Hot Sticky Random Multipath or Energy Pooling

Jon Crowcroft,

http://www.cl.cam.ac.uk/~jac22

Multipath: Network & End2End

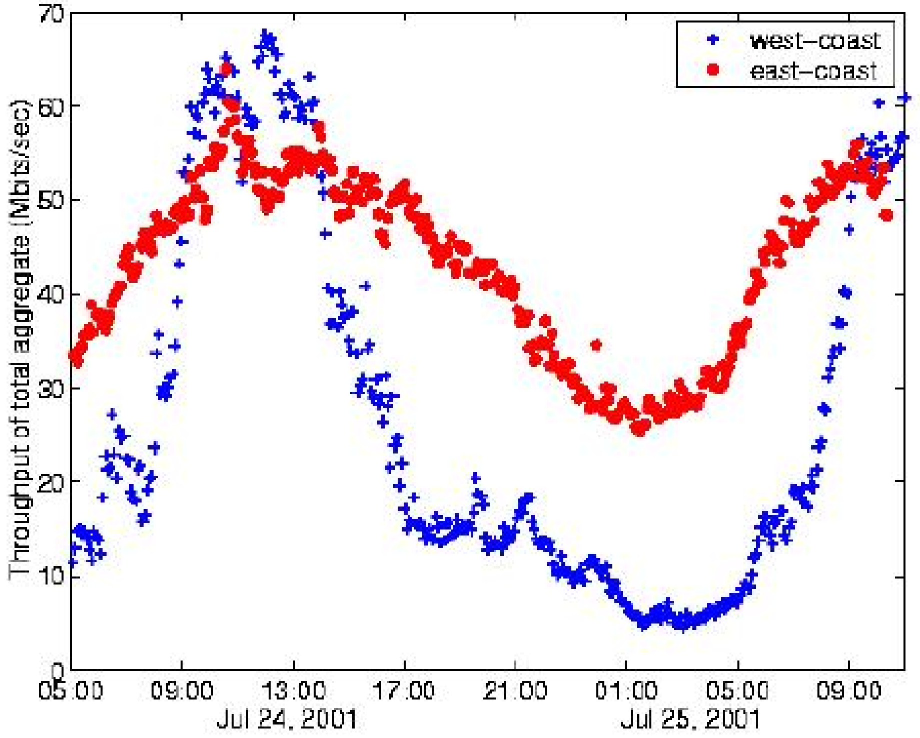
- High level argument is simply a Trilogy rerun:
  - *Multipath is resource pooling in capacity terms*
  - *Multipath IP spreads traffic*
    - *Currently only ECMP. pre-computed/designated equivalent paths*
    - *Better would be random j out of k shortes paths*
    - *Basis: stick random (DAR), per Gibbens/Kelly/Key et al*
  - *Multipath TCP splits load*
    - *Goal is same rate as std TCP, but use eqn to determine proportion on* subpath
    - Subpath is indeicated easily for multihomed host(s)
    - More tricky for single home, multiple route...tbd
    - Rate set from loss/rtt or ECN, ideally

S/Congestion/Carbon

- Let us set a price for power in the routers and transmission lines
- Then we give feedback (EECN?) based on this price

- Time of day variation in traffic means that at night, typically we see
- night:daytime traffic matrix about 6 fold reduction in load
  - So re-confugure the *notional* cost of the shortest paths only to be 6-10 times the price
  - And reduce the *actual* rate on all links by the same ratio
- Need energy proportional interfaces (of course)
  - i.e. need line cards/transmission that can
  - Slow down and use less energy
  - That isn't rocket science (phones and laptops do this routinely
  - with typically 3 settings)

Sprint 24 hour coast-coast trace



Cue:Optimization-based congestion control (ack: uMass for slides)

Resource allocation as optimization problem:

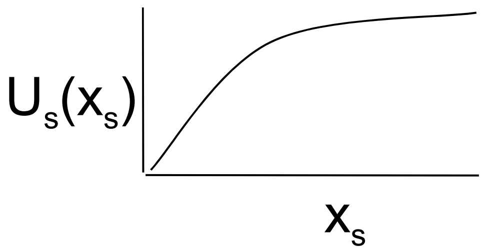
- how to allocate resources (e.g., bandwidth) to optimize some objective function
- maybe not possible that optimality exactly obtained but…
  - optimization framework as means to explicitly steer network towards desirable operating point
  - practical congestion control as distributed asynchronous implementations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

Model

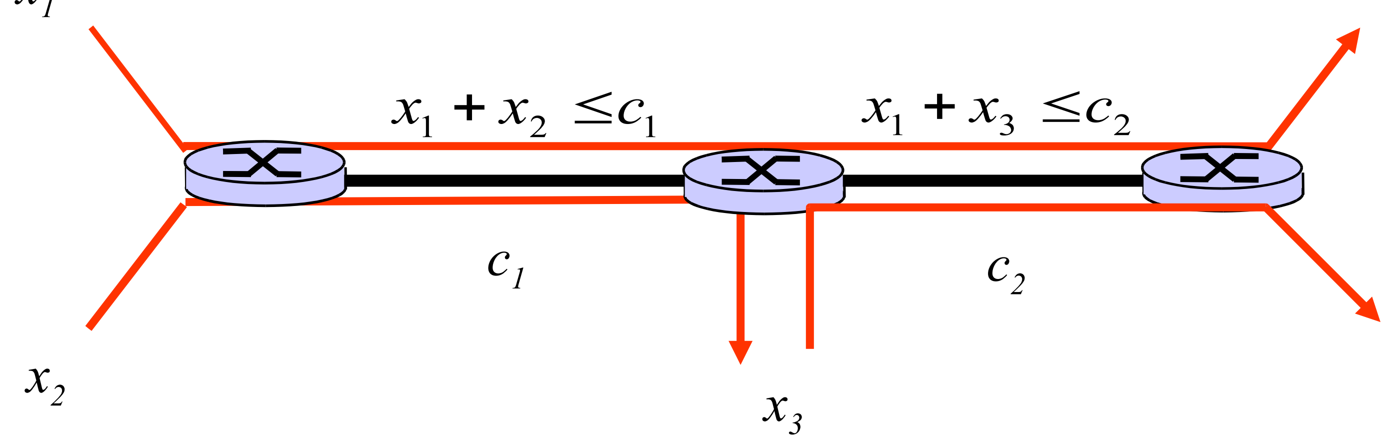
- Network: Links l each of capacity c_l
- Sources s: $(L(s), U_s(x_s))$

  $L(s)$ - 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

$$\max \quad \underbrace{\boxed{U_s\left(\frac{w_s}{p_s}\right)}}_{\text{user's utility}} - \underbrace{\boxed{w_s}}_{\text{cost}} \qquad \text{user problem}$$

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

The network view

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

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

network problem

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

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

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

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

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

Solving the network problem

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

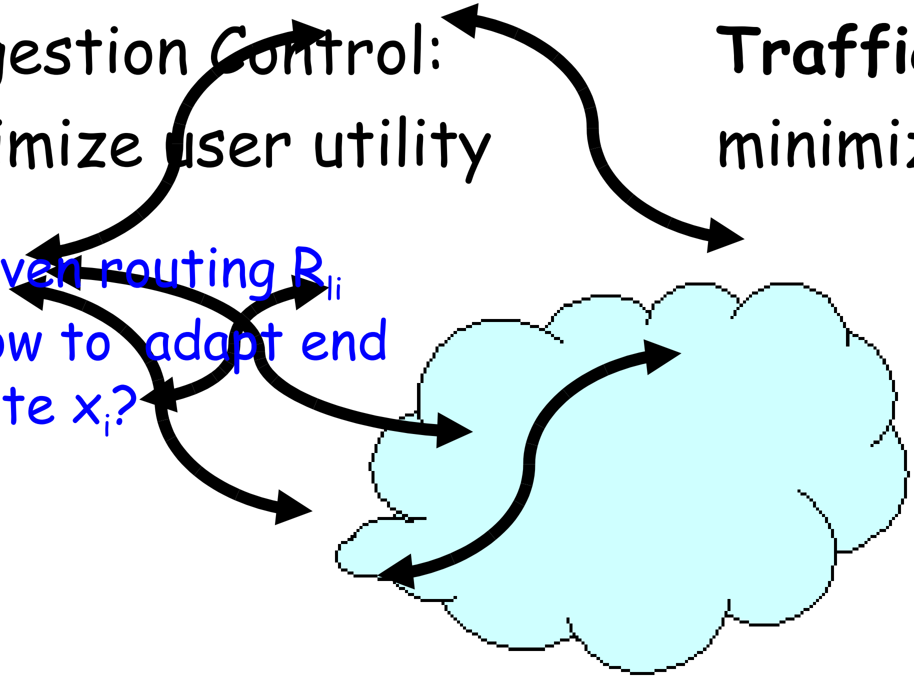
- Results:
  - * converges to solution of “relaxation” of network problem
  - $x_s(t)\Sigma p_l(t)$ converges to w_s

- 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

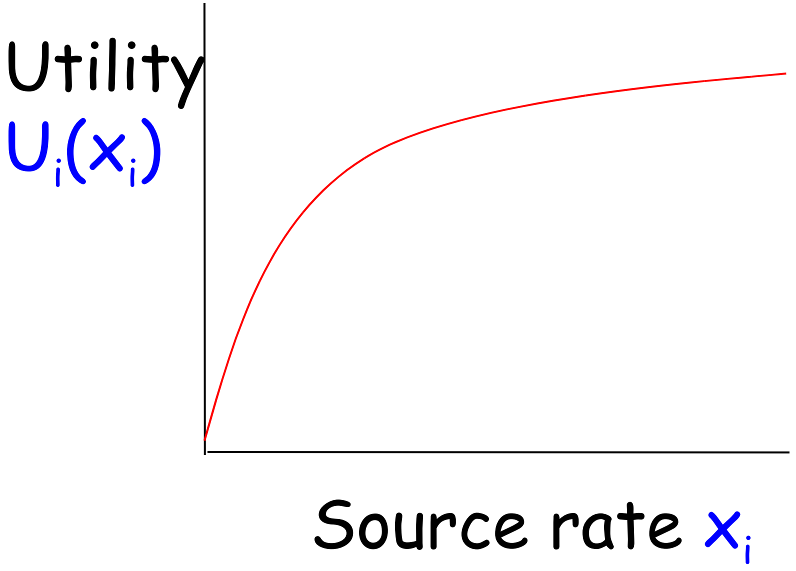
Traffic Engineering:
minimize network congestion

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

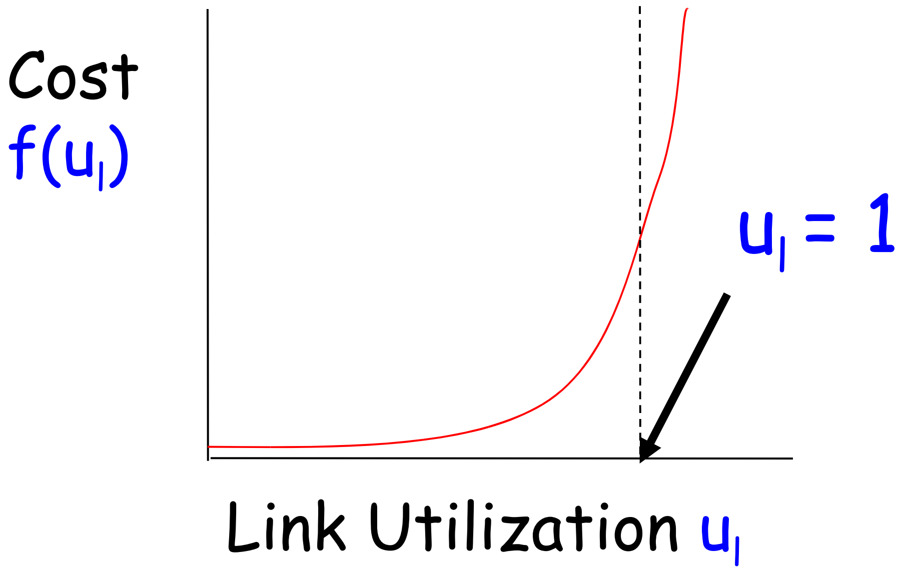
$$\begin{aligned} \text{max.} \quad & \textstyle\sum_i U_i(x_i) \\ \text{s.t.} \quad & \textstyle\sum_i R_{li} x_i \le c_l \\ \text{var.} \quad & x \end{aligned}$$

capacity constraints

Congestion control provides fair rate allocation amongst users

Traffic Engineering Model

Links are indexed by l



aggregate cost

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

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:

$\min \sum_l f(u_l),$

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

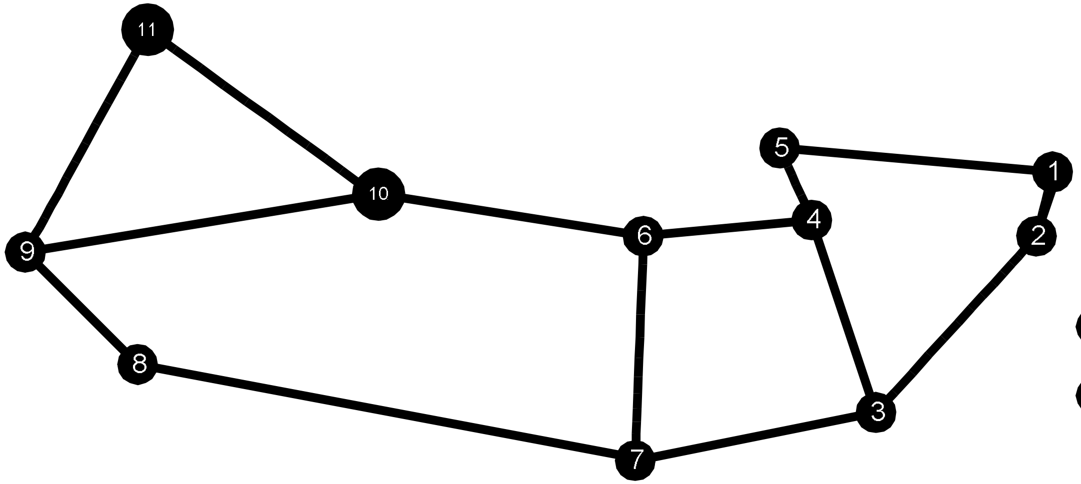
System Properties

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

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

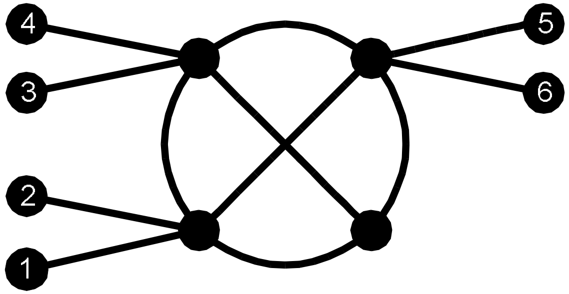
- Utility gap between the joint system and benchmark

Numerical Experiments



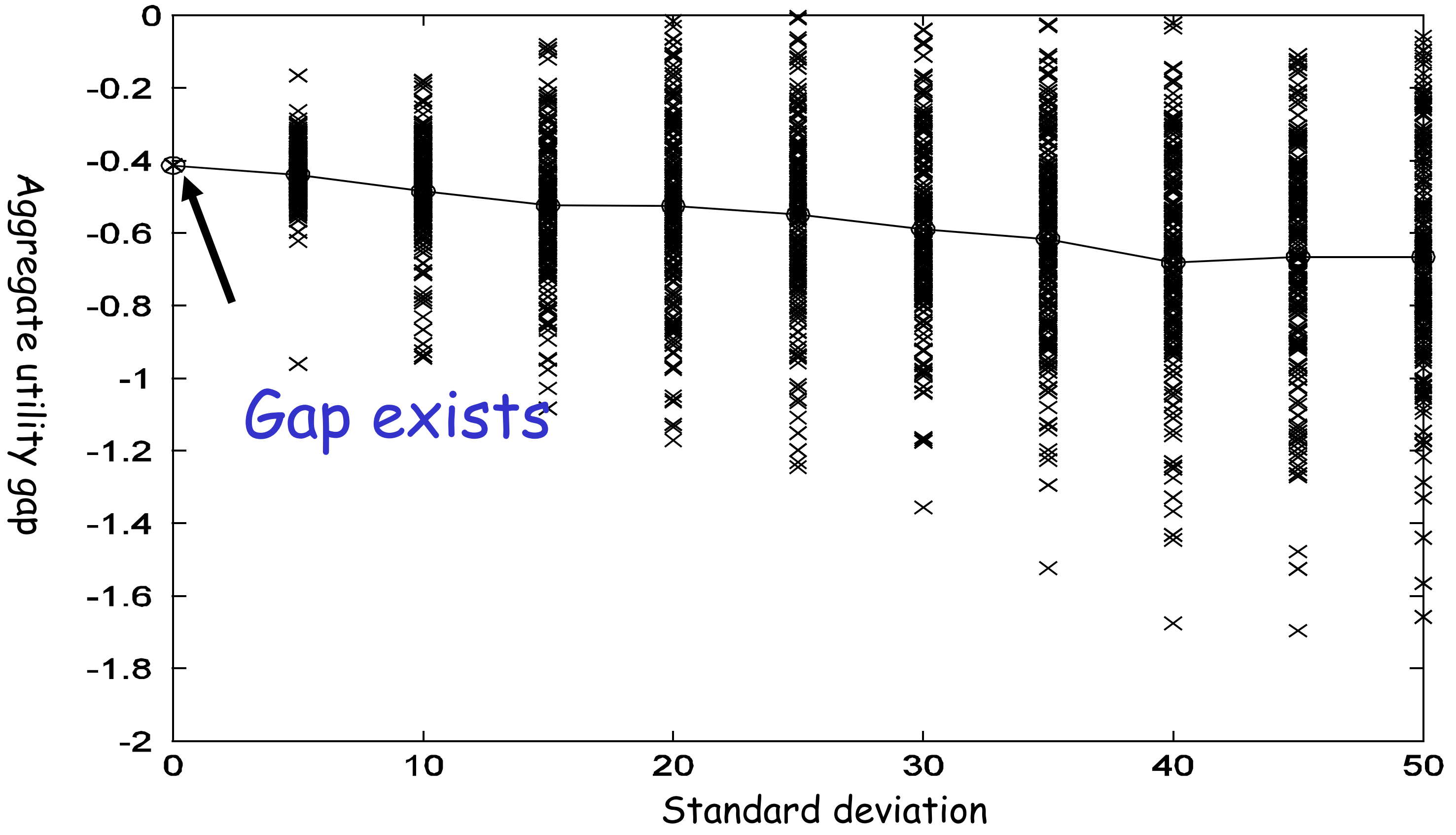
Abilene Internet2

Access-Core



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

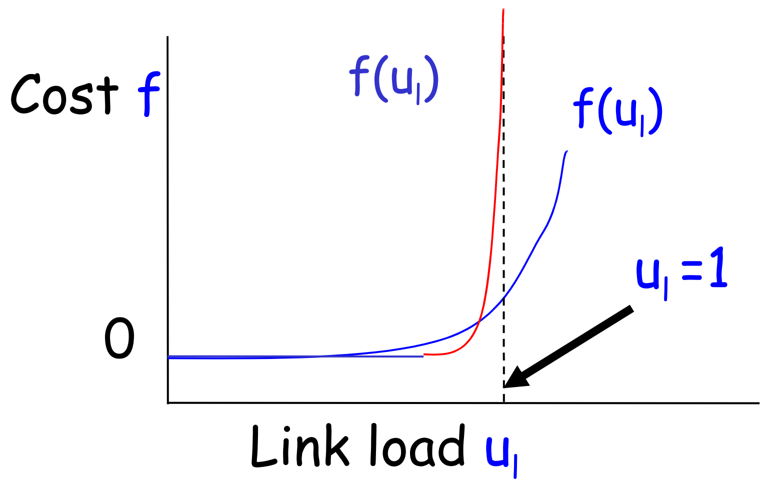
Results for Abilene: $f = e^{u_l}$



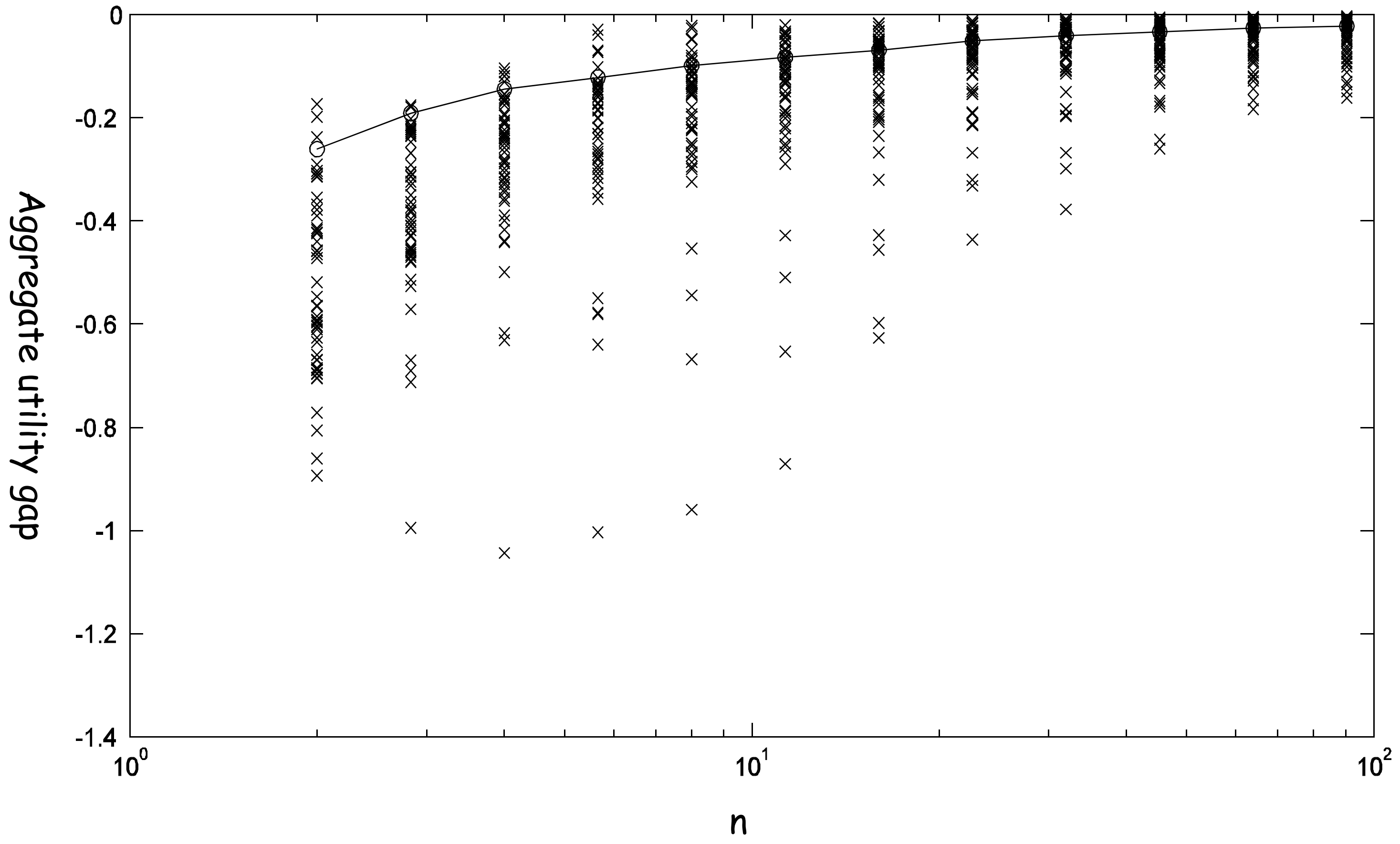
Gap exists

Backward Compatible Design

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



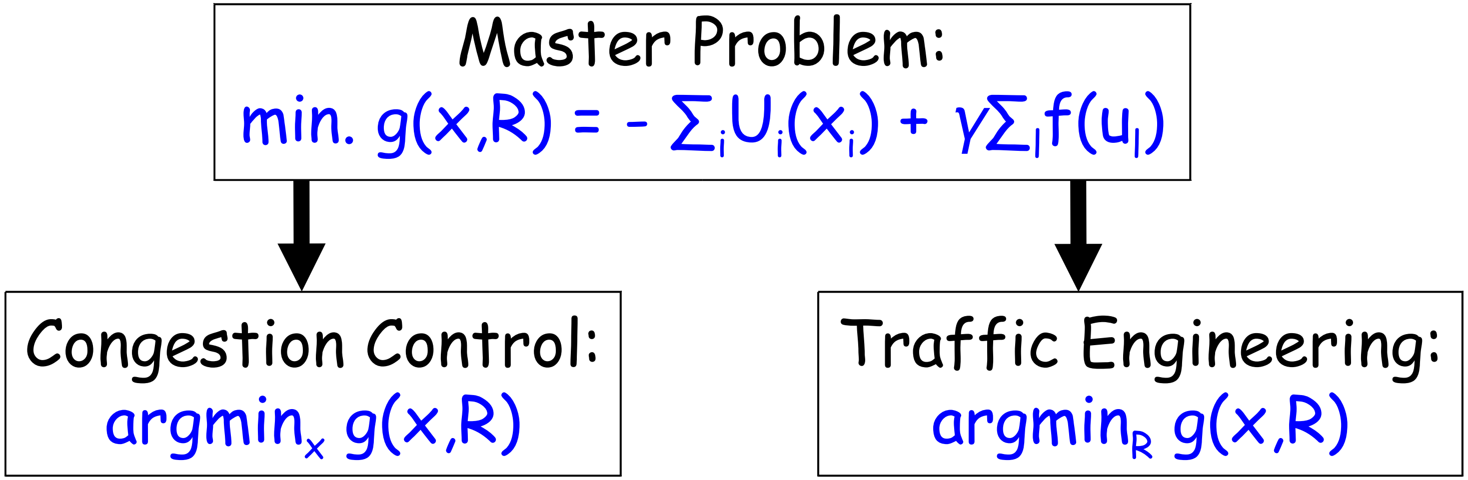
Abilene Continued: $f = n(u_l)^n$



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 $U_i''(x_i) \le -U_i'(x_i) / x_i$



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

Congestion Control:
$\mathrm{argmin}_x\ g(x,R)$

Traffic Engineering:
$\mathrm{argmin}_R\ g(x,R)$

Now re-do for Energy…

- Need energy propotional interfaces from vendor
- Need k-shortest path (& BGP equiv)
- Traffic Manager report aggregate demand
- Configure rates/energy based on aggregate demand
- Future:-
  - put topology&energy* figures (e.g. from JANET) in Xen/VM based emulator
  - See how much we can gain…