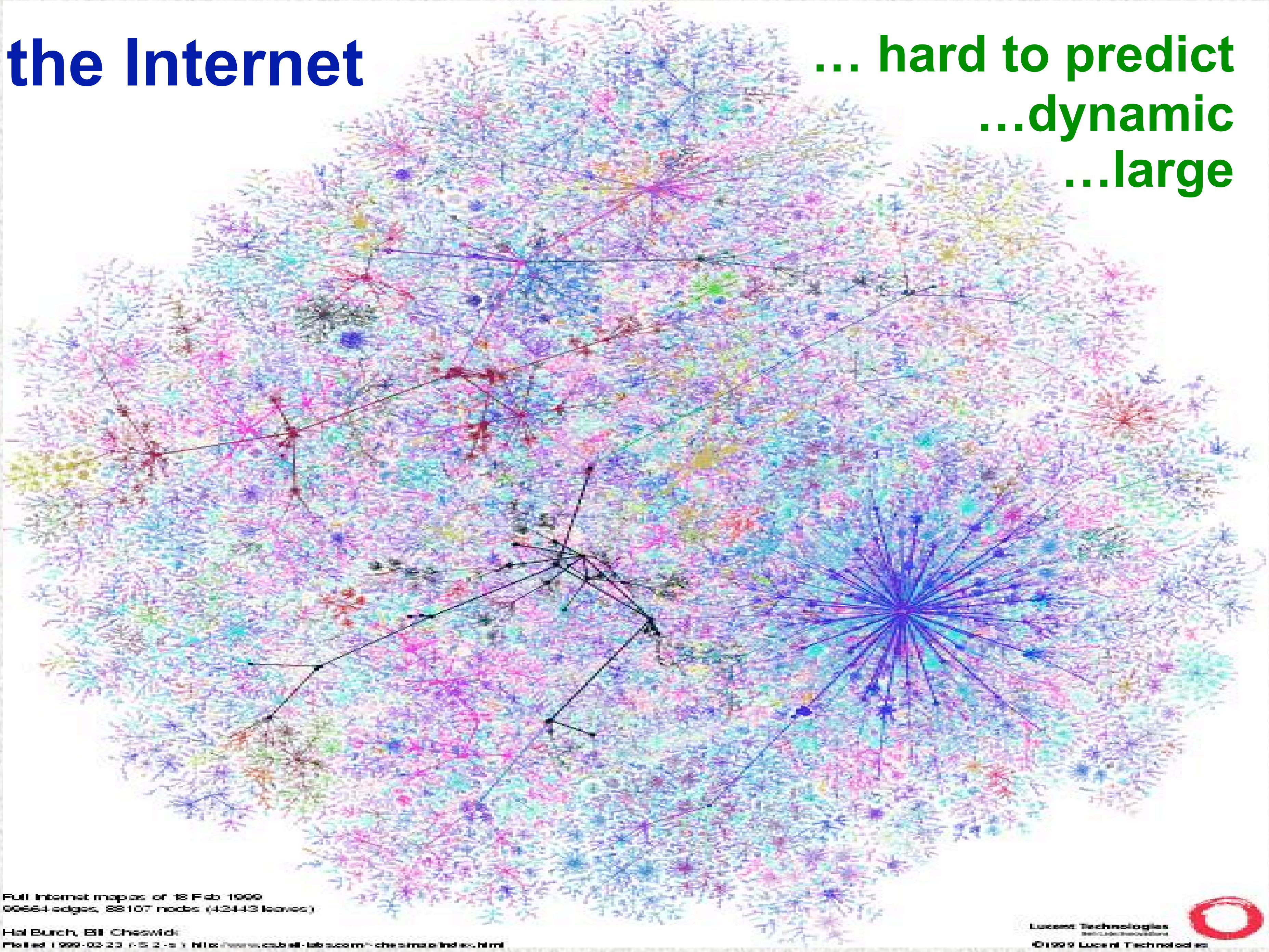
On-line End-to-End Congestion Control

Naveen Garg
IIT Delhi

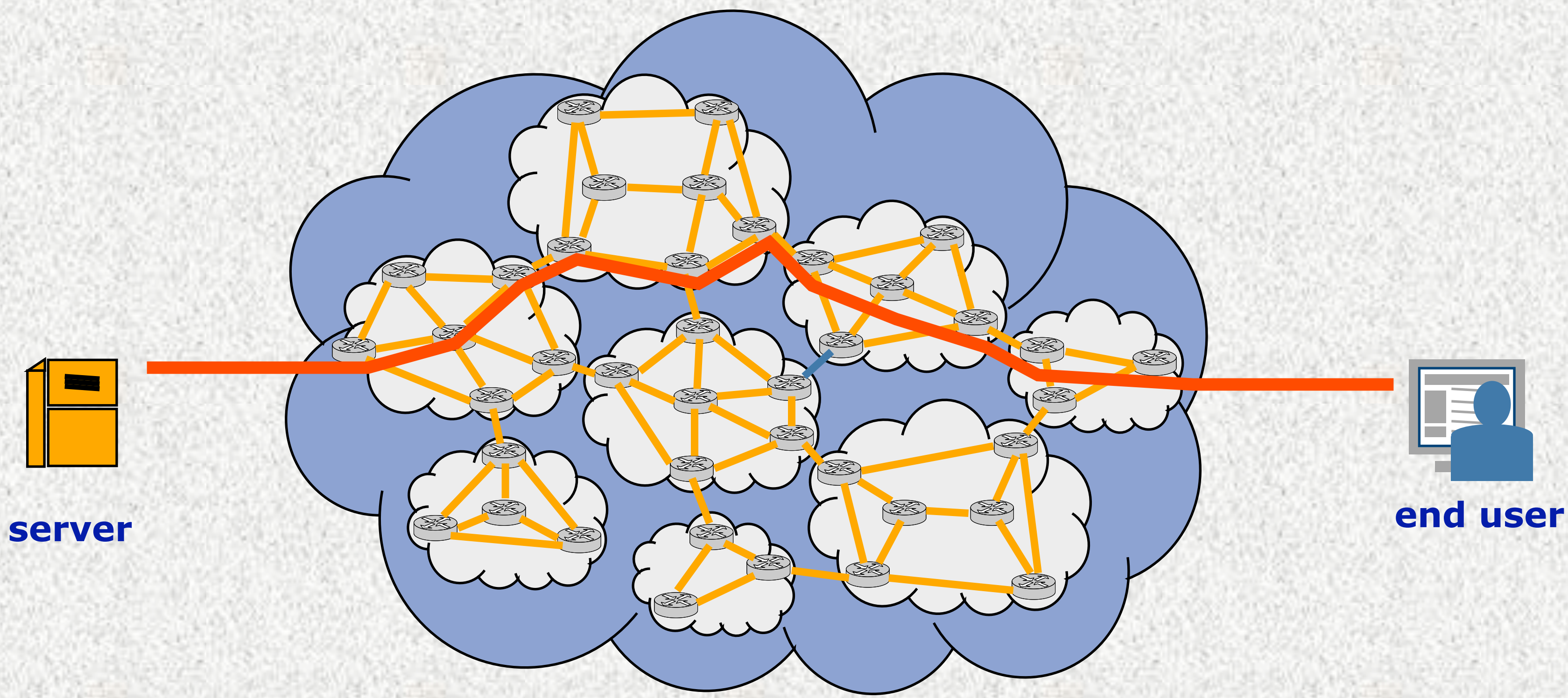
Neal Young
UC Riverside

the Internet

… hard to predict
…dynamic
…large



End-to-end (design principle of Internet)



Routers provide only best-effort packet delivery,
other functionality must be implemented at end-points.
No communication between different paths.

End-to-end congestion control basics

TCP/IP carries the bulk of Internet traffic.
Most connections short-lived.
Most bytes carried by long-lived connections.

What can be said about *global* dynamics induced by an end-to-end protocol?

Stability? Efficiency? Fairness?
Large body of existing work.

Framework: protocol maximizes some *global objective*, such as total throughput

Introduction to existing work

Mathematical modeling and control of Internet congestion

Ramesh Johari, SIAM News, volume 33, March 2000.

Internet congestion control: an analytical perspective

Steven H. Low, Fernando Paganini, J. C. Doyle

IEEE Control Systems Magazine, February 2002

Mathematical modeling of the Internet

Frank Kelly, Mathematics Unlimited - 2001 and Beyond, Springer-Verlag 2001

---

typical result: continuous-time analogues of TCP/IP
system of differential eqn's → convergence in limit

Here: explicit performance guarantees, convergence rates.

Lagrangian-relaxation alg's for packing / covering problems

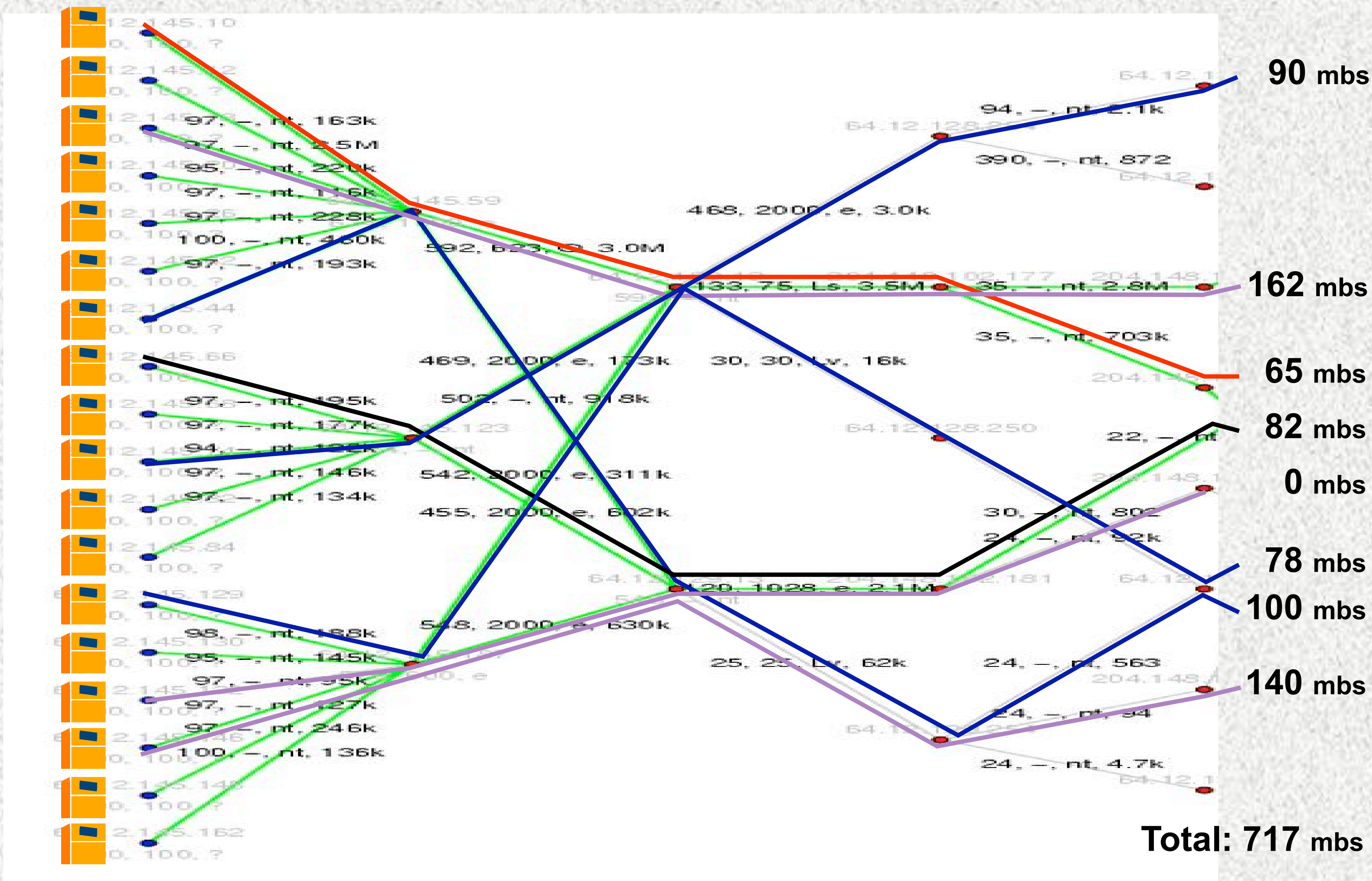
A numerical method for determination... von Neumann 1930
A suggested computation for maximal multicommodity network flow. Ford, Fulkerson 1958
Decomposition principle for linear programs. Dantzig, Wolfe 1960
A linear programming approach to the cutting stock problem. Gilmore, Gomory 1961
The traveling-salesman problem and minimum spanning trees. Held, Karp 1971

The maximum concurrent flow problem. Shahroki, Matula 1990
Fast approximation algorithms for multicommodity flow... Leighton, Makedon, Plotkin, Stein, Tardos, Tragoudas 1993
A simple local-control approximation algorithm... Awerbuch, Leighton 1993
Fast approximation algorithms for fractional packing... Plotkin, Shmoys, Tardos 1995
Randomized rounding without solving the linear program. Y. 1995

Game theory, on-line prediction and boosting. Freund, Schapire 1996
Faster and simpler algorithms for multicommodity flow... Garg, Könemann 1997
On the number of iterations for Dantzig-Wolfe optimization... Klein, Y. 1999
Approximating fractional multicommodity flows… Fleischer 2000
K-medians, facility location, and the Chernoff-Wald bound. Y. 2001
Sequential and parallel algorithms for mixed packing and covering. Y. 2002

Global optimization using local information with applications to flow control Bartal, Byers, Raz 1997

Maximize throughput



Solve it *off-line* if network is known

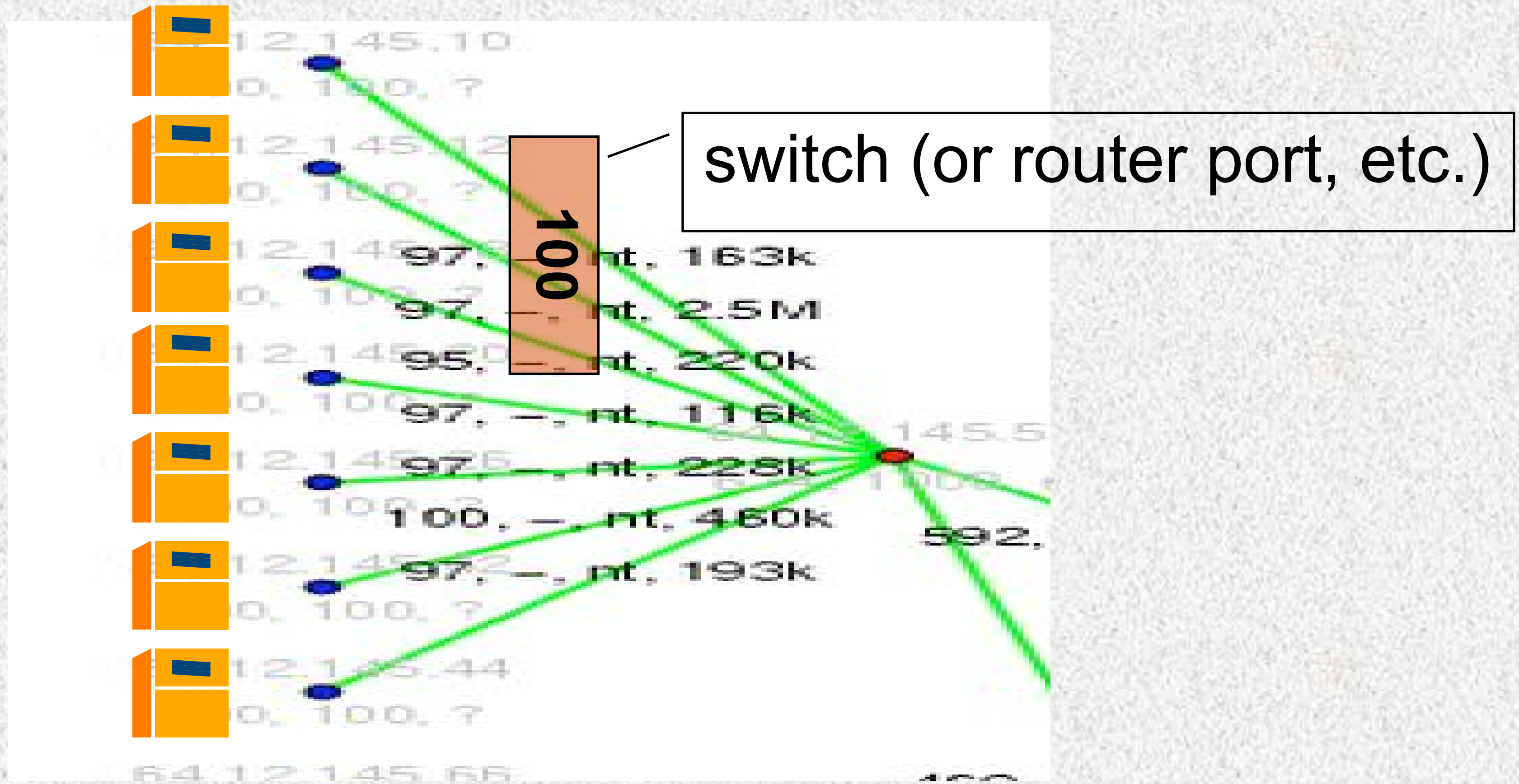
Find flow(p) ≥ 0 for each path p.
Meet capacity constraints (for each edge e)

$$\sum_{p \text{ using } e} \text{flow}(p) \leq \text{capacity}(e).$$

Maximize total flow $\sum_{p} \text{flow}(p)$.

Multicommodity flow -- a packing problem.

Network is not known (hidden bottlenecks)



Challenge

1. Solve it even with hidden bottlenecks?

   -- learning about network only via packet loss

2. Solve it using an end-to-end protocol?

Approach

1. Solve it even with hidden bottlenecks?

View network as “oracle” for testing if a flow is feasible.

Use Lagrangian-relaxation algorithm.

2. Solve it using an end-to-end protocol?

Implement alg using just rate control, packet loss???

Formal network model: dynamic, hidden

Game played in rounds t = 1,2,3,…

1. Each path chooses its sending rate for round
2. Packets sent induce loads on resources
3. Resources may lose packets if capacities violated
4. Each path learns its own loss for the round

Rate on a path is determined by loss observed on that path

-- "on-line, end-to-end congestion control"

Protocol

Start with arbitrary flow on each path p.

Each round, set sending rate on p to $(1+\varepsilon)$ times previous round's receiving rate.

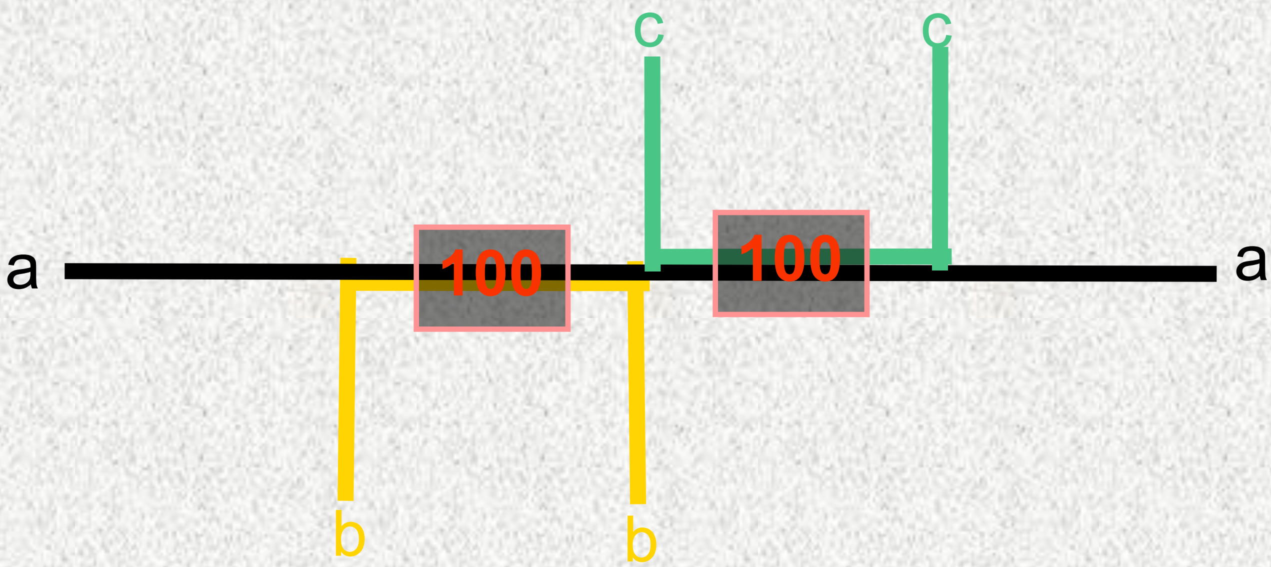
$$sent(p,t+1) = (1+\varepsilon)*received(p,t)$$

ε is a global constant.

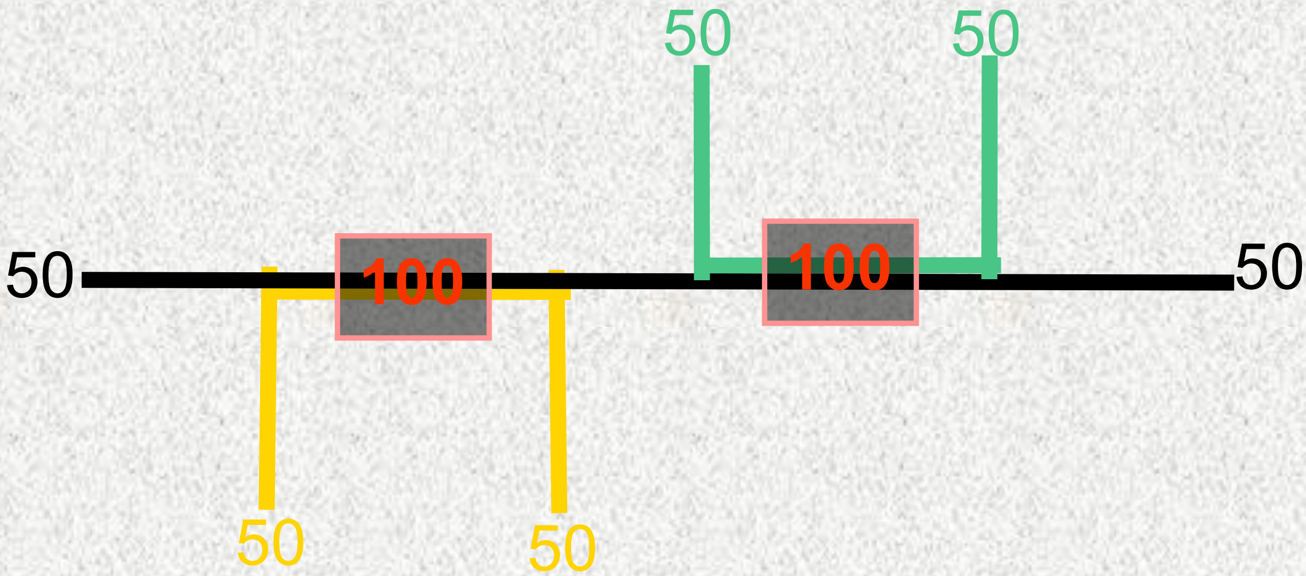
Equivalent to:

$$sent(p,t+1) = sent(p,t)*(1+\varepsilon)*[1-lost(p,t)/sent(p,t)]$$

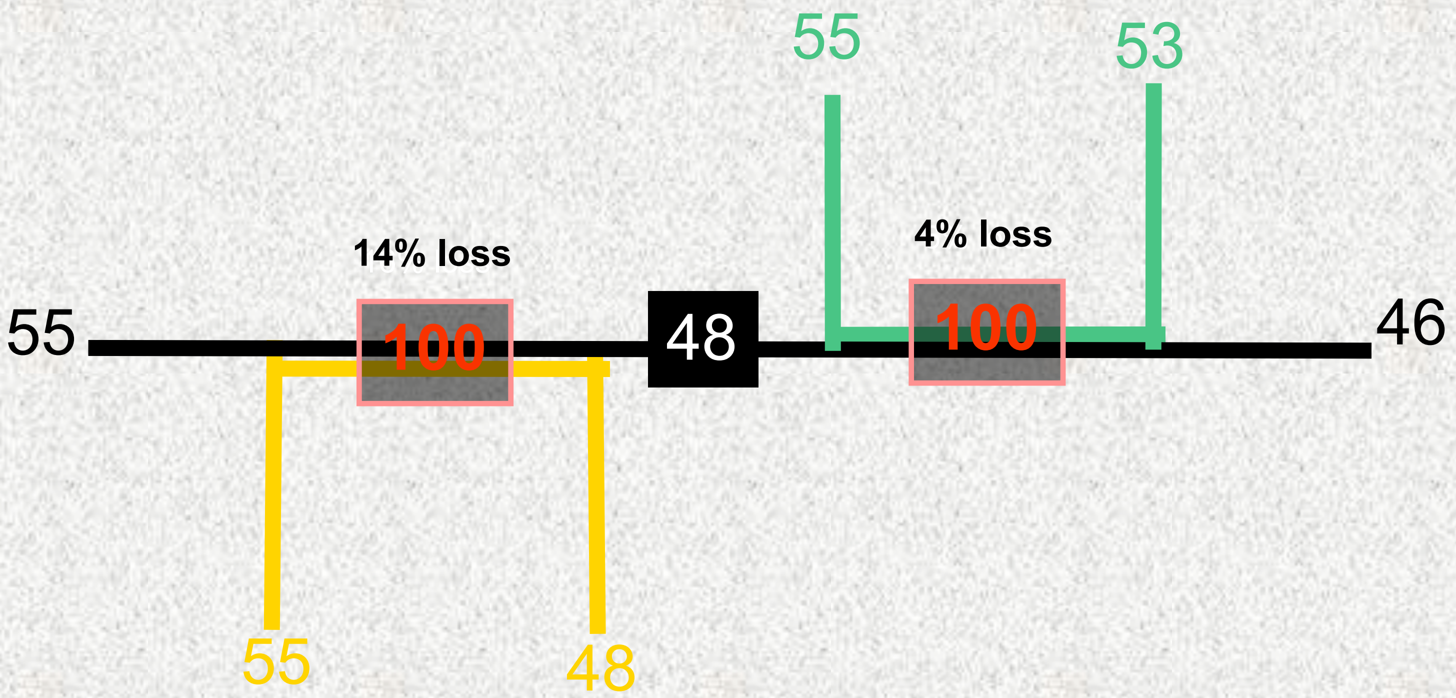
Example



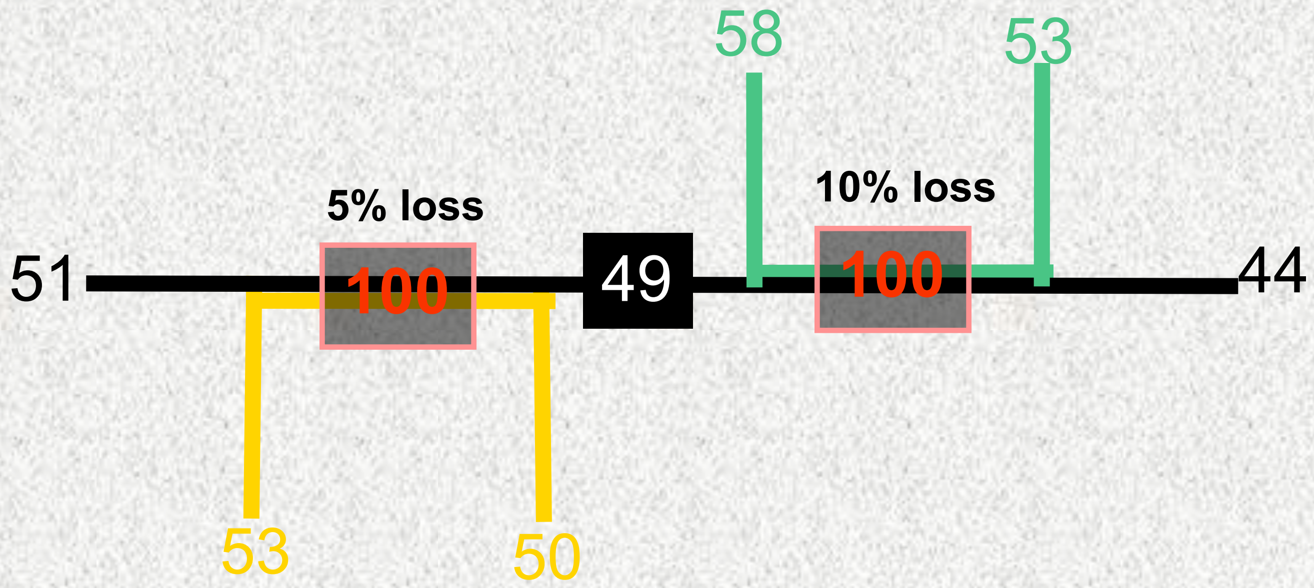
Round 1 (ε = 0.1)



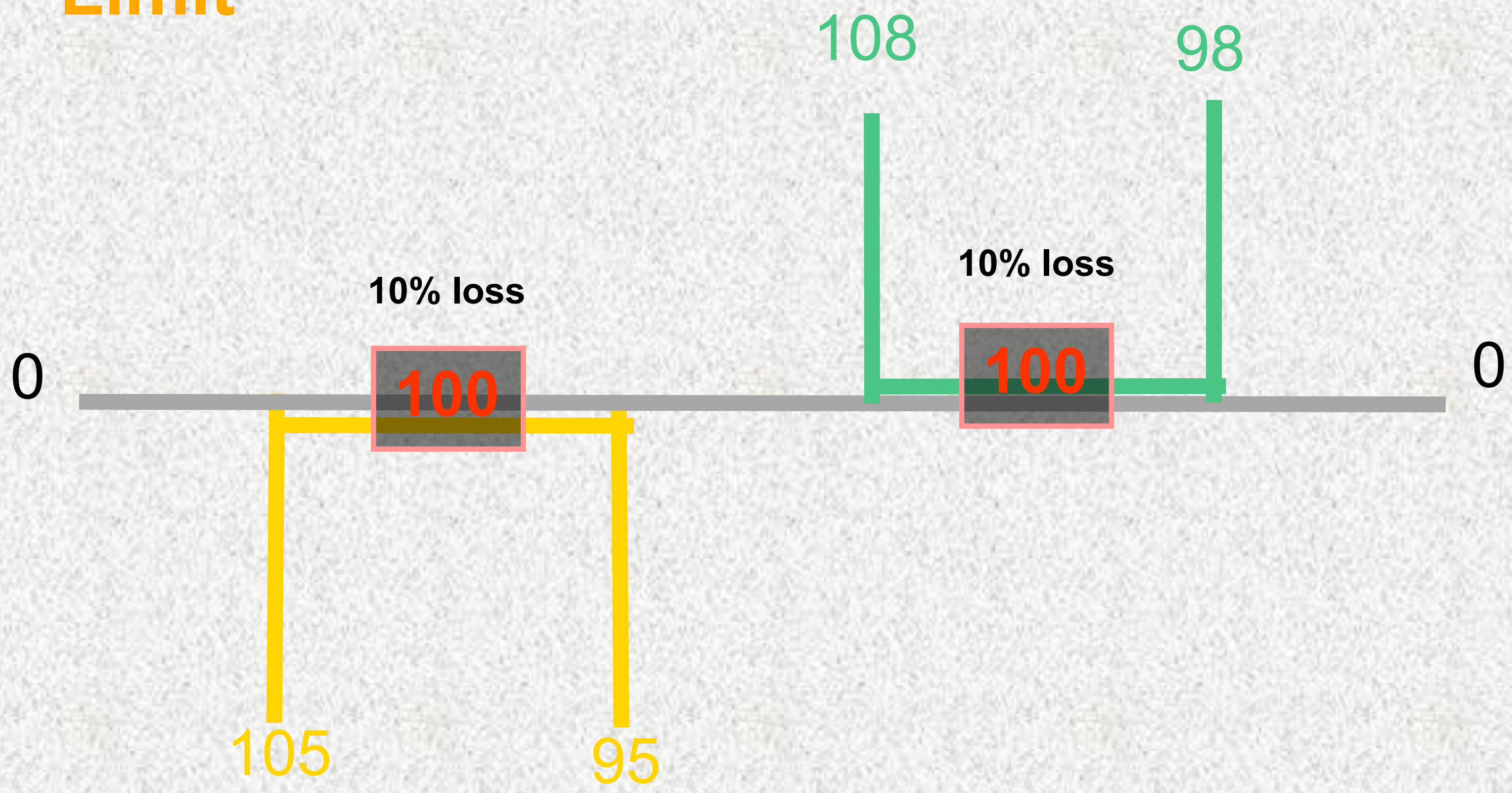
Round 2

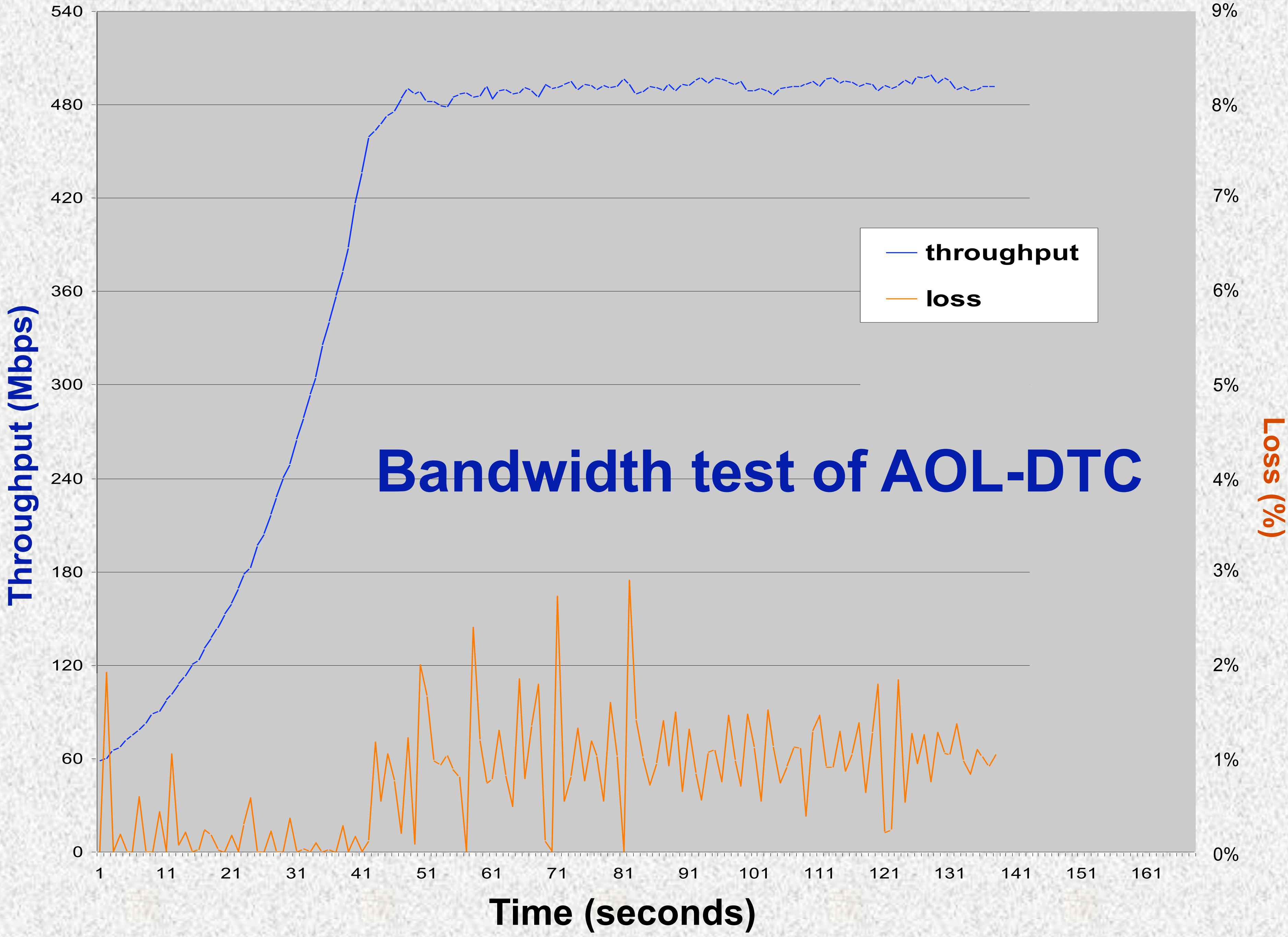


Round 3



Limit




Bandwidth test of AOL-DTC
throughput
loss
Throughput (Mbps)
Loss (%)
Time (seconds)
0
60
120
180
240
300
360
420
480
540
0%
1%
2%
3%
4%
5%
6%
7%
8%
9%
1
11
21
31
41
51
61
71
81
91
101
111
121
131
141
151
161

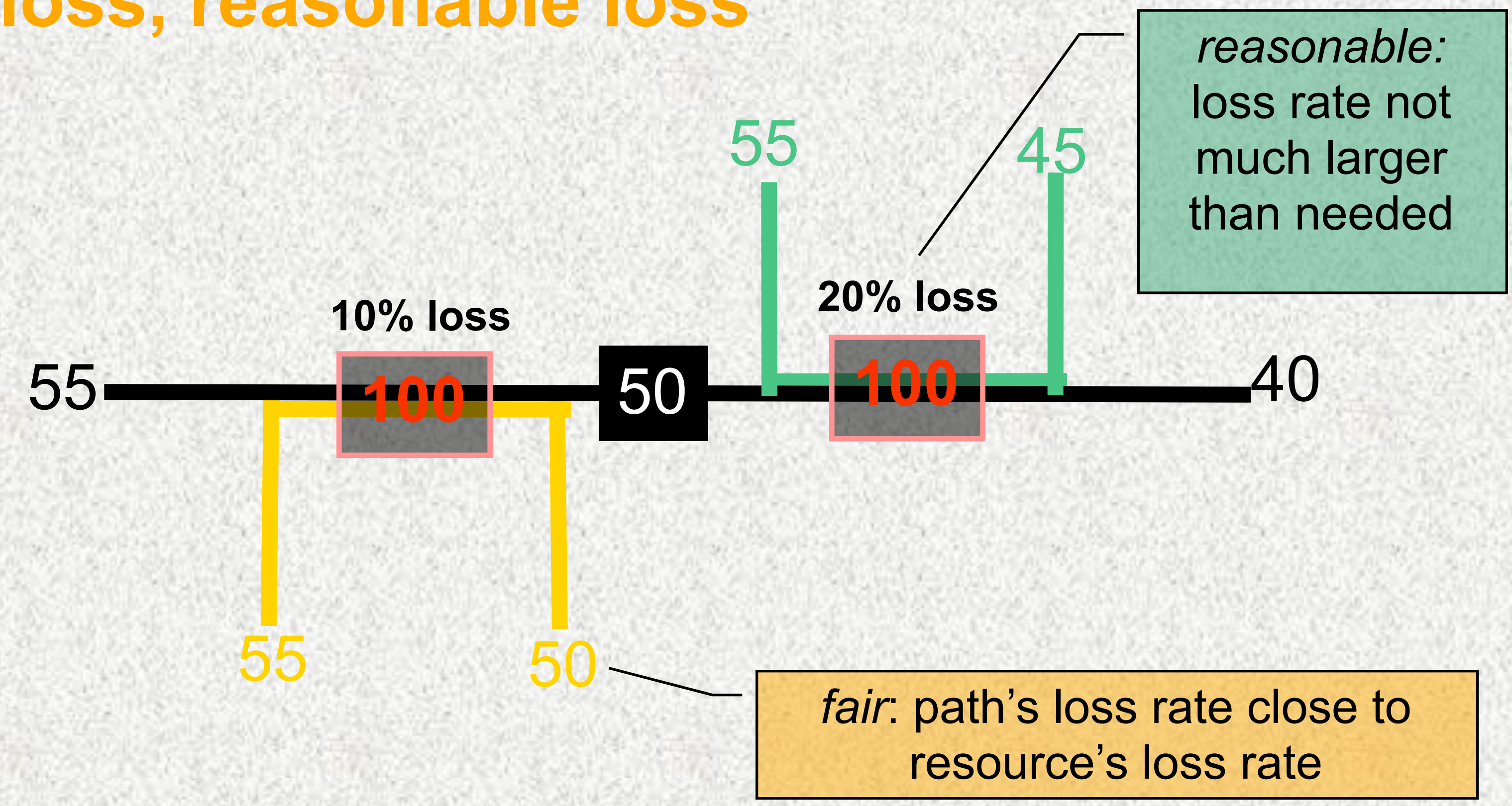
Performance guarantee

THM: Assuming fair and reasonable loss, the total throughput over the first T rounds is at least (1-O(ε)) OPT provided

$$T \geq \varepsilon^{-2} \max_p \log \frac{\text{capacity}(p)}{\text{sent}(p,1)}.$$

OPT = maximum possible throughput,
throughput = bytes received

Fair loss, reasonable loss



Proof sketch

1. *Conservative overall*:

   packets lost $\leq \varepsilon$ * packets received

2. *Aggressive on each path*:

   on each path, average loss ratio over time $\geq \varepsilon - \varepsilon^2$.

3. These two properties, and fair loss, imply

   total throughput $\geq$ (1-O(ε)) OPT.

immediate from protocol

exp. growth of send rate

Linear-programing duality. (dual solution implicitly defined by loss at edges)

1. Overall loss rate at most ε:

want: packets lost $\leq$ ε packets received

easy: sent = $(1+\varepsilon)$*received

sent – received = ε * received

lost $\leq$ ε * received

2. Average loss ratio over time on each path $p \geq \varepsilon - \varepsilon^2$:

want:

$$\sum_{t=1}^{T} \frac{\text{lost}(p,t)}{\text{sent}(p,t)} \geq \varepsilon T - \varepsilon^2 T$$

have:

$$\frac{\text{sent}(p,t+1)}{\text{sent}(p,t)} = (1+\varepsilon)\left(1 - \frac{\text{lost}(p,t)}{\text{sent}(p,t)}\right)$$

assuming reasonable loss

$$\approx \exp\left(\varepsilon - \frac{\text{lost}(p,t)}{\text{sent}(p,t)}\right)$$

$$\exp\left(\varepsilon T - \sum_{t=1}^{T} \frac{\text{lost}(p,t)}{\text{sent}(p,t)}\right) \approx \frac{\text{sent}(p,T)}{\text{sent}(p,1)}$$

$$\approx \exp(\varepsilon^2 T) \qquad \text{... when } T \geq \varepsilon^{-2} \log \frac{\text{sent}(p,T)}{\text{sent}(p,1)}$$

The dual linear program

Fix length(e) ≥ 0 for each edge e.

If, for each path p,

$$\sum_{e \text{ on } p} \text{length}(e) \ \geq \ 1,$$

... then ... $$\text{OPT} \ \leq \ T \sum_{e} \text{capacity}(e)\text{length}(e).$$

remark: The problem of finding lengths that give the best bound on OPT is the same as *fractional set cover*.

Define dual soln   length(e) = (average loss ratio on e over time) / ε'

Feasible?   For all paths p, total length of p at least 1?

Yes, because average loss rate on $p \geq$ ε' (and fair loss).

Dual solution value?

$$\sum_e \text{length}(e)\text{cap}(e) \;\leq\; \sum_{e,t} \text{lost}(e,t)/\varepsilon' T = \text{packets lost } /\varepsilon' T$$

Because cap(e) < entered(e,t) when lost(e,t) > 0,

so (ave loss ratio * cap) < ave loss.

Conclusion:

packets received $\geq$ packets lost / ε

$\geq$ dual solution value * ε' T / ε

$\geq$ OPT ε' / ε = OPT (1-ε).

Generalized protocol

For lossy networks

$$\text{send}(p,t+1) = (1-\alpha_p)\,\text{send}(p,t) + \alpha_p(1+\varepsilon)\,\text{receive}(p,t).$$

Per-path reactivity control.
Works in presence of unreasonable loss if $\alpha_p \leq \varepsilon$.
Convergence slower by a factor of $\max_p 1/\alpha_p$.

For quality of service (weighted throughput)

$$\text{send}(p,t+1) = (1-\alpha_p)\,\text{send}(p,t) + \alpha_p(1+\varepsilon\, v_p)\,\text{receive}(p,t).$$

Maximizes total value-weighted flow: Σ flow$(p)*v_p$

More realistic network model

- capacities vary with time
- connections start and stop
- round-trip times (packet latency)
- approximate fair loss over time

THM: Total throughput is (1-O(ε)) OPT if every connection lasts at least T rounds,

$$T \geq \varepsilon^{-3} \max_p \log \frac{\text{capacity}(p)}{\text{sent}(p,1)}$$

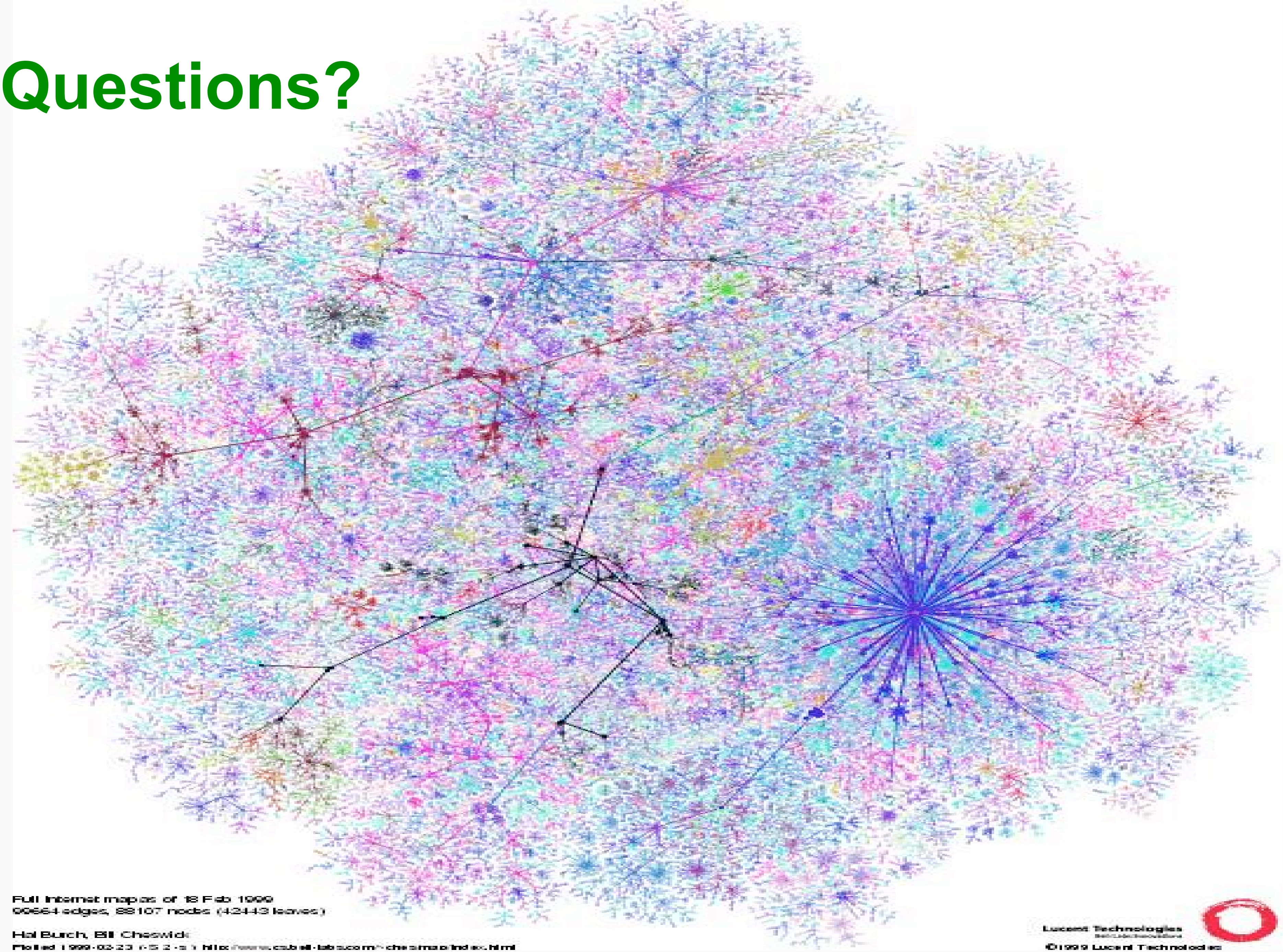
OPT restricted:

cannot vary rate on a connection while connection is active.

Directions and open questions

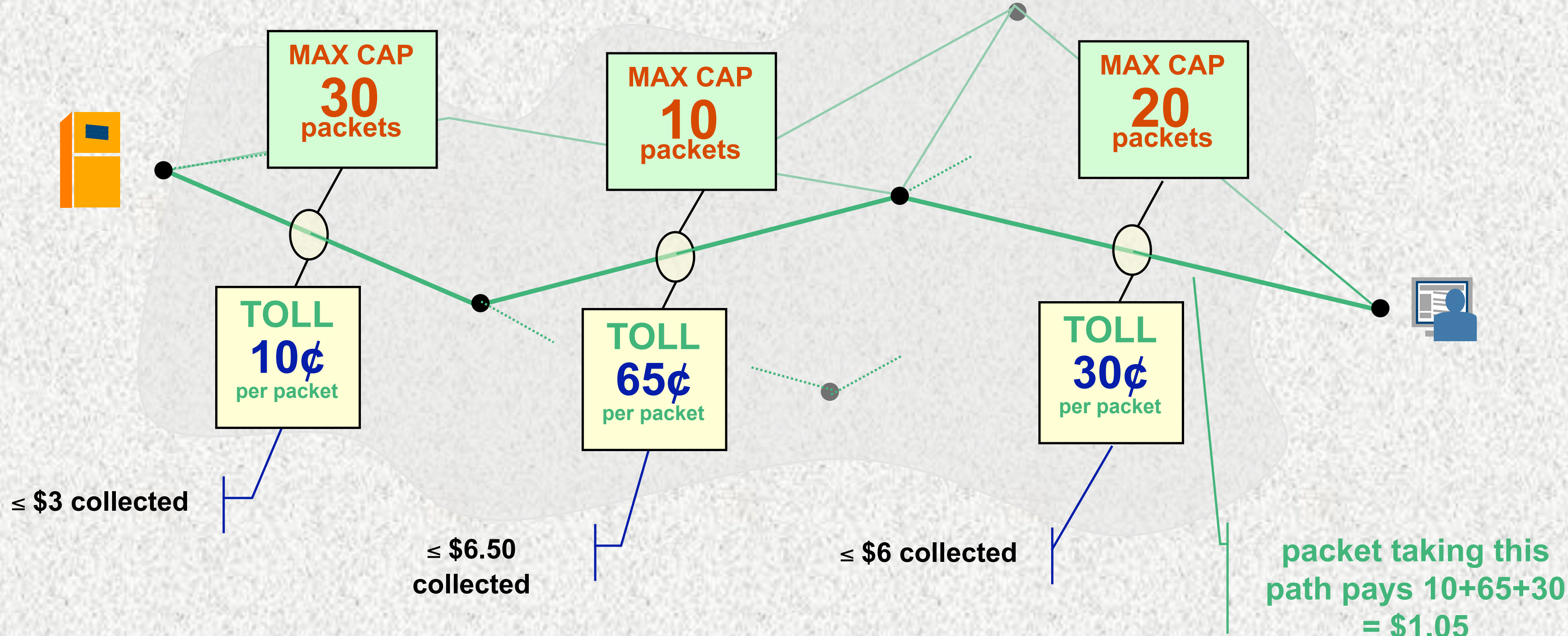
- lower bounds on convergence rates
  in end-to-end model (competitive analysis)
- other objective functions (proportional fairness)
- routing
- other *dynamic, hidden-information* optimization problems
  (e.g. min-cost assignment with hidden, varying demands
  -- one component of Akamai's network-wide load-balancing)

Questions?


Full Internet maps of 18 Feb 1999
90664 edges, 88107 nodes (42443 leaves)
Hal Burch, Bill Cheswick
Lucent Technologies
©1999 Lucent Technologies

Interpretation of l.p. duality using *tolls*:

Select edge tolls such that (for each path p) $\sum_{e \text{ on } p} \text{toll}(e) \geq 1$...



$$\text{OPT} \leq \text{total } \$\text{collected} \leq \Sigma_r \text{ toll}(r)*\text{cap}(r)$$