

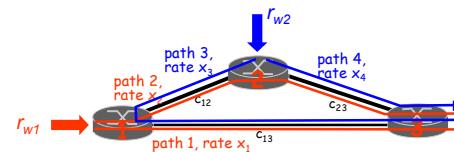
Optimization-based routing and congestion control

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

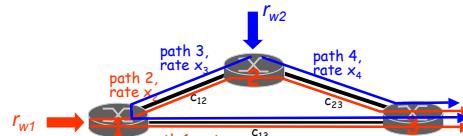
Optimization Framework

- 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})



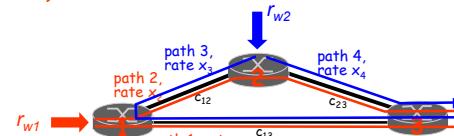
Optimization Framework

- *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



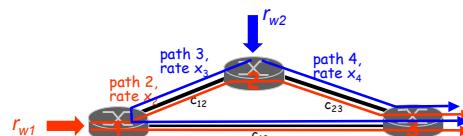
Optimization Framework

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



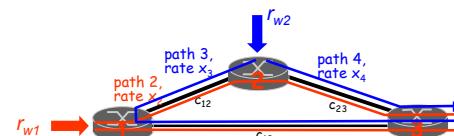
Optimization Framework

- 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 multiple routers within network



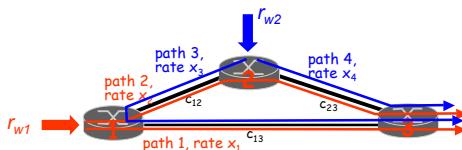
Optimization Framework

- 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, $\sum_w U_w(r_w)$, 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

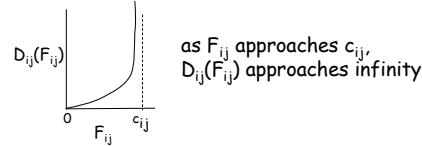
$$\text{minimize: } \sum_{\substack{\text{all links} \\ ij}} D_{ij}(F_{ij}) = \sum_{\substack{\text{all links} \\ ij}} \frac{F_{ij}}{c_{ij} - F_{ij}}$$

subject to:

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

$$x_p \geq 0 \quad \text{for all } p \in P_w, \quad w \in W$$

(fixed) must be routed to dest.



Optimization-based routing

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

subject to:

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

$$x_p \geq 0 \quad \text{for all } p \in P_w, \quad w \in W$$

(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!

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

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:

$$\delta \frac{\partial D(x^*)}{\partial x_p} \geq \delta \frac{\partial D(x^*)}{\partial x_{p'}} \quad \text{otherwise } x_{p'}^* \text{ in } x^* \text{ would not be optimal}$$

How to solve optimal routing problem?

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

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

At the optimum:

$$\frac{\partial D(x^*)}{\partial x_{p1}} = \frac{\partial D(x^*)}{\partial x_{p2}} \quad \text{for all non-zero } x_{p1}, x_{p2} \text{ in } P_w, \text{ for all } w$$

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

This suggests an algorithm!

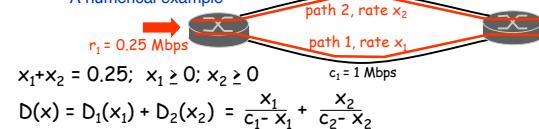
For each sd pair, w:

evaluate: $\frac{\partial D(x)}{\partial x_{p1}} \dots \frac{\partial D(x)}{\partial x_{pk_w}}$ 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



$$\begin{aligned} \frac{\partial D(x)}{\partial x_1} &= \frac{c_1}{(c_1 - x_1)^2} & \frac{\partial D(x)}{\partial x_2} &= \frac{c_2}{(c_2 - x_2)^2} \\ \text{current} & & & \\ \text{iteration: } & \frac{\partial D(x)}{\partial x_1} = \frac{1}{(1 - 0.2)^2} & \frac{\partial D(x)}{\partial x_2} = \frac{1}{(1 - 0.05)^2} \\ x_1 = 0.2 & & x_2 = 0.05 & \\ x_2 = 0.05 & & & \\ & \sim 1.6 & & \sim 1.1 \end{aligned}$$

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

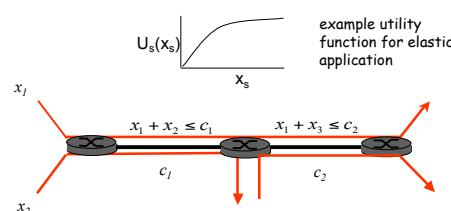
Optimization-based approach towards congestion control

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



Optimization Problem

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

subject to $\sum_s x_s \leq c_l, \forall l \in L$

"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

$$\begin{aligned} & \max \quad U_s \left(\frac{w_s}{p_s} \right) - w_s \\ & \text{subject to } w_s \geq 0 \\ & \text{user's utility} \quad \text{cost} \end{aligned}$$

user problem

The network view

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

$$\begin{aligned} & \max_{x_s \geq 0} \quad \sum_s w_s \log x_s \\ & \text{subject to } \sum_{s \in S(l)} x_s \leq c_l \end{aligned}$$

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**

$$\begin{aligned} & \max \quad U_s \left(\frac{w_s}{p_s} \right) - w_s \\ & \text{subject to } w_s \geq 0 \end{aligned}$$

$$\begin{aligned} & \max_{x_s \geq 0} \quad \sum_s w_s \log x_s \\ & \text{subject to } \sum_{s \in S(l)} x_s \leq c_l \end{aligned}$$

$$\begin{aligned} & \max_{x_s \geq 0} \quad \sum_s U_s(x_s) \\ & \text{subject to } \sum_{s \in S(l)} x_s \leq c_l, \forall l \in L \end{aligned}$$

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 $w_i=1$, then $\{X_s\}$ solves the network problem IFF it is proportionally fair

Related results exist for the case that w_i 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

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

change in linear multiplicative bandwidth increase decrease allocation at s

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

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

Solving the network problem

$$\frac{d}{dt} x_s(t) = k \left(w_s - x_s(t) \sum_{i \in I(s)} p_i(t) \right)$$

- Results:

- * converges to solution of "relaxation" of network problem
- $x_s(t) \sum p_i(t)$ 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

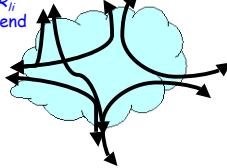
Motivation

Congestion Control:
maximize user utility

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

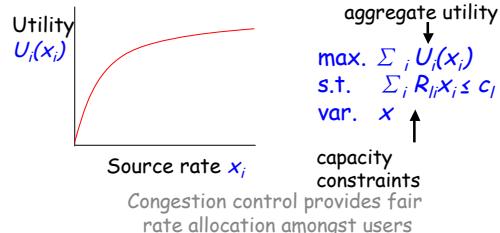
Traffic Engineering:
minimize network congestion

Given traffic x_i ,
how to perform
routing R_{ii} ?



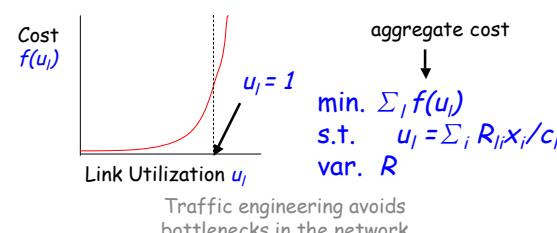
Congestion Control Model

Users are indexed by i



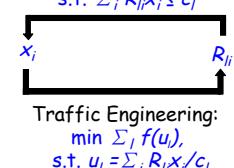
Traffic Engineering Model

Links are indexed by l



Model of Internet Reality

Congestion Control:
 $\max \sum_i U_i(x_i)$,
s.t. $\sum_i R_{ii}x_i \leq c_i$



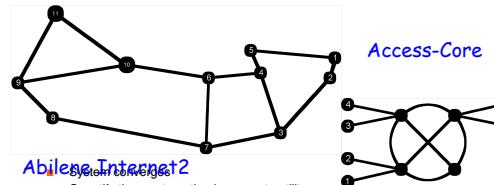
System Properties

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

$$\begin{aligned} & \max. \sum_i U_i(x_i) \\ & \text{s.t. } Rx \leq c \end{aligned}$$

- Utility gap between the joint system and benchmark

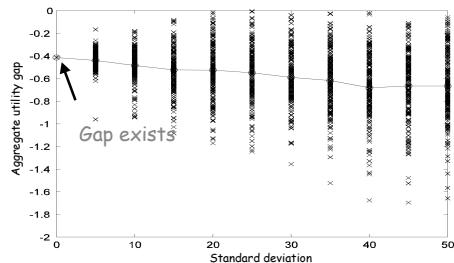
Numerical Experiments



Abilene Internet2

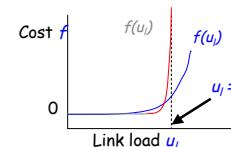
- System configuration
- Quantify the gap to optimal aggregate utility
- Capacity distribution: truncated Gaussian with average 100
- 500 points per standard deviation

Results for Abilene: $f = e^{u-1}$

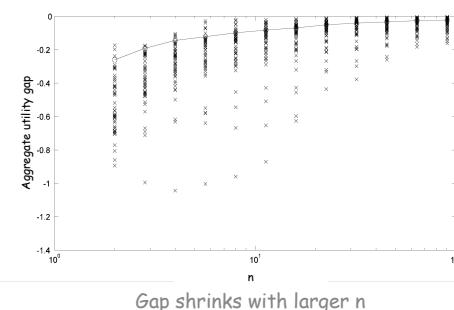


Backward Compatible Design

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



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



Theoretical Results

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

$$\begin{aligned} & \text{Master Problem:} \\ & \min. g(x, R) = -\sum_i U_i(x_i) + \gamma \sum_i f(u_i) \end{aligned}$$

$$\text{Congestion Control: } \arg\min_x g(x, R)$$

$$\text{Traffic Engineering: } \arg\min_R g(x, R)$$