Rate-based Congestion Control for Short-lived Flows

Nandita Dukkipati, Rui Zhang-Shen, Nick McKeown
{nanditad, rzhang, nickm}@stanford.edu
Computer Systems Laboratory
Stanford University

Problem Motivation

TCP does not work well for long flows

- Gain factor of
- Assumes long flows
- Proposed Solutions: FAST, XCP

TCP does not work well for short flows

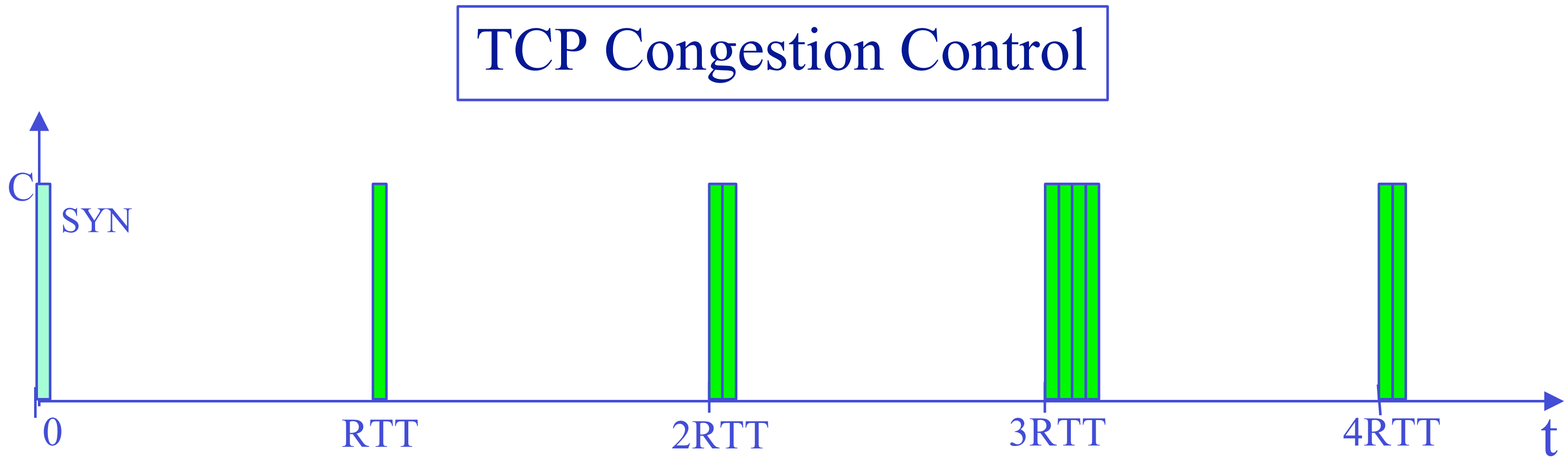
- Short flow: (flow-size)/(bottleneck link capacity) << RTT
- Forces even short flows to last multiple RTTs

Guess: Fraction of "mice" will increase

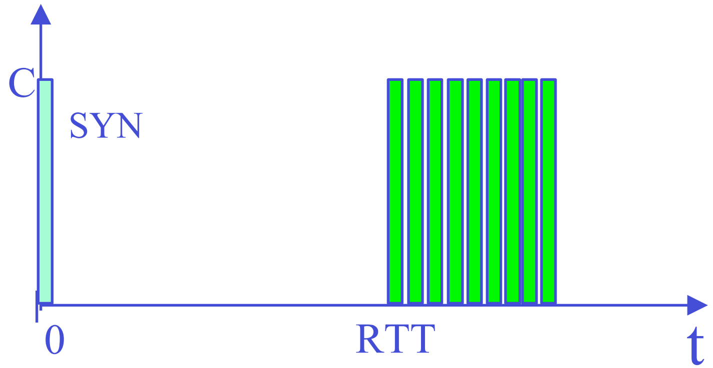
- Link capacities will continue to increase
- More flows could fit within the one-way "pipe"

Example: A Short Flow

Average TCP Flow is 10 packets

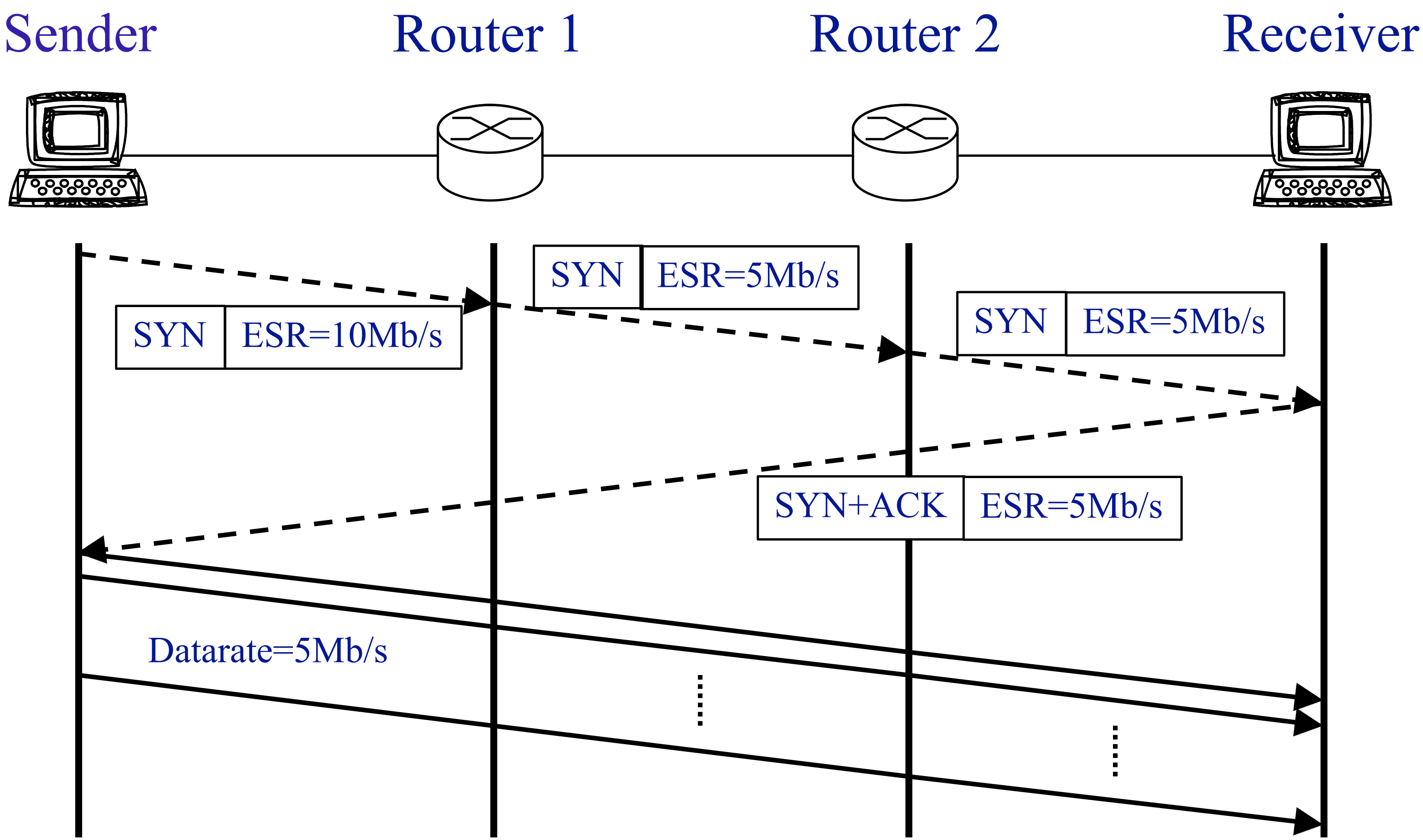


But the flow could finish much faster!



TCP forces the flow to last four times longer than necessary!

Explicit Starting Rate (ESR)



The Goal of Congestion Control

Metric for short flows: User response time

Goal: Minimize flow duration

Optimal for single link

- Shortest Remaining Processing Time (SRPT) scheduler
- FIFO and allot rates to emulate SRPT
- Optimal = complex discipline
- Optimal = need to know flow sizes and per-flow state

Allot equal rate to all flows

Minimize Flow Duration

Flow duration:

$$\tau = \frac{L}{R} + d_l(R)$$

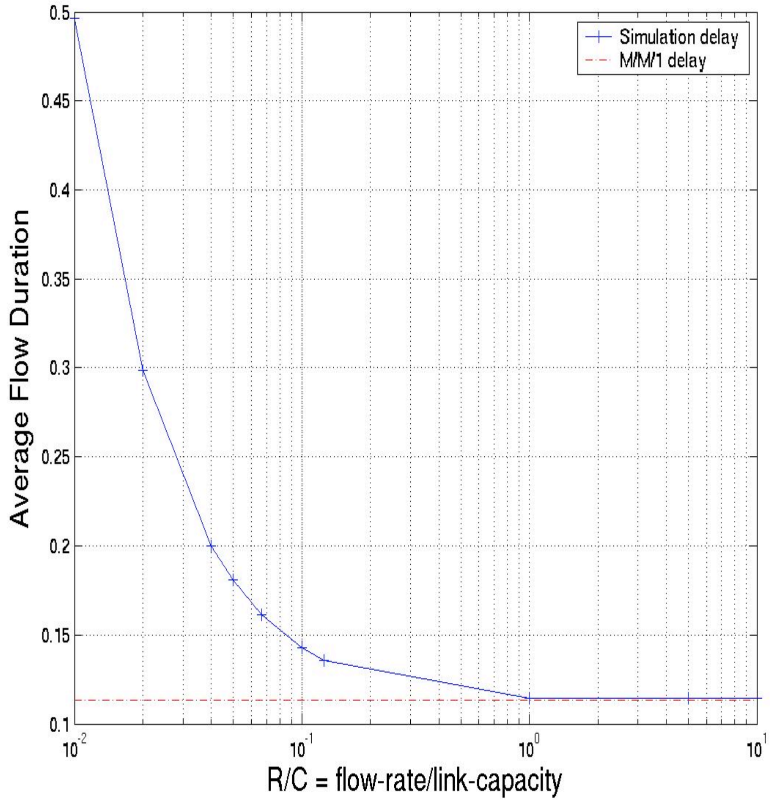
L = flow size; R = flow rate; $d_l(R)$=queueing delay of the last packet

Is there an optimal rate that minimizes the flow duration, E[t]?

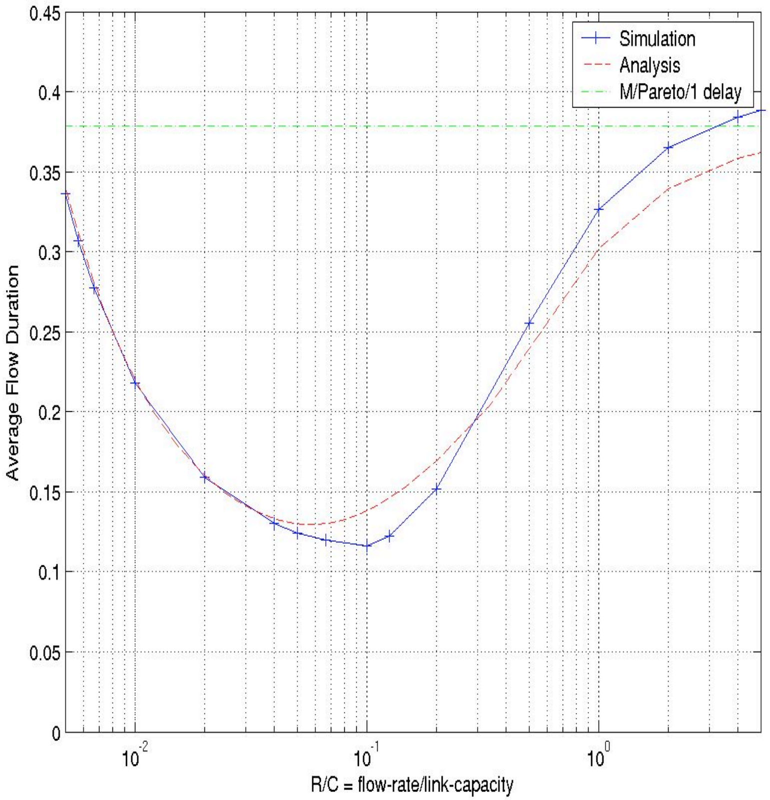
L/R $\downarrow$ as R $\uparrow$; d(R) $\uparrow$ as R $\uparrow$

Is there an Optimal Rate ?

Exponential

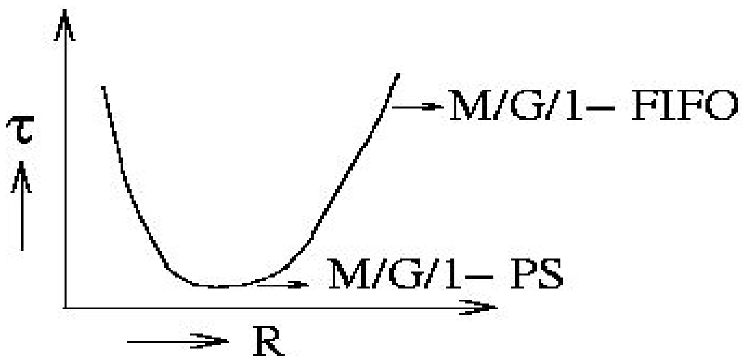


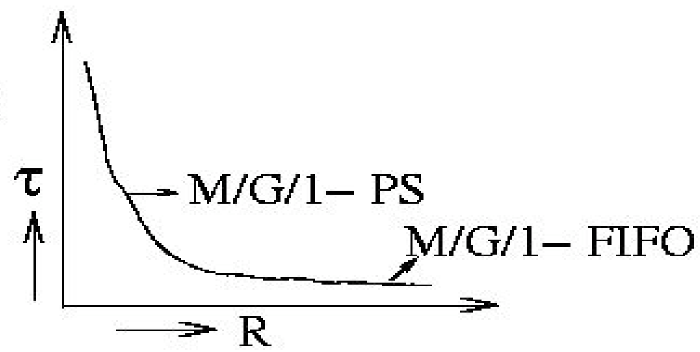
Pareto



Intuition for the 'U' curve

$$\frac{L}{R} + d_1(R)$$





M/G/1–PS < M/G/1–FIFO

$$\frac{EL / C}{1 - \rho} < \frac{\lambda\, EL^2}{2\, C^2 (1 - \rho)} + EL / C$$

$$(EL)^2 < Var(L)$$

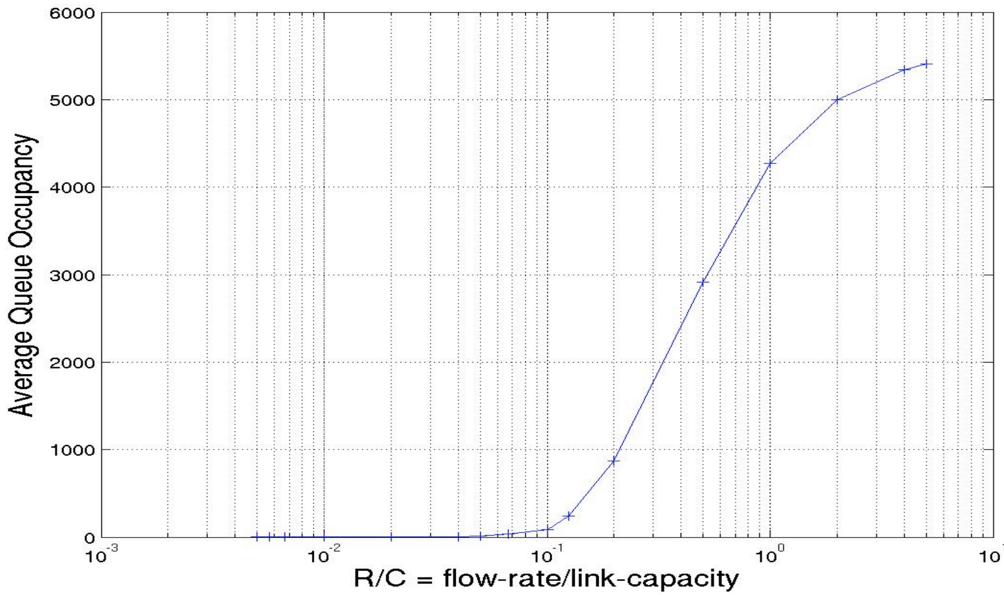
True for pareto, bimodal with appropriate 'p'

Not true for Uniform, Exponential, Deterministic

Picking a Rate

Solving 'U' curve = complex

Observation:



Minimizing queue occupancy AND Maximizing link utilization <-----> Minimizing flow duration

Rate Control Protocol

Router has a global rate 'R'

Updates rate 'R' once every RTT or shorter

Keeps a running average of RTT

New flows get rate through three-way handshake

Source continuosly sends 'rate update' packets

RCP Algorithm

$$R = C / N(t)$$

Problem: feedback delay, N(t) changes

$$R(t) = R(t-d) + \frac{\alpha (C - input_T) - \beta \frac{Q}{d}}{N}$$

N difficult to obtain, estimate as C/R

$$R(t) = R(t-d) \left(1 + \frac{\alpha (C - input_T) - \beta \frac{Q}{d}}{C} \right)$$

Control Interval: T < d

$$R(t) = R(t-d) \left(1 + \frac{\frac{T}{d} [\alpha (C - input_T) - \beta \frac{Q}{d}]}{C} \right)$$

RCP Algorithm

Using only queue length information

$$\text{if } q(t) > 0$$

$$\text{input}_T = \dot{q}(t) + C$$

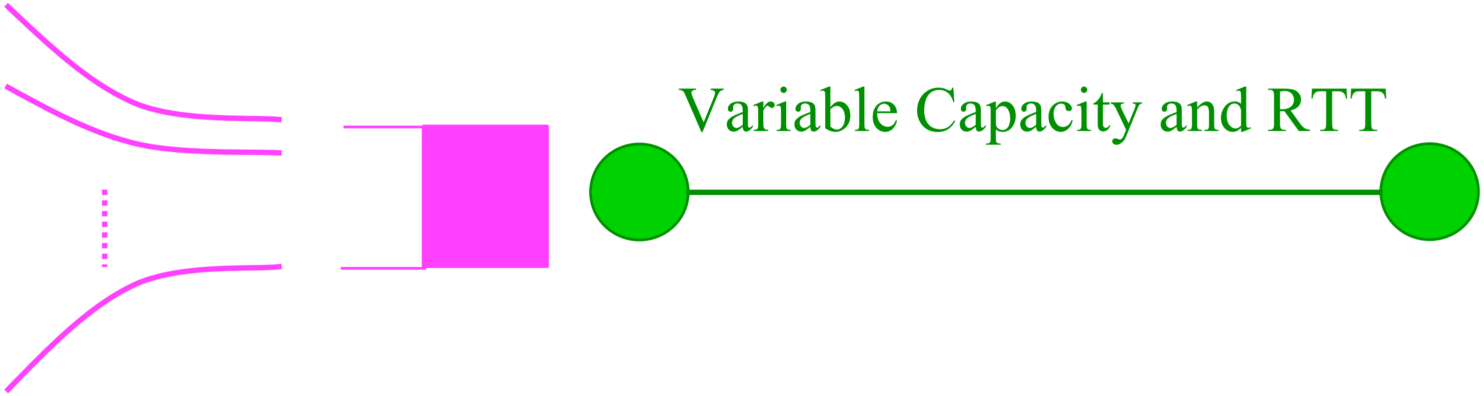
$$\text{else}$$

$$\text{input}_T = C - \eta C$$

Parameters = "run and see"

alpha = 0.4, beta = 0.4, eta = 0.05, T = 10ms

Performance of RCP

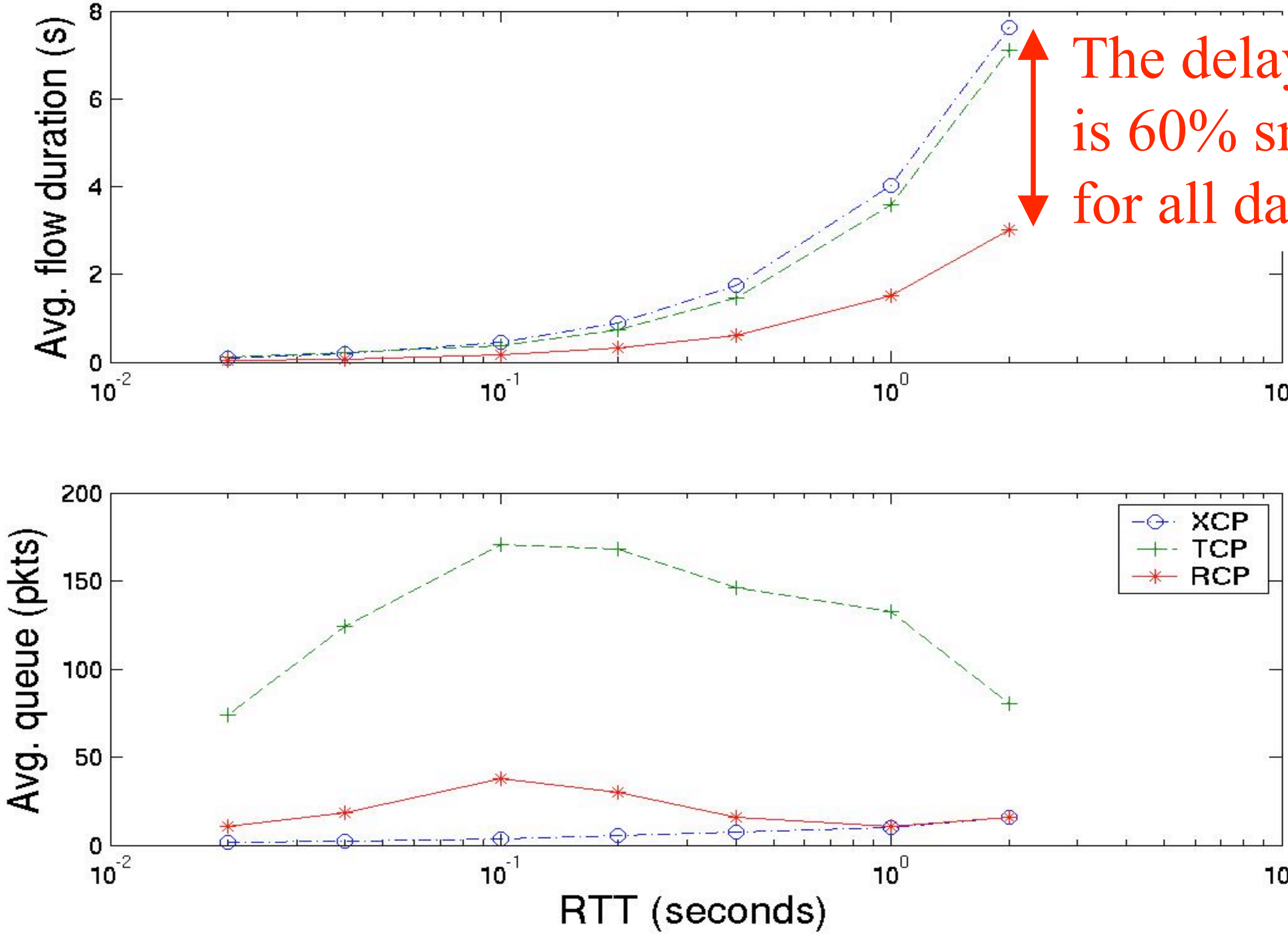


Flows arrive as a Poisson process, load=0.8

Flow size distribution: Pareto(1.2, 25)

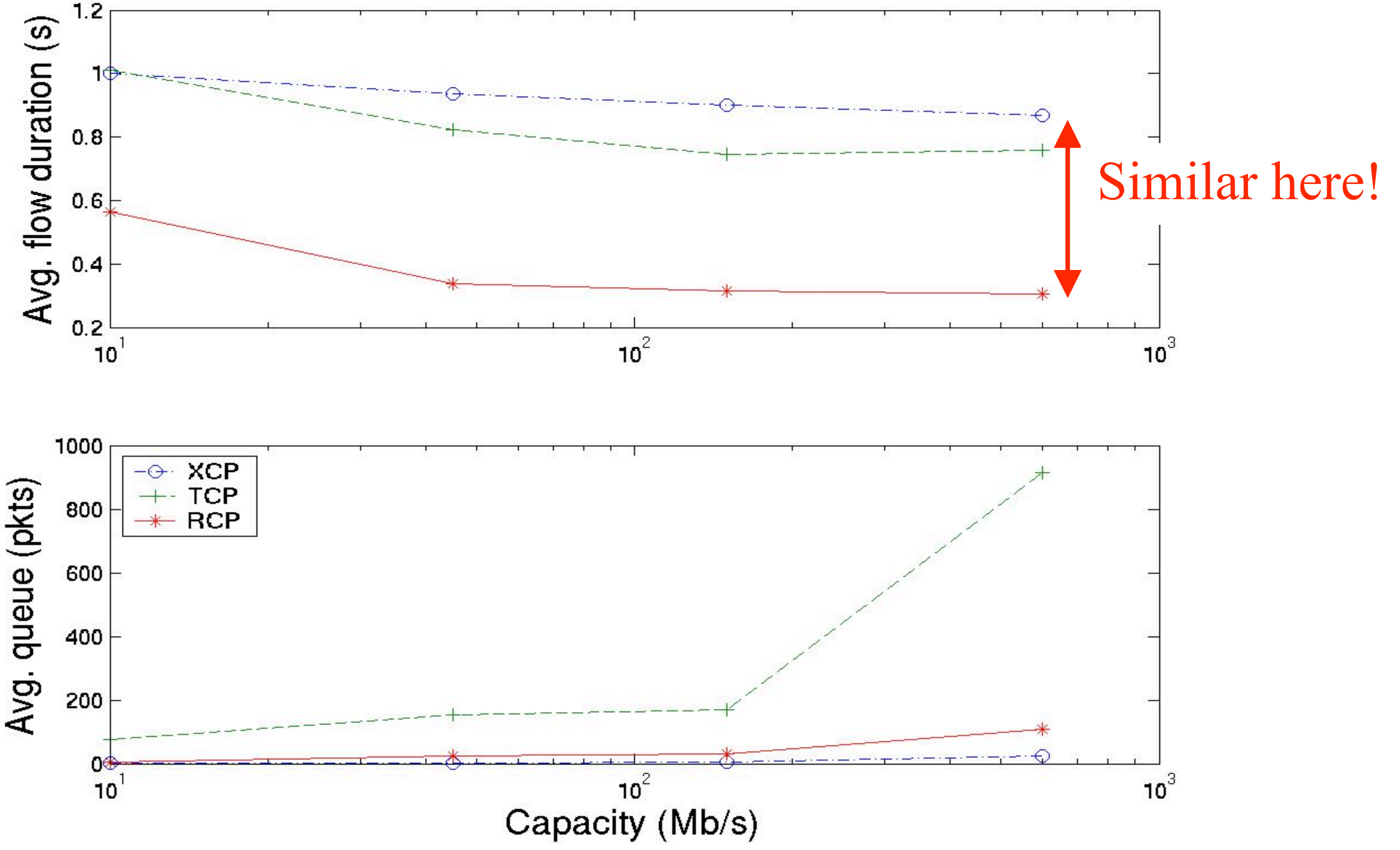
Buffer = 2*Capacity*RTT

Different Link RTTs



The delay of RCP is 60% smaller for all data points!

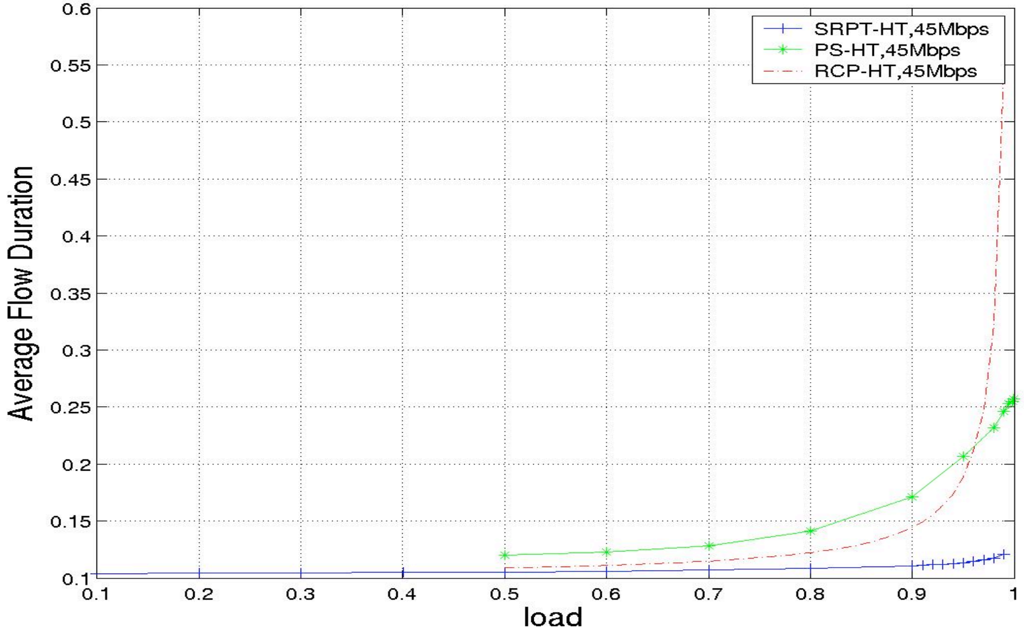
Different Link Capacities



Comparison with Optimal

Comparison under Heavy-tailed traffic

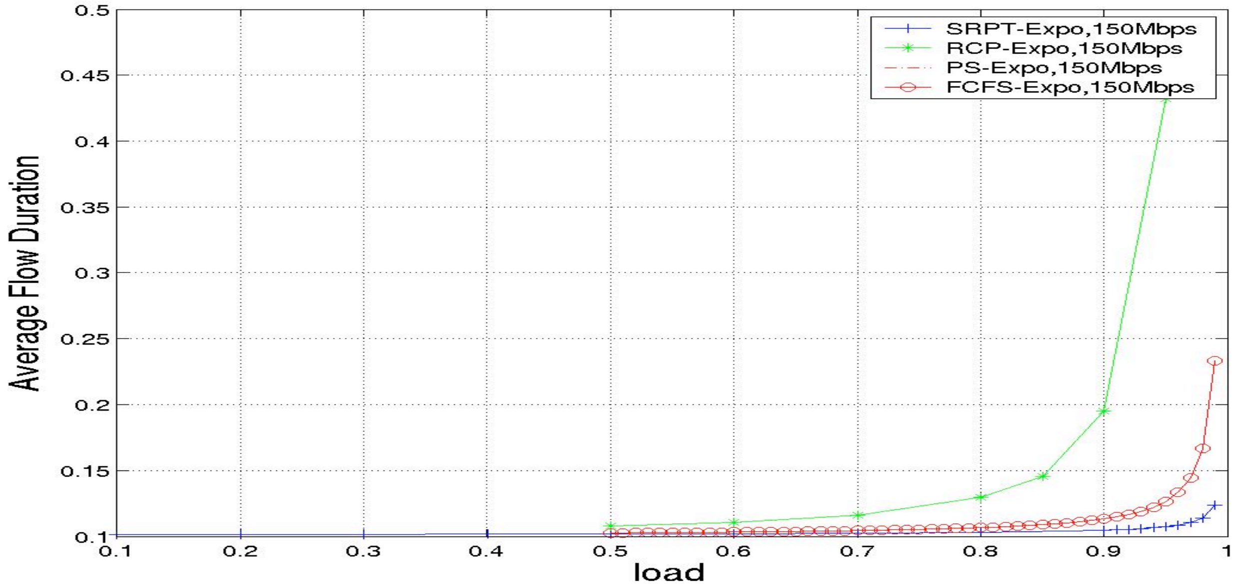
- FIFO delay
- PS delay
- SRPT delay



Comparison with Optimal

Comparison under Exponential traffic

- FIFO delay
- PS delay
- SRPT delay



Ongoing Work

RCP Analysis

- RCP designed intuitively
- Parameters = "test and see"
- Control theoretic analysis for stability, design

RCP under stress

- Rapid changes in load
- Congested reverse link
- Heterogenuous delays
- Network scenario: multiple bottleneck links