

# On the Sensitivity of Network Simulation to Topology

Kostas G. Anagnostakis

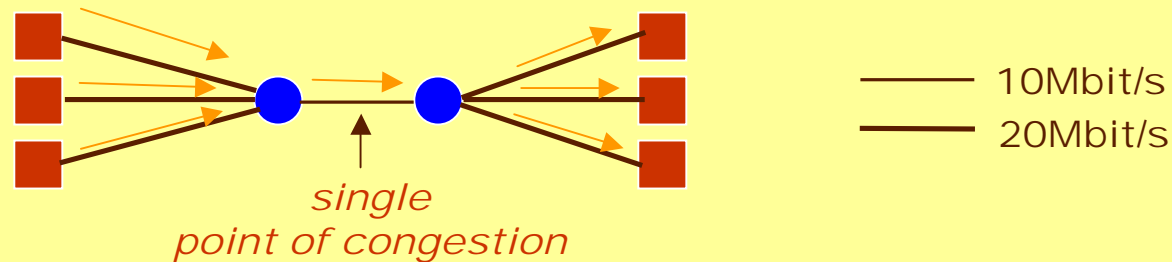
CIS Department - University of Pennsylvania

`anagnost@dsl.cis.upenn.edu`

joint work with Raphael S. Ryger (Yale)  
and Michael B. Greenwald (Penn)

# Network congestion in Spyce

- Goal is to identify **opportunities** for diffuse computing in **congestion control**
- Evaluation of congestion control protocols assumes simple **barbell topology**

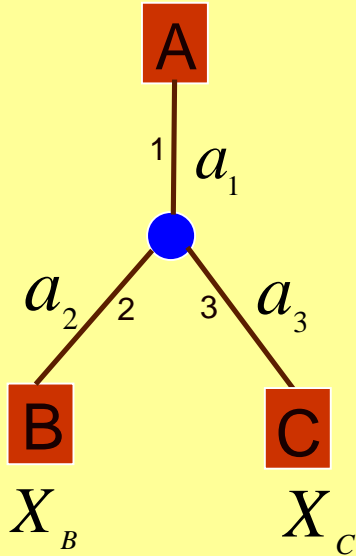


- Test this assumption by **measuring the Internet**
- Determine **implications** on modeling, simulation, and protocol design

# Measuring Internet congestion points

- What is the fraction of network paths with multiple congestion points ?
- Indicators of congestion are **delay** and **loss**
- But direct access to router stats or large-scale instrumentation not practical
- Alternative is **indirect**, end-to-end probing, termed **network tomography**

# Indirect Network Tomography



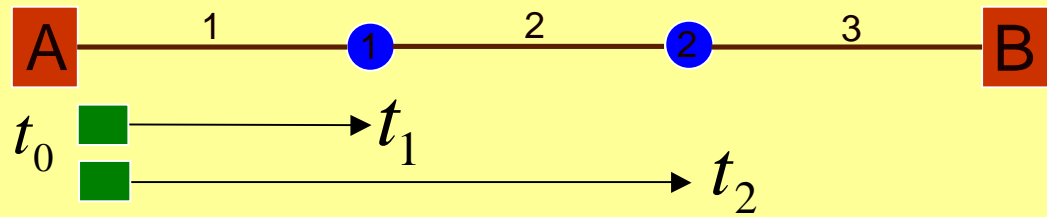
- Given end-to-end measurements  $X$   
Estimate distrib. of link delays  $a$
- Formulated as a **MLE** problem,  
Solved using **EM** algorithm

- Requires deployment of infrastructure
- Not useful for our study, need alternative

(MINC project @ UMASS, AT&T Research)

# Network tomography: a direct method

- We designed a direct **packet-pair** technique using router **ICMP Timestamp**



$$t_2 - t_1 =$$

$$(t_0 + d_1^{prop} + d_1^q + d_2^{prop} + d_2^q) - (t_0 + d_1^{prop} + d_1^q) =$$

$\underbrace{d_2^{prop}}_{\text{fixed}} + \underbrace{d_2^q}_{\text{variable}}$

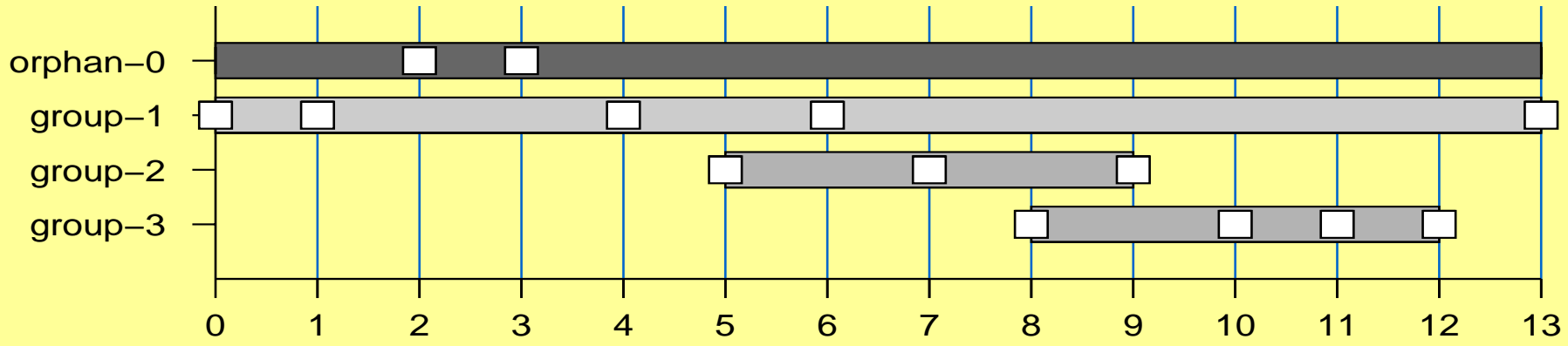
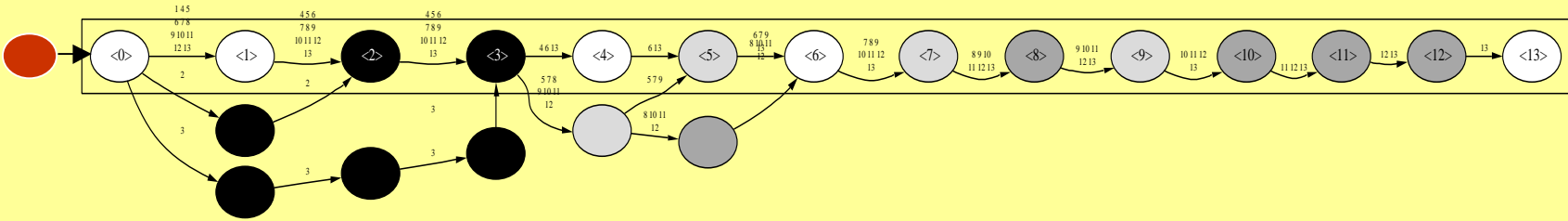
← Queueing Delay on link 2!

$d^{prop}$  : propagation delay  
 $d^q$  : queueing delay

# Network Tomography: feasibility

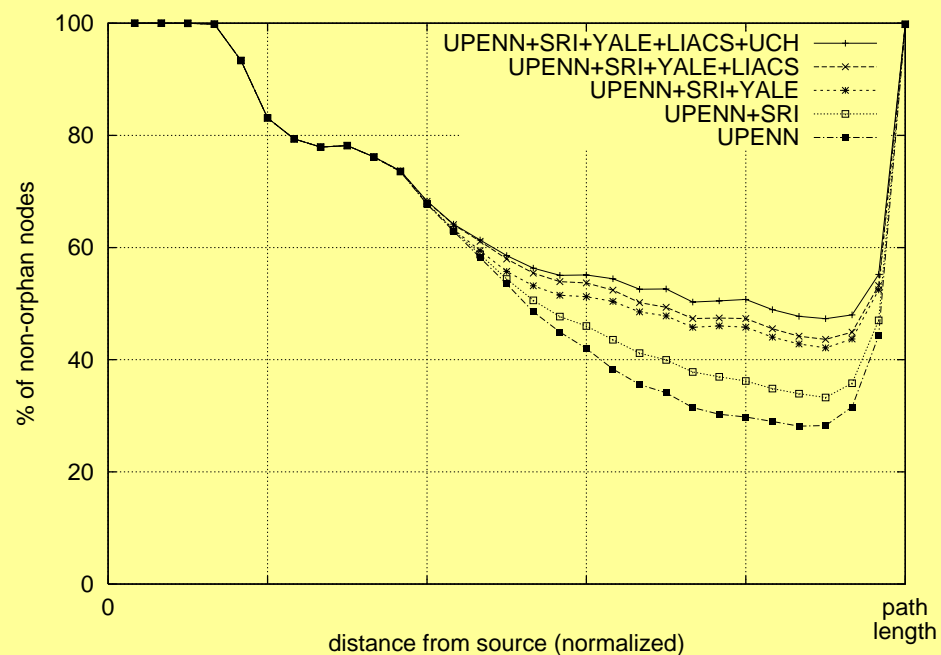
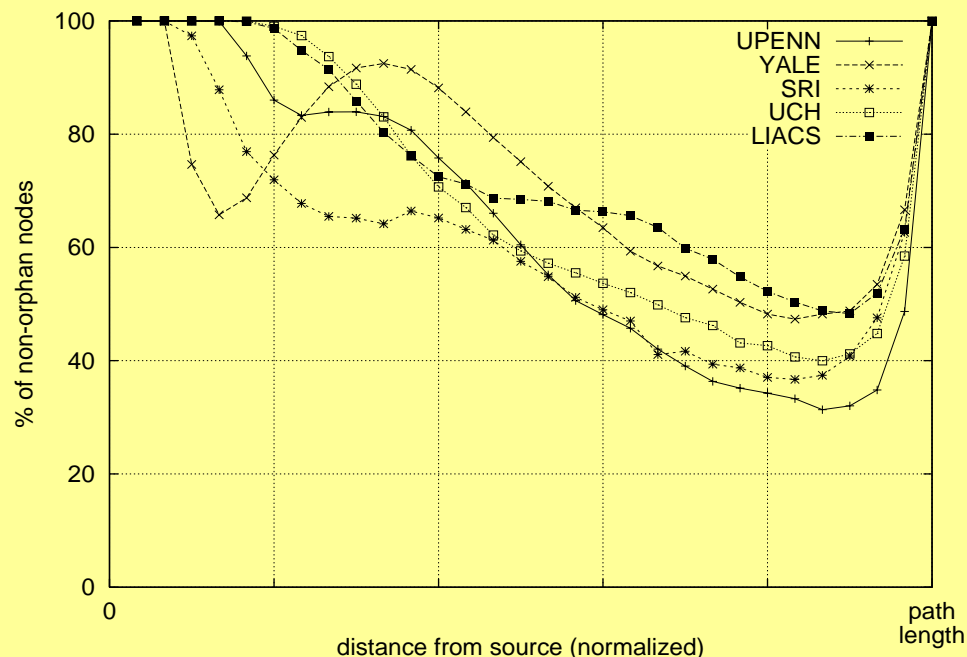
- 96% of nodes respond to Timestamp queries
- Irregular routing limits choice of nodes

Example: Path structure, Penn to Sprintlabs



Corresponding feasible measurement partitions, Penn to Sprintlabs

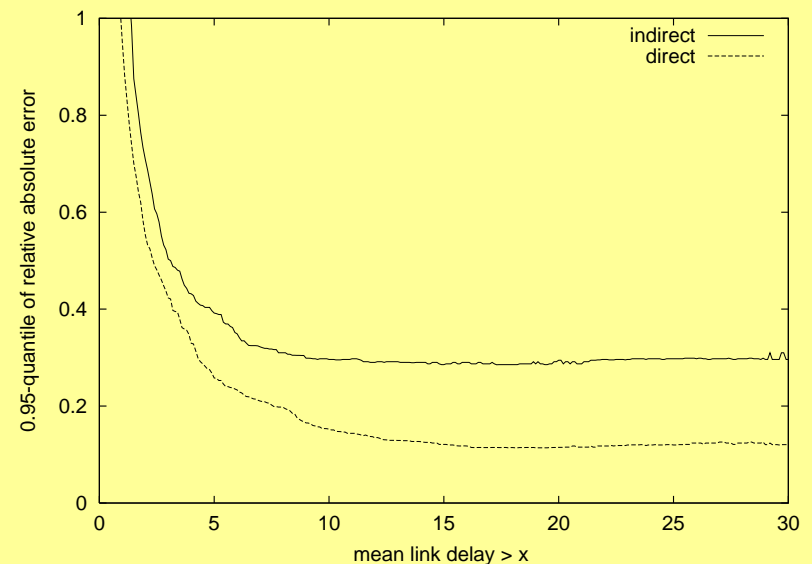
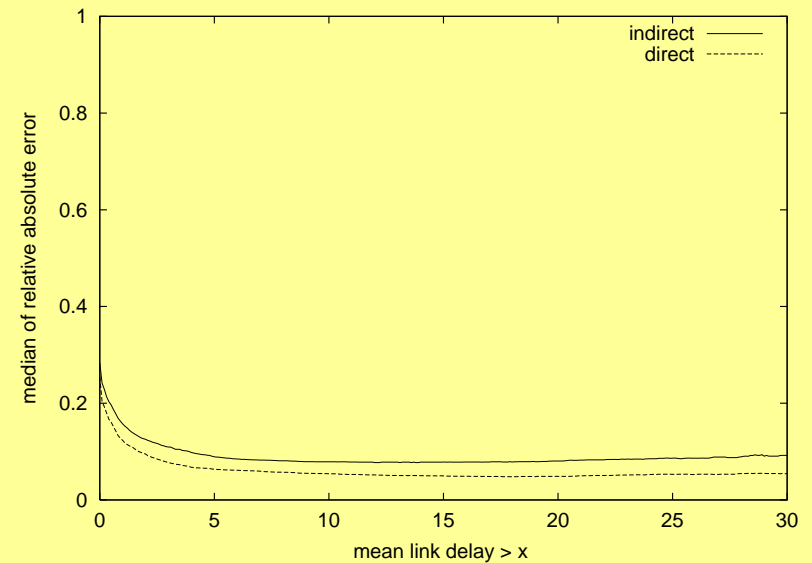
# Network tomography: feasibility (2)



- **Data:** ~10k paths from 5 different sources
- **Metric:** fraction of nodes usable for tomography
- **Results:** ~50% nodes are usable, more difficult as distance from source increases, better when probing from multiple sources

# Network tomography: performance

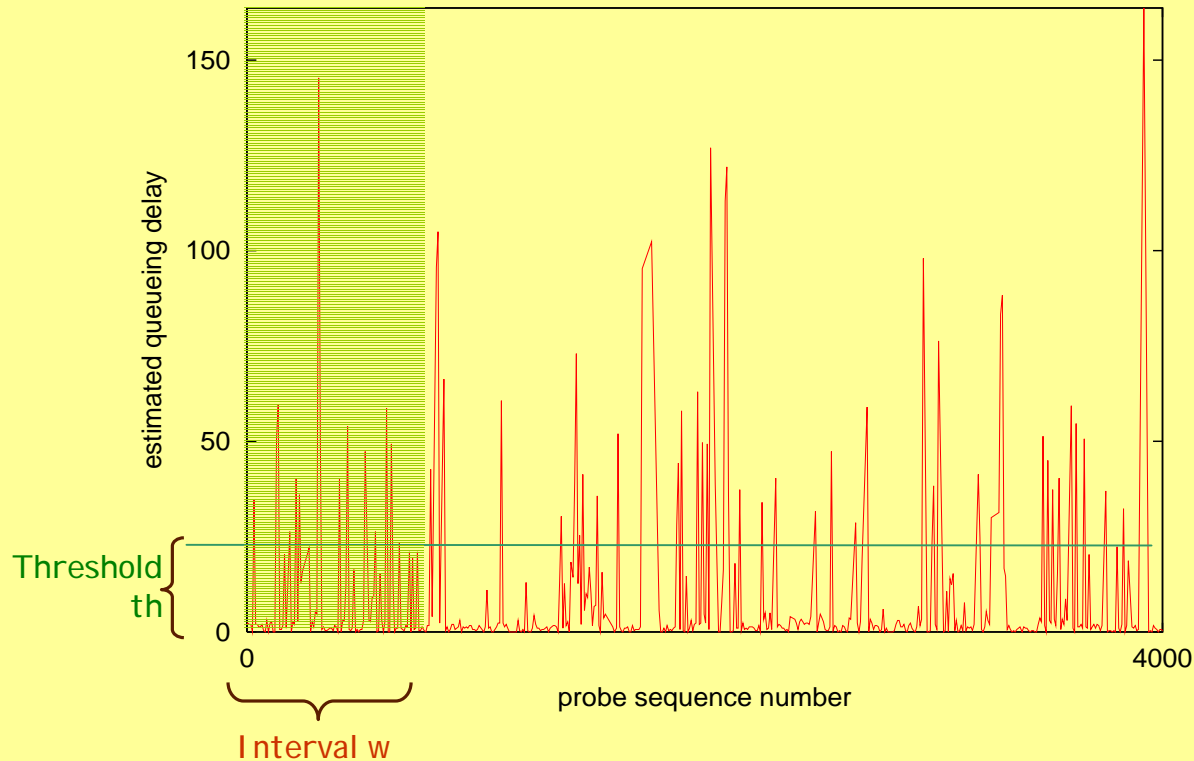
- Conducted **simulation** study to compare direct and indirect method
  - **Result**: Direct technique is slightly better on average, much better in worst case
  - **Bonus**: Direct technique is simpler to compute
- Conclusion:
  - Coverage of direct method **problem for operations**
  - Coverage and performance **ok for our study**



# Internet path study

- Experiment data:
  - ~50k targets taken from Penn Web server logs
  - Probing from Penn, Yale, SRI nodes
  - ~30k paths screened for congestion
  - ~2k paths chosen for 5-segment tomography
  - Obtained ~30 minutes of data for each path

# Testing a segment for congestion



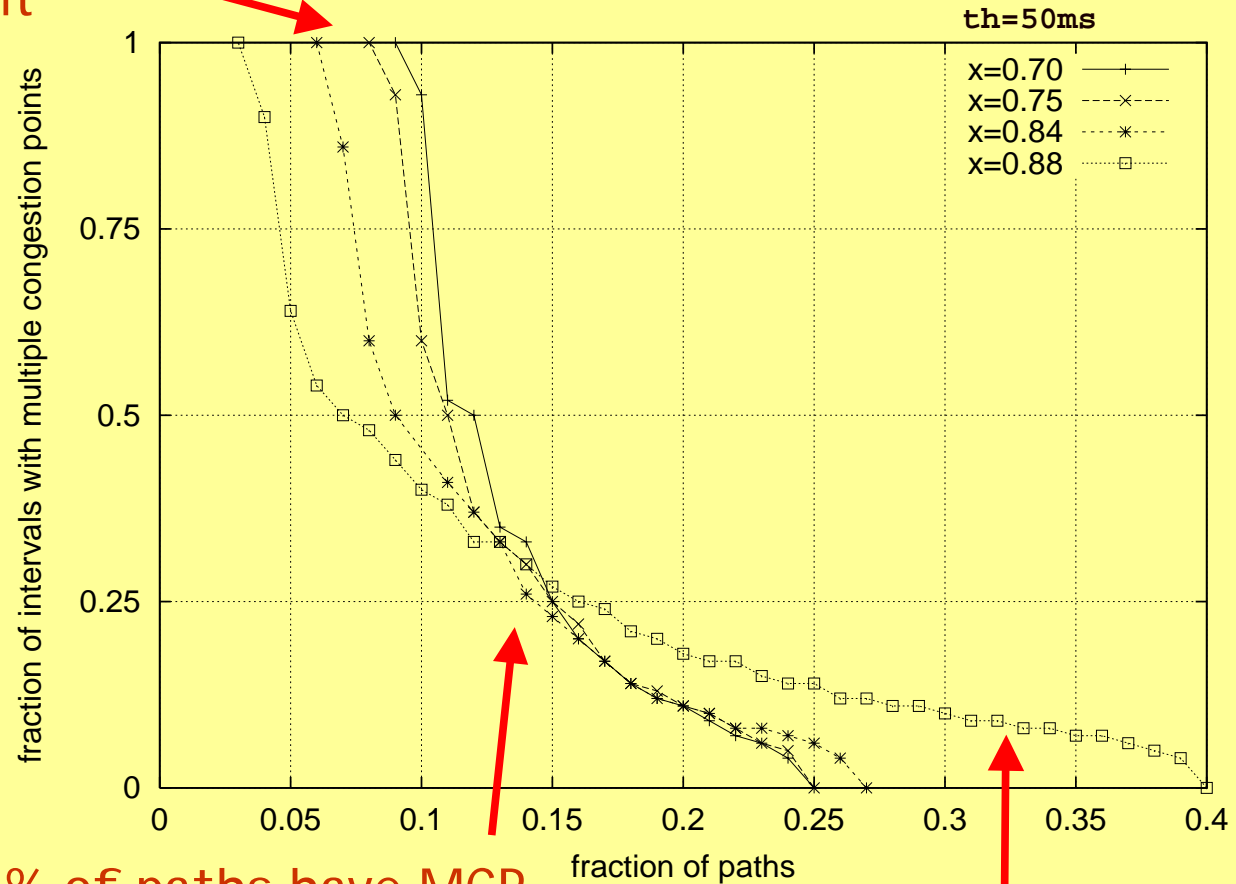
- Need to filter out transient queueing and measurement error but preserve congestion-induced delays
- We use a simple *quantile-threshold* test on queueing delay over time window  $w$
- Choice of parameters  $x$ -th important

# Example: one path



# Results over all paths

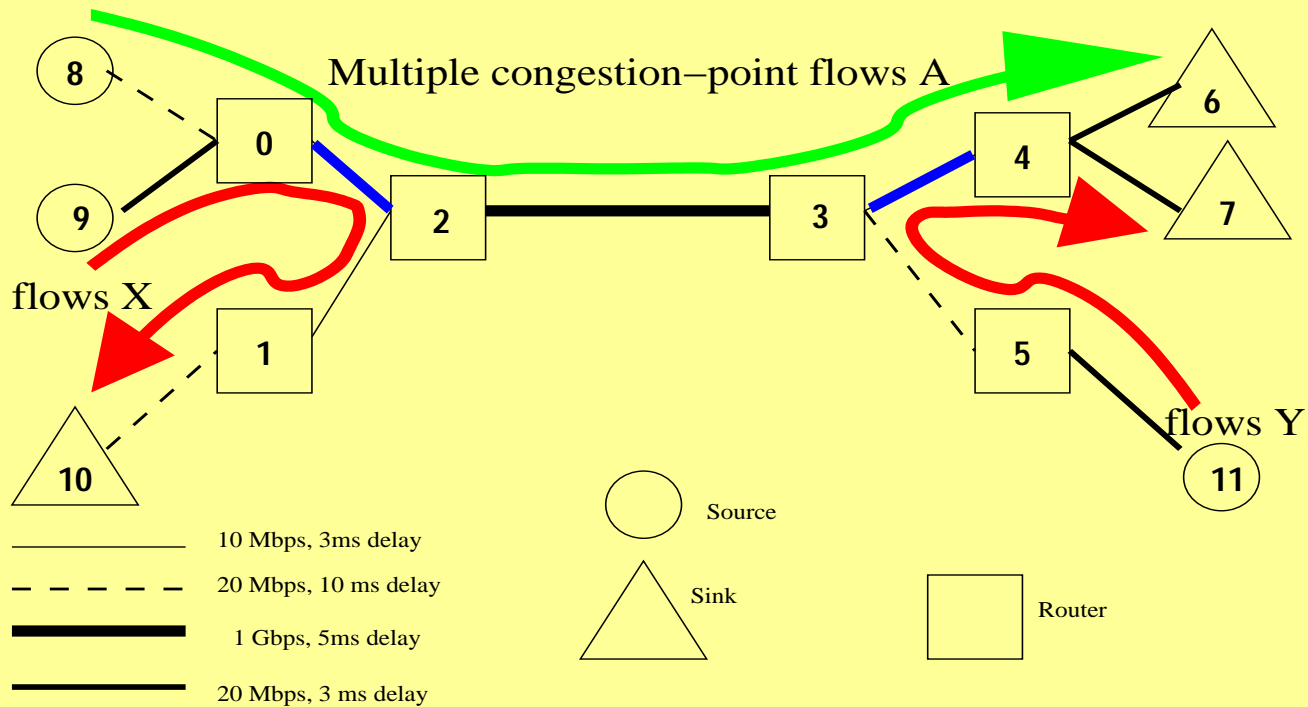
4-10% of paths have MCP throughout the experiment



15% of paths have MCP for a quarter of Measurement intervals

25-37% of paths have MCP at least once during the experiment

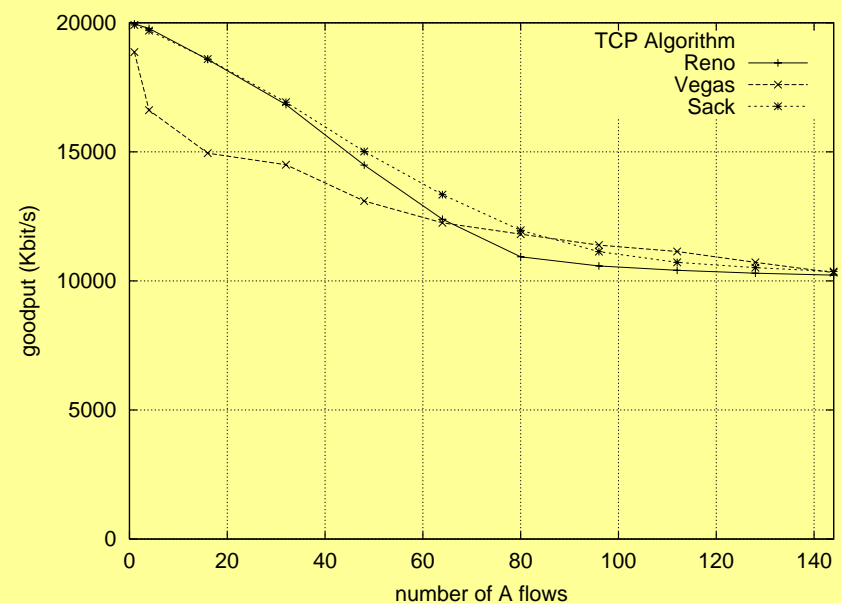
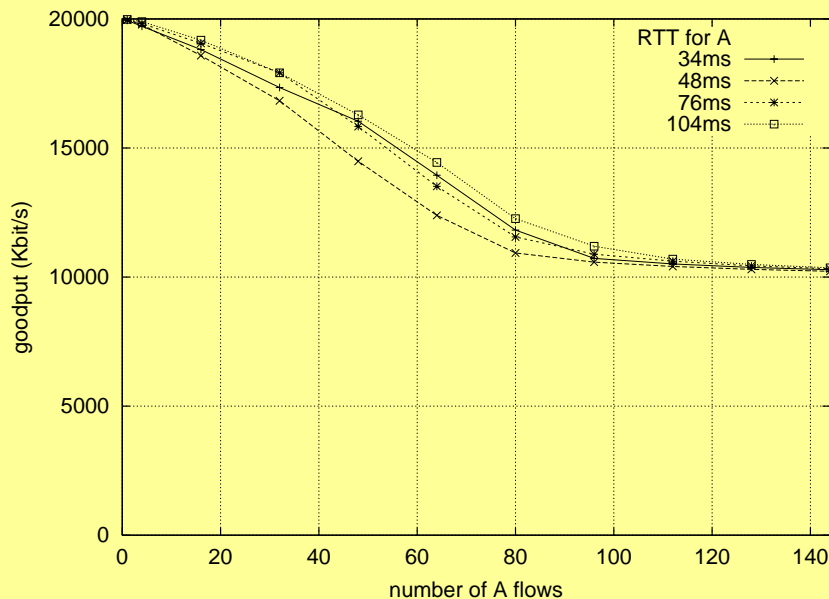
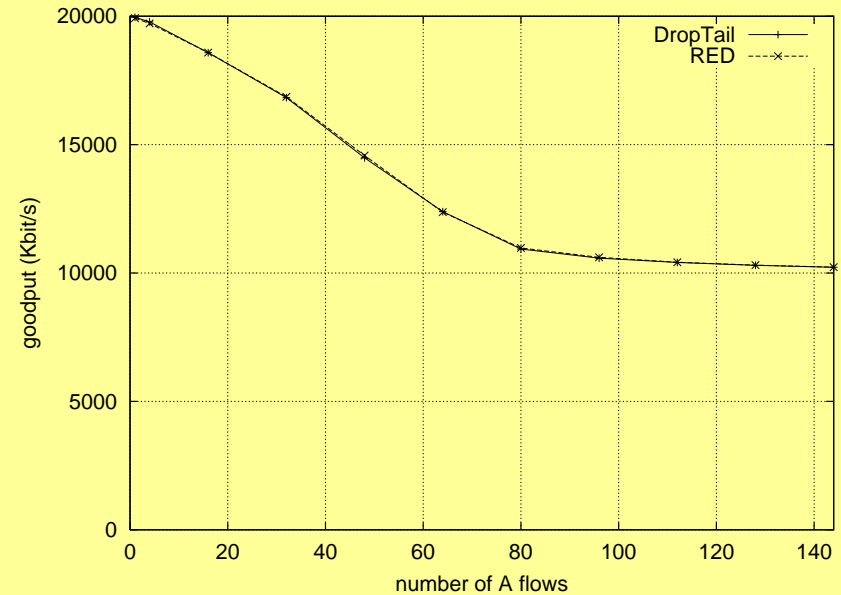
# Simulation scenario



- Fixed # of flows **X** and **Y**, variable # of flows **A**
- Congestion points are links **0-2** and **3-4**
- Metric of interest is aggregate system **goodput**

# Goodput in MCP scenario

- Goodput drops as load increases, technically a form of **congestion collapse**
- **Results pervasive** with changing simulation parameters
- Some TCP variants collapse more rapidly



# Observations

- Simulation of multiple congestion point scenario shows behavior not exposed by the common barbell topology
- Measurements show such behaviors are likely to be found on the Internet
- Problem is that TCP optimizes **locally** on each congestion point resulting in tension between fairness and utility
- Utility-centric design and distributed control may be necessary, pointing to diffuse computing

# Ongoing and future work

- Continue data collection and analysis
- Improve technique (hybrid direct-indirect), to help understand **where** congestion occurs
- Produce congestion control benchmark/regression suite
- Develop (diffuse?) congestion mgmt. model

# Update on Spyce Experimental Platform

- 3 nodes: Penn, Yale and SRI
  - Off-the-shelf setup (OpenBSD, ssh, ...)
  - Good connectivity, high flexibility (no firewalls)
  - Currently used for network tomography experiments (~60GB of data collected so far)
  - Available for other experiments
- 2 spare nodes ready to be shipped out