partial D_{ij}\big(\sum_{\substack{\text{all paths } p \\ \text{with link } ij}} x_p\big)}{\partial x_p}$$

and let: $x^* = \{x_p^*\}$ be optimal path flows

Suppose move small amount flow δ from p (with non-zero x^*_p at x^*) to path p' between same OD pair. Then:

$$\delta \frac{\partial D(x^*)}{\partial x'_p} \geq \delta \frac{\partial D(x^*)}{\partial x_p}$$

otherwise x^*_p in x^* would not be optimal

How to solve optimal routing problem?

Insight: looking at properties of optimal routing solution suggests algorithms (protocols) for finding optimum!

minimize:

$$\sum_{\substack{\text{all links}\\ ij}} D_{ij}\Big(\sum_{\substack{\text{all paths } p\\ \text{with link } ij}} x_p\Big)$$

<u>At the optimum:</u>

$$\frac{\partial D(x^*)}{\partial x_{p1}} = \frac{\partial D(x^*)}{\partial x_{p2}}$$

for all non-zero x_{p1}, x_{p2} in P_w, for all w

set flow rate on alternative path so partial derivatives (marginal utilities) are equal

This suggests an algorithm!

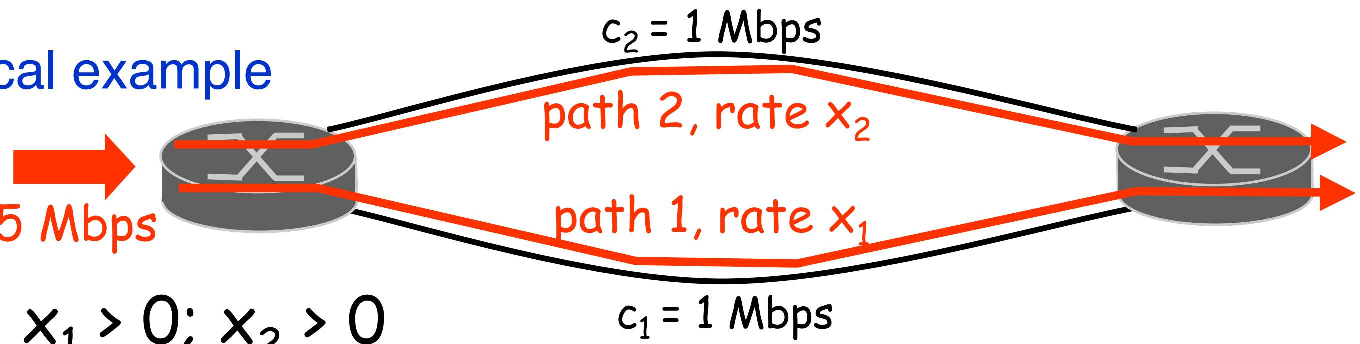
For each sd pair, w:

evaluate: $\dfrac{\partial D(x)}{\partial x_{p1}} \ldots \dfrac{\partial D(x)}{\partial x_{pk_w}}$ for all k_w paths for sd pair w

move "small" amount of flow to path with minimum marginal increase from other k_w-1 paths for sd pair w in attempt to balance marginal utilities on paths with non-zero flows

until for each sd pair, marginal utilities equal for all paths with non-zero flow for that sd pair for all sd pairs

A numerical example



$$x_1 + x_2 = 0.25; \quad x_1 \geq 0; \; x_2 \geq 0$$

$$D(x) = D_1(x_1) + D_2(x_2) = \frac{x_1}{c_1 - x_1} + \frac{x_2}{c_2 - x_2}$$

$$\frac{\partial D(x)}{\partial x_1} = \frac{c_1}{(c_1 - x_1)^2} \qquad \frac{\partial D(x)}{\partial x_2} = \frac{c_2}{(c_2 - x_2)^2}$$

current iteration:
x_1=0.2
x_2 = 0.05

$$\frac{\partial D(x)}{\partial x_1} = \frac{1}{(1 - 0.2)^2} \qquad \frac{\partial D(x)}{\partial x_2} = \frac{1}{(1 - 0.05)^2}$$

~ 1.6 ~ 1.1

partial derivative of D wrt x_2 is smaller, so shift "small" amount of flow from p_1 to p_2 and iterate

Routing optimization: “hill descent”

- *key idea:* iteratively evaluate marginal path costs (gradient descent)
- various algorithms to determine which flows can accept a “little” more flow, which should give up a “little” flow , subject to:
  - maintaining flow conservation
  - respecting capacity constraints
  - maintaining loop freedom

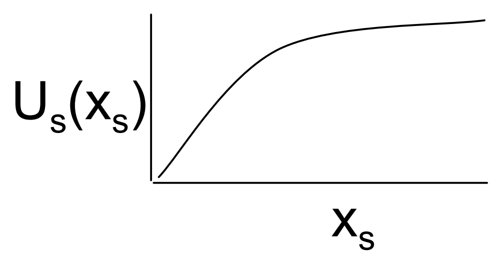
Optimization-based approach towards congestion control

Resource allocation as optimization problem:

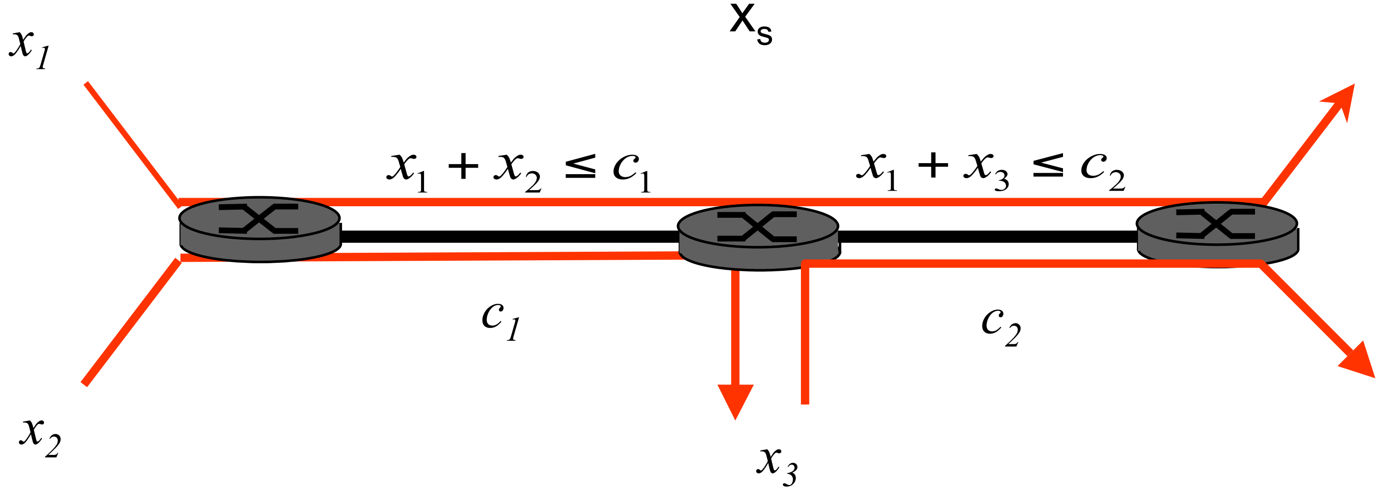
- how to allocate resources (e.g., bandwidth) to optimize some objective function
- maybe not possible that optimality exactly obtained but…
  - optimization framework as means to explicitly steer network towards desirable operating point
  - practical congestion control as distributed asynchronous implementations of optimization algorithm
  - systematic approach towards protocol design

Model

- Network: Links l each of capacity c_l
- Sources s: $(L(s), U_s(x_s))$
  $L(s)$ - links used by source s
  $U_s(x_s)$ - utility if source rate = x_s



example utility function for elastic application



Optimization Problem

$$\max_{x_s \geq 0} \quad \sum_s U_s(x_s)$$

$$\text{subject to} \quad \sum_{s \in S(l)} x_s \leq c_l, \forall l \in L$$

"system" problem

- maximize system utility (note: all sources "equal)
- constraint: bandwidth used less than capacity
- centralized solution to optimization impractical
  - must know all utility functions
  - impractical for large number of sources
  - we'll see: congestion control as distributed asynchronous algorithms to solve this problem

The user view

- user can choose amount to pay per unit time, w_s
- Would like allocated bandwidth, x_s in proportion to w_s

$$x_s = \frac{w_s}{p_s}$$

p_s could be viewed as charge per unit flow for user s

$$\max \quad \boxed{U_s\left(\frac{w_s}{p_s}\right)} - \boxed{w_s}$$

$$\text{subject to} \quad w_s \geq 0$$

user problem

user's utility

cost

The network view

- suppose network knows vector $\{w_s\}$, chosen by users
- network wants to maximize logarithmic utility function

$$\max_{x_s \geq 0} \quad \sum_s w_s \log x_s$$

$$\text{subject to} \quad \sum_{s \in S(l)} x_s \leq c_l$$

network problem

Solution existence

- There exist prices, p_s, source rates, x_s, and amount-to-pay-per-unit-time, $w_s = p_s x_s$ such that
  - $\{W_s\}$ solves user problem
  - $\{X_s\}$ solves the network problem
  - $\{X_s\}$ is the unique solution to the system problem

$$\max \quad U_s\left(\frac{w_s}{p_s}\right) - w_s$$

$$\text{subject to} \quad w_s \geq 0$$

$$\max_{x_s \geq 0} \quad \sum_s w_s \log x_s$$

$$\text{subject to} \quad \sum_{s \in S(l)} x_s \leq c_l$$

$$\max_{x_s \geq 0} \quad \sum_s U_s(x_s)$$

$$\text{subject to} \quad \sum_{s \in S(l)} x_s \leq c_l, \forall l \in L$$

Proportional Fairness

- Vector of rates, $\{X_s\}$, proportionally fair if feasible and for any other feasible vector $\{X_s^*\}$:

$$\sum_{s \in S} \frac{x_s^* - x_s}{x_s} \leq 0$$

result: if $w_r=1$, then $\{X_s\}$ solves the network problem IFF it is proportionally fair

Related results exist for the case that w_r not equal 1.

Solving the network problem

- Results so far: *existence* - solution exists with given properties
- How to *compute* solution?
  - ideally: distributed solution easily embodied in protocol
  - insight into existing protocol

Solving the network problem

$$\underbrace{\frac{d}{dt}x_s(t)}_{\text{change in bandwidth allocation at } s} = k\left(\underbrace{w_s}_{\text{linear increase}} - \underbrace{x_s(t)\sum_{l\in L(s)} p_l(t)}_{\text{multiplicative decrease}}\right)$$

where $\quad p_l(t) = g_l\left(\sum_{l\in L(s)} x_s(t)\right)$

congestion "signal": function of aggregate rate at link l, fed back to s.

Solving the network problem

$$\underbrace{\frac{d}{dt}x_s(t)}_{} = k\left(\underbrace{w_s}_{} - \underbrace{x_s(t)\sum_{l \in L(s)} p_l(t)}_{}\right)$$

- Results:
  - * converges to solution of “relaxation” of network problem
  - $x_s(t)\Sigma p_l(t)$ converges to w_s
- Interpretation: TCP-like algorithm to iteratively solve optimal rate allocation!

Optimization-based congestion control: summary

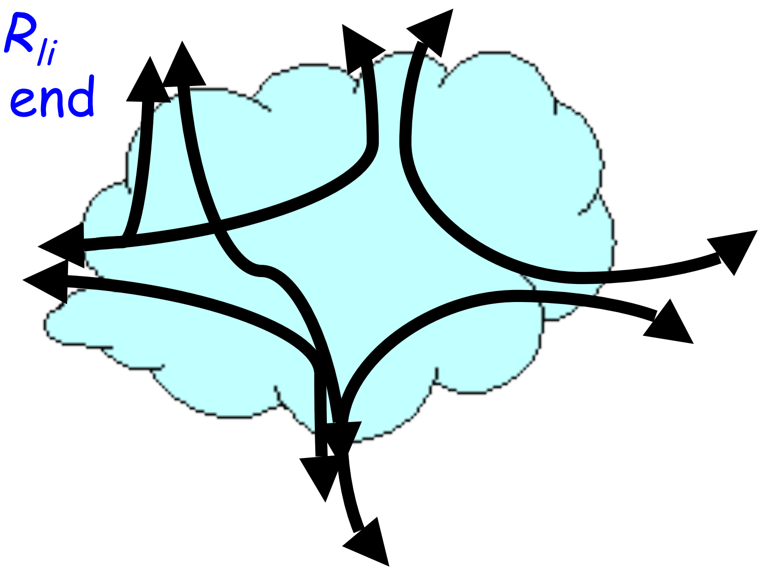
- bandwidth allocation as optimization problem:
- practical congestion control (TCP) as distributed asynchronous implementations of optimization algorithm
- optimization framework as means to explicitly steer network towards desirable operating point
- systematic approach towards protocol design

Motivation

Congestion Control: maximize user utility

Traffic Engineering: minimize network congestion

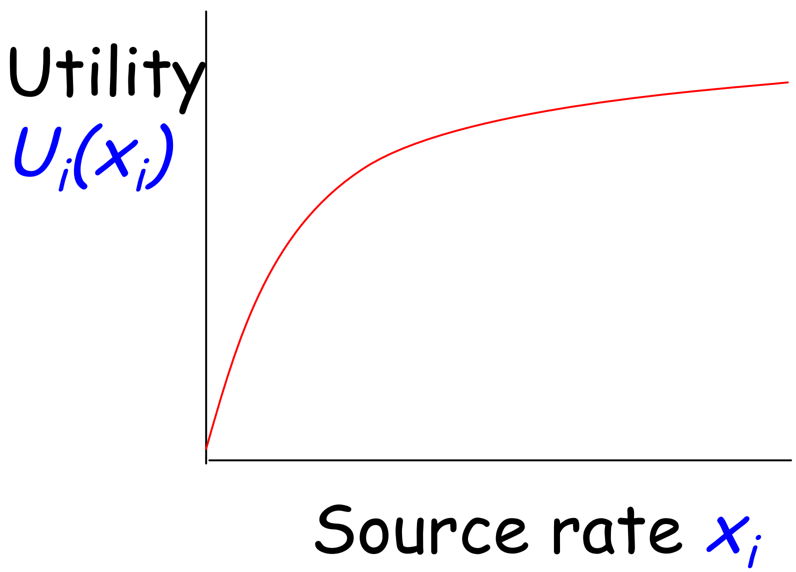
Given routing R_{li} how to adapt end rate x_i?



Given traffic x_i how to perform routing R_{li}?

Congestion Control Model

Users are indexed by i



aggregate utility

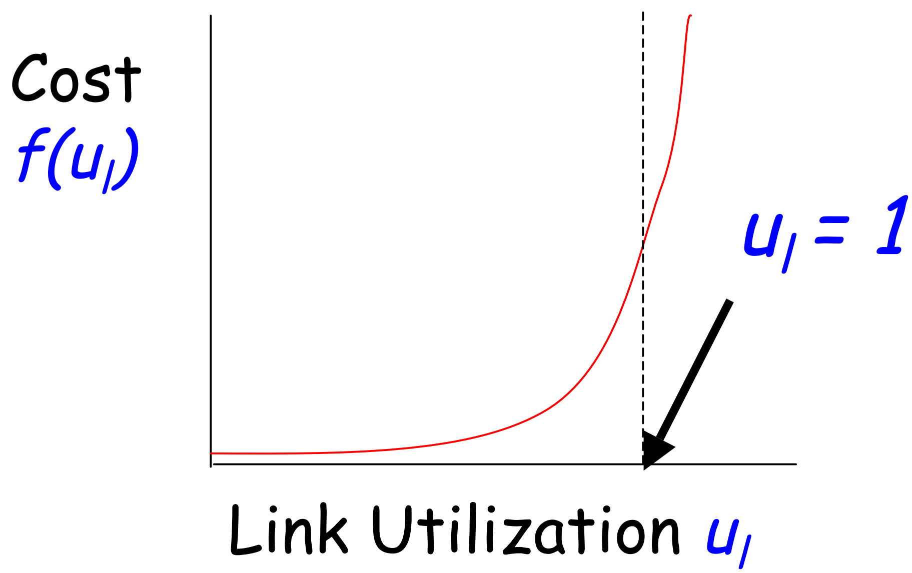
$$\begin{aligned}
\text{max. } & \textstyle\sum_i U_i(x_i) \\
\text{s.t. } & \textstyle\sum_i R_{li} x_i \le c_l \\
\text{var. } & x
\end{aligned}$$

capacity constraints

Congestion control provides fair rate allocation amongst users

Traffic Engineering Model

Links are indexed by l



aggregate cost

$$\text{min. } \sum_l f(u_l)$$
$$\text{s.t. } \quad u_l = \sum_i R_{li} x_i / c_l$$
$$\text{var. } \quad R$$

Traffic engineering avoids bottlenecks in the network

Model of Internet Reality

Congestion Control:

$\max \sum_i U_i(x_i),$

$\text{s.t. } \sum_i R_{li} x_i \le c_l$

Traffic Engineering:

$\min \sum_l f(u_l),$

$\text{s.t. } u_l = \sum_i R_{li} x_i / c_l$

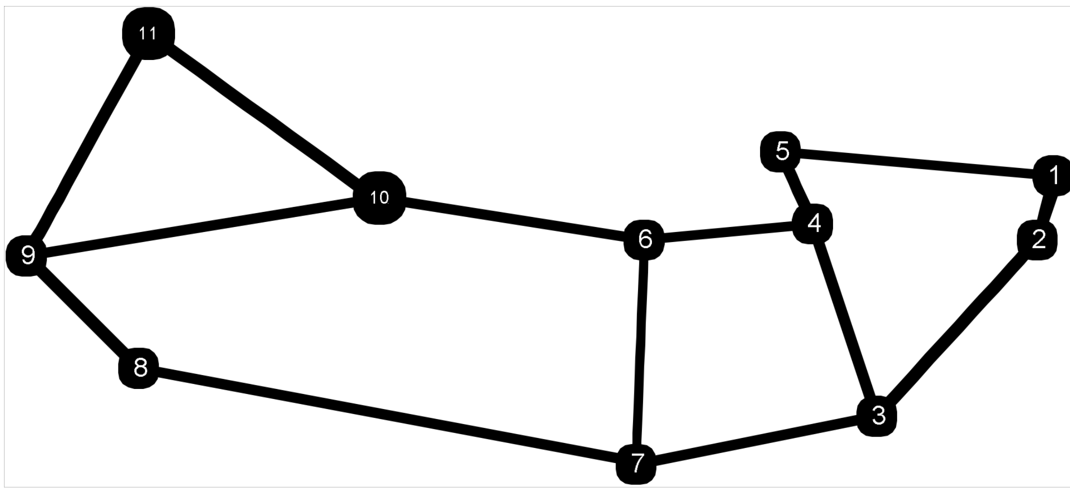
System Properties

- Convergence
- Does it achieve some objective?
- Benchmark:

$$
\begin{aligned}
&\text{max.} && \textstyle\sum_i U_i(x_i) \\
&\text{s.t.} && Rx \le c \\
&\text{Var.} && x, R
\end{aligned}
$$

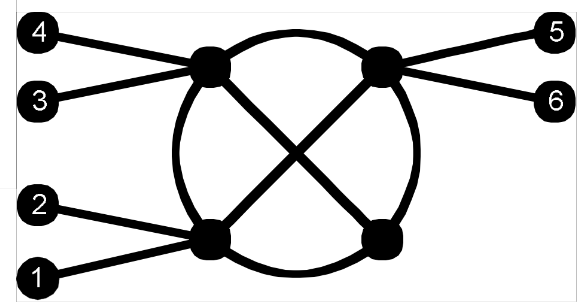
- Utility gap between the joint system and benchmark

Numerical Experiments



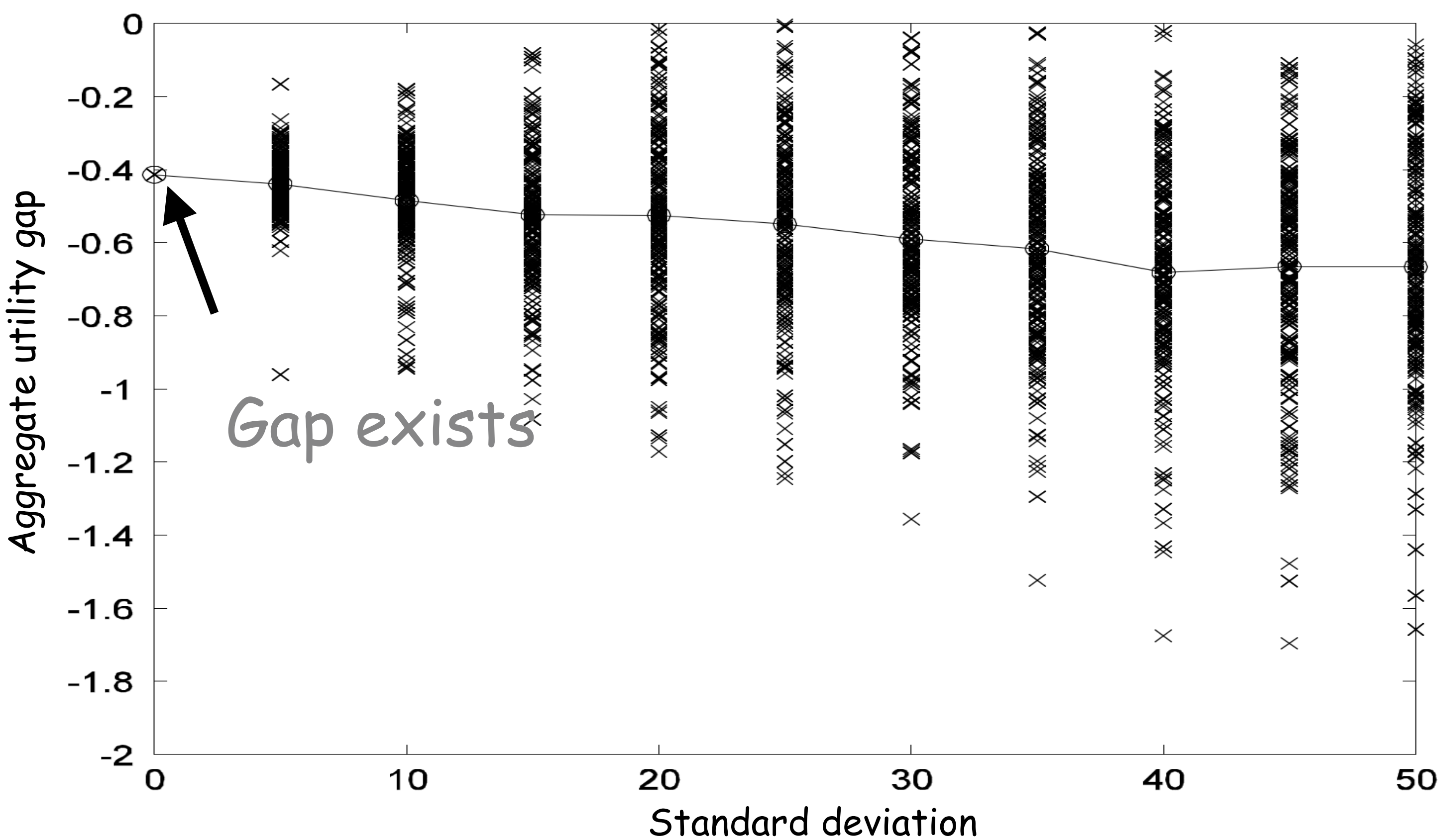
Abilene Internet2

Access-Core



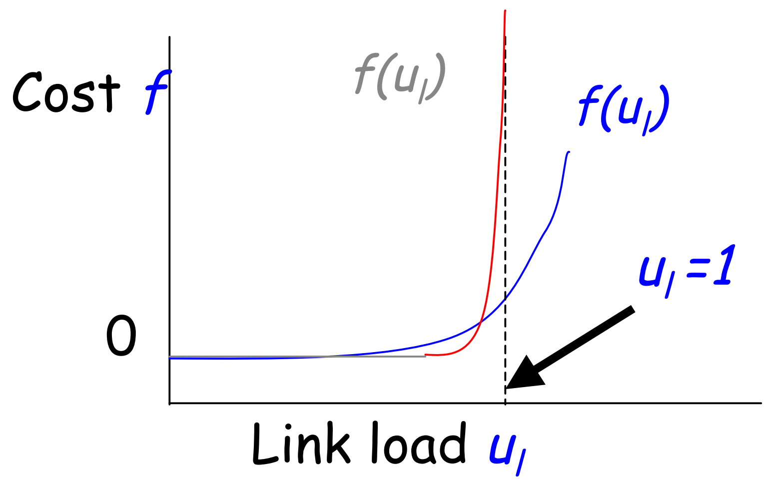
- System converges
- Quantify the gap to optimal aggregate utility
- Capacity distribution: truncated Gaussian with average 100
- 500 points per standard deviation

Results for Abilene: $f = e^{u\_l}$

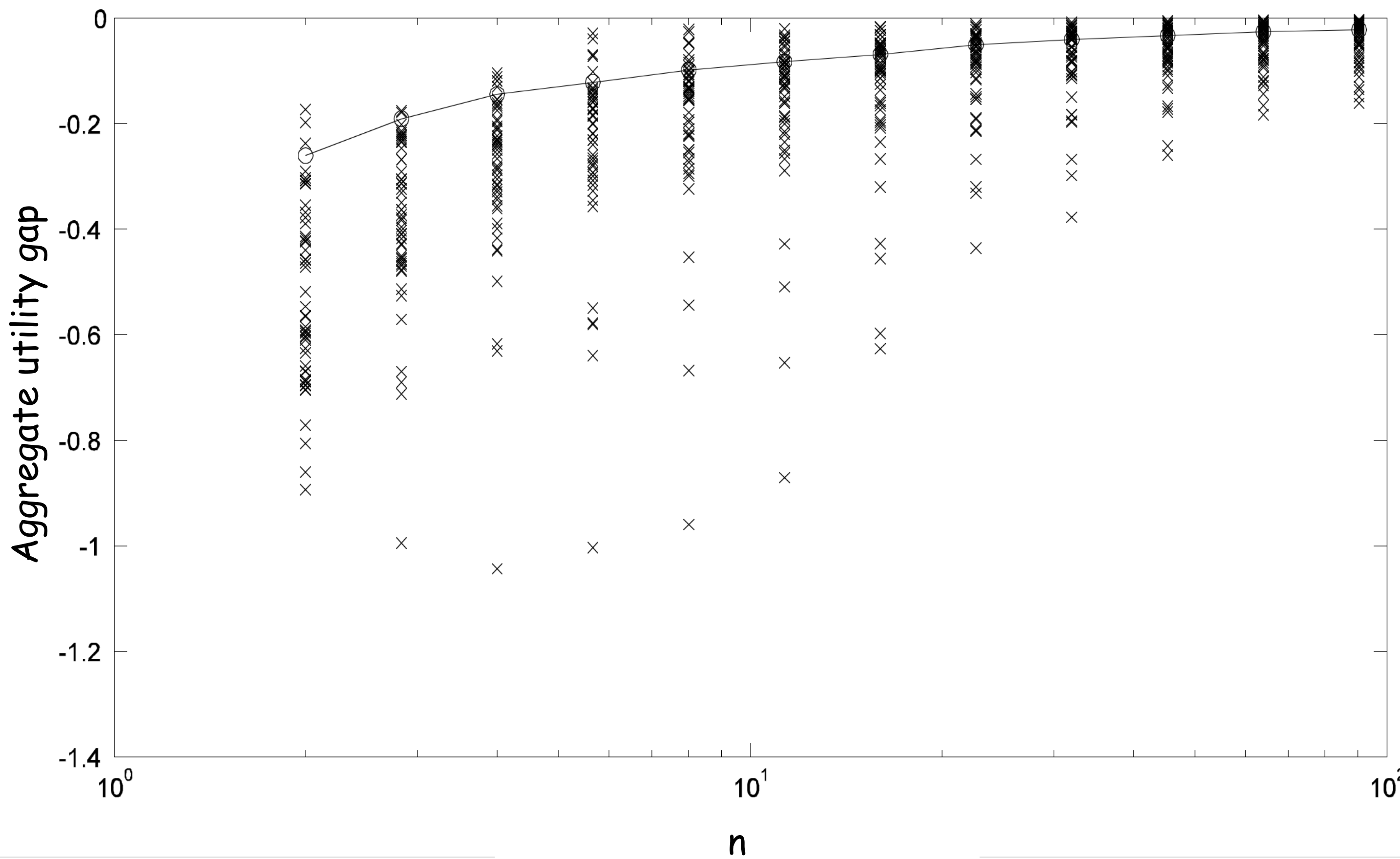


Backward Compatible Design

- Simulation of the joint system suggests that it is stable, but suboptimal
- Gap reduced if we modify *f*



Abilene Continued: $f = n(u_l)^n$



Gap shrinks with larger n

Theoretical Results

- Modify congestion control to approximate the capacity constraint with a penalty function
- Theorem: modified joint system model converges if $U_i''(x_i) \leq -U_i'(x_i)/x_i$

Master Problem:

$$\min.\ g(x,R) = -\sum_i U_i(x_i) + \gamma \sum_l f(u_l)$$

Congestion Control:

$$\operatorname{argmin}_x g(x,R)$$

Traffic Engineering:

$$\operatorname{argmin}_R g(x,R)$$