# TCP performance analysis through processor sharing modeling

Pasi Lassila[a,b], Hans van den Berg [a,c],

Michel Mandjes [a,d], and Rob Kooij [c]

[a] Faculty of Mathematical Sciences, University of Twente

[b] Networking Laboratory, Helsinki University of Technology

[c] KPN Research, the Netherlands

[d] CWI, the Netherlands

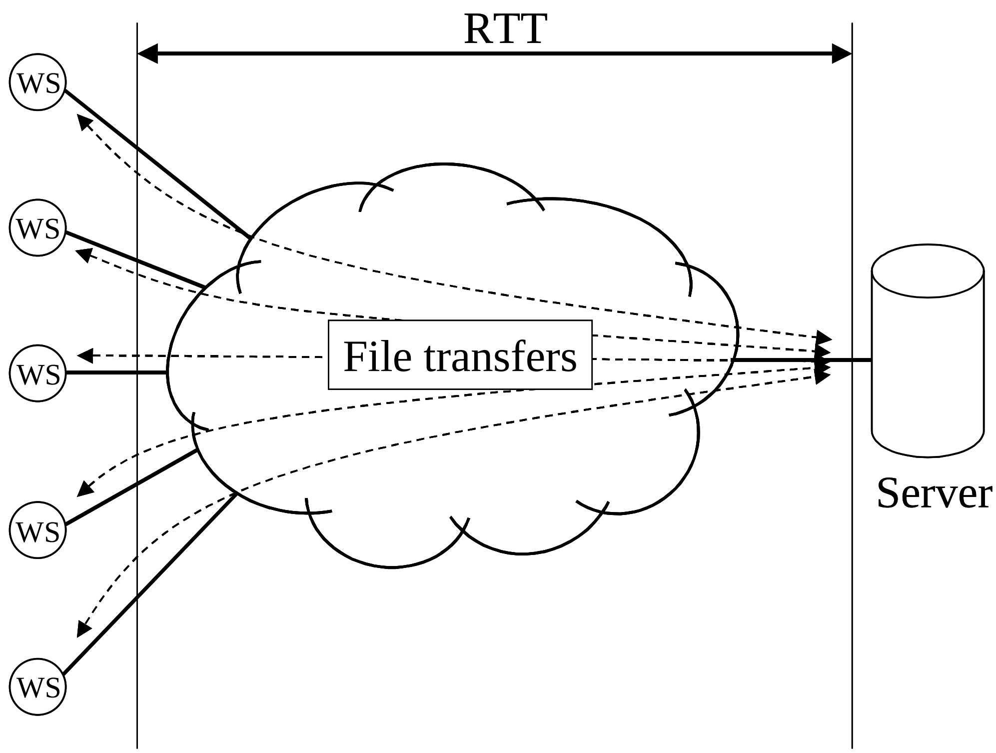
# Outline

- Motivation
- Existing models: flow level vs. packet level
- Our approach: combined flow-packet level model
- Numerical results
- Conclusions

# Motivation

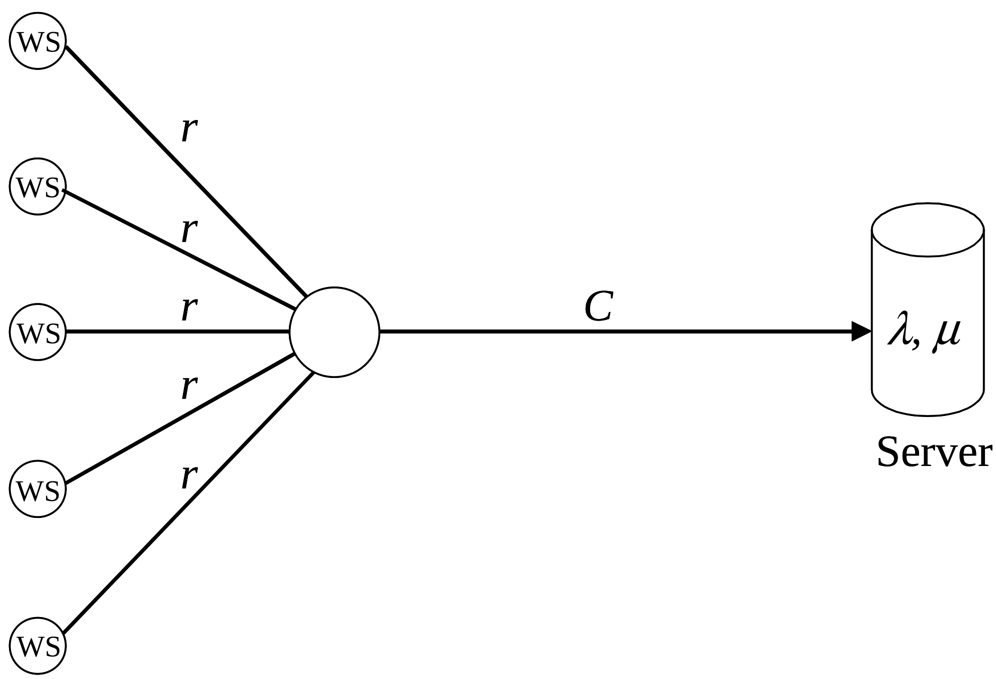
- Most Internet traffic carried by TCP
  - Elastic traffic: tolerates variations in throughput
  - Packet losses used as indications of congestion
  - If no packet losses, TCP increases its sending rate
  - For each packet loss, rate is (typically) halved
- Main performance measures: throughput and delay
- For practical purposes, simple yet accurate enough models are needed

# Scenario




- Requests arrive randomly and files have random lengths
- Issues: packet losses and RTT delays
- Bottleneck: access link, network, server link
  - Assume limitation is due to access or *one* bottleneck link
- What is the mean file transfer delay?

# Flow level models




- Generalized Processor Sharing, GPS (Cohen,1975)
  - Poisson file requests at rate$\lambda$
  - File lengths i.i.d. with mean $1/\mu$ (insensitivity)
  - $r_n$ = joint sending rate given $n$ flows
  - Each flow gets $r_n/n$

# GPS steady state distribution

- First define

$$\phi(n) := \begin{cases} \lambda \cdot (\mu r_n)^{-1} & \text{for } n \in \mathbb{N} \\ 1 & \text{for } n = 0 \end{cases} \quad , \quad \psi(n) := \prod_{i=0}^{n} \phi(i)$$

- Then $\mathbb{P}(N = n)$ equals

$$\mathbb{P}(N = n) = \frac{\psi(n)}{\sum_{m=0}^{\infty} \psi(m)}$$

- Observations

  – Letting $C \to \infty$, we obtain the infinite server Erlang system

  – Choosing $r_n = C$ we obtain the traditional PS-system with geometric distribution (each flow gets its fair share $C/n$)

  – Choosing $r_n = \min(rn, C)$ models case where sources have max rate $r$. Poisson-type left tail and geometric right tail.

- Mean delay (Little): $\mathrm{E}[D] = \mathrm{E}[N]/\lambda$

# GPS properties

- Features:
  - Insensitivity to file size distribution
  - Conditional mean delay linear in file size
- Idealizations:
  - Assumes instantaneous rate adaptation (new flow gets its fair share immediately)
  - Does not take into account packet losses (assumes infinite buffers)
  - Does not take into account RTT delays
  - Gives too optimistic results

# TCP modeling

- Assumption $n$ persistent flows
- The "square-root"-formula for TCP throughput (single flow)

$$t \approx \min\left\{r, \frac{\Gamma}{\mathrm{RTT}\sqrt{p}}\right\}$$

- Iterative approach to determine $t_n$ and $p$
  - Given $n$ flows, $t_n$ is the total arrival rate from these
  - Assume that at packet level arrivals are Poisson. Packets enter an M/D/1/$K$ queue, where they observe a loss rate $p(t_n) \Rightarrow$ fixed point

$$t_n = \min\left\{nr, \frac{n\Gamma}{\mathrm{RTT}}\frac{1}{\sqrt{p(t_n)}}\right\},$$

- Features: captures losses and RTT delays, but no flow level dynamics

# Combined flow-packet level model (1)

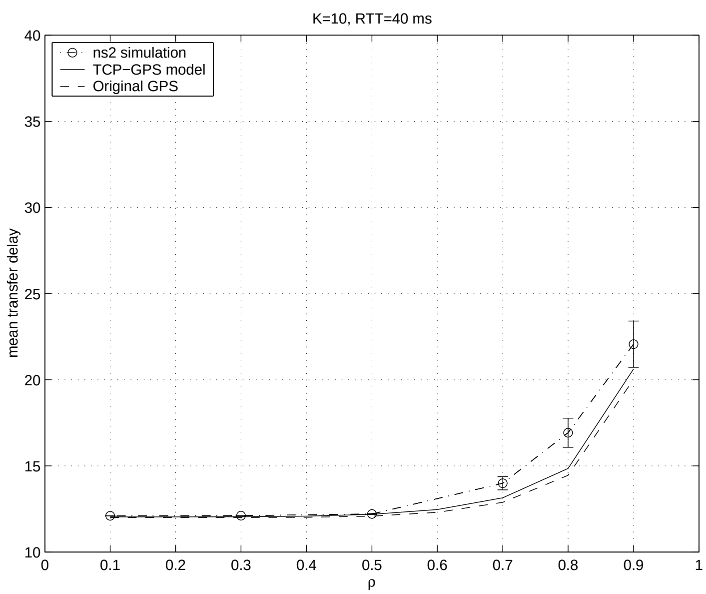
- Idea: Couple previous two models together
- Procedure:
  - Using the TCP equation, we can determine conditional sending rates $t_n$, given $n$ flows
  - The goodput at the packet level equals $t_n(1 - p(t_n))$
  - On the flow level, the system is assumed to behave as a GPS system with rates $r_n = t_n(1 - p(t_n))$
- Other improvements (hacks?)
  - Effect of queuing delay: replace RTT by RTT $+ \bar{q}(t_n)$
  - Initial slow start effect: compute the number of unsent packets due to slow start and compensate mean delay with the time to send them at average rate

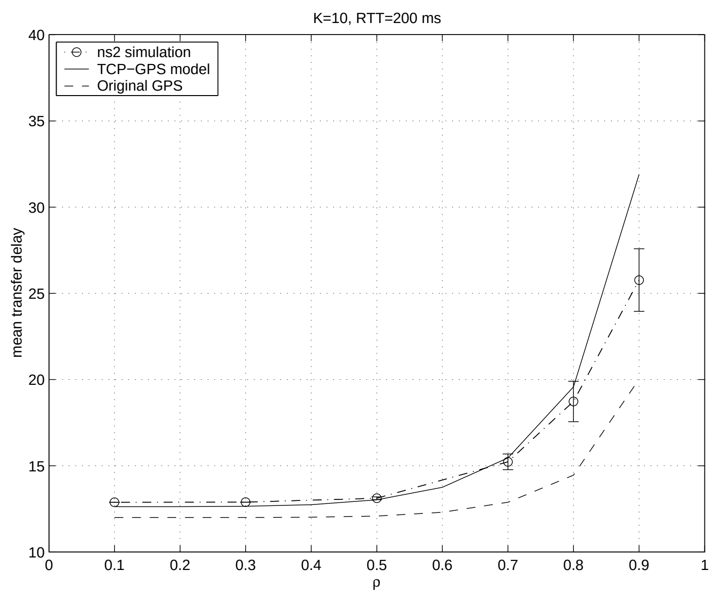
# Combined flow-packet level model (2)

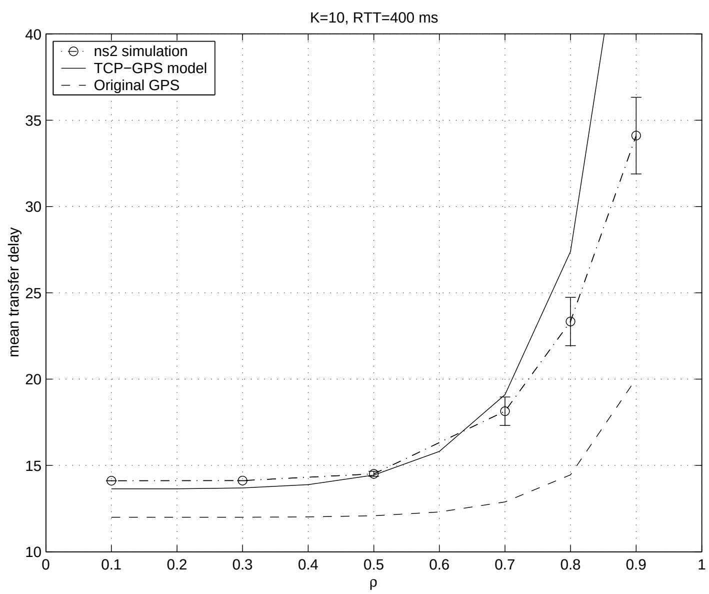
- Applicability? (method completely heuristic)
  - TCP throughput equations are approximate and generally assume low loss rates ($< 10\%$)
  - Time scale decomposition: new flow obtains its fair share quickly (compared to the mean file transfer time)
  - Effect of RTT only seen when RTT relatively large
  - Poisson packet arrival assumption probably never valid, but how bad is it?
- Simulations
  - Done by using ns2 (2.1b8a)
  - $C = 10$ Mbps, flength 1000 pkts (constant), psize 1500B
  - RTT=$\{40, 200, 400\}$ ms, $K = \{10, 50\}$, $r = \{1, 2\}$ Mbps

# Numerical results (mean delay, small buffer)

$r = 1$ Mbps, $K = 10$, RTT $= \{40, 200, 400\}$ ms

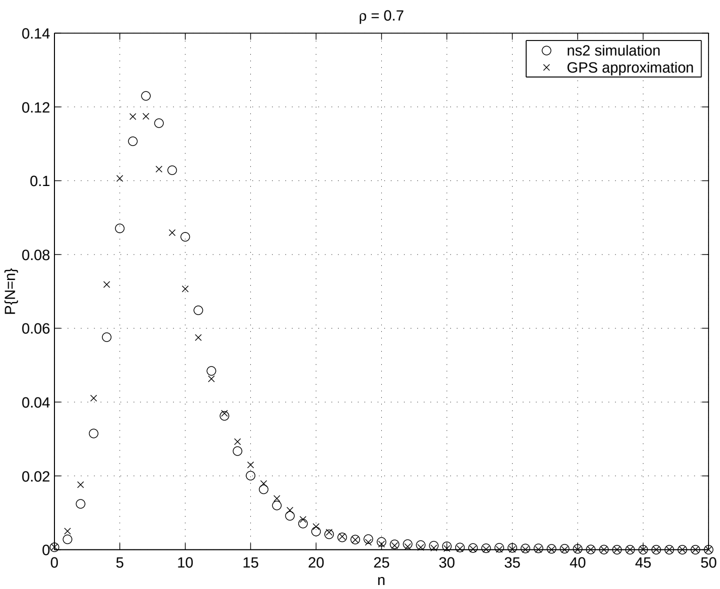


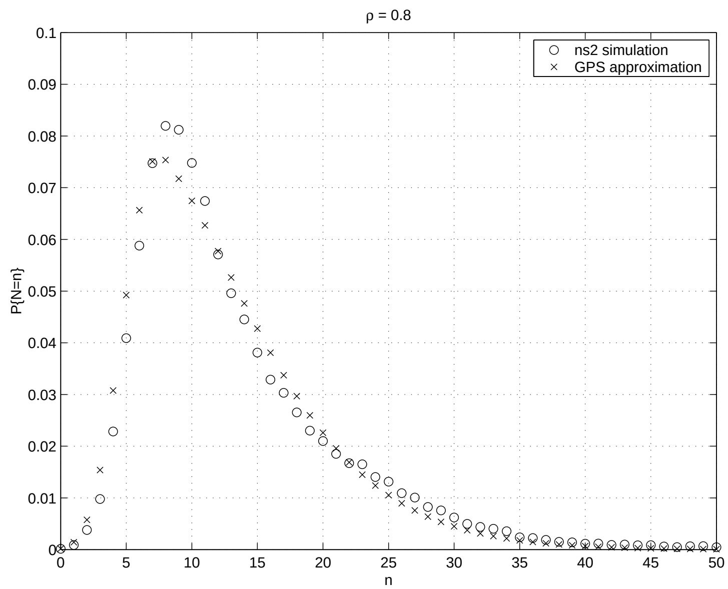



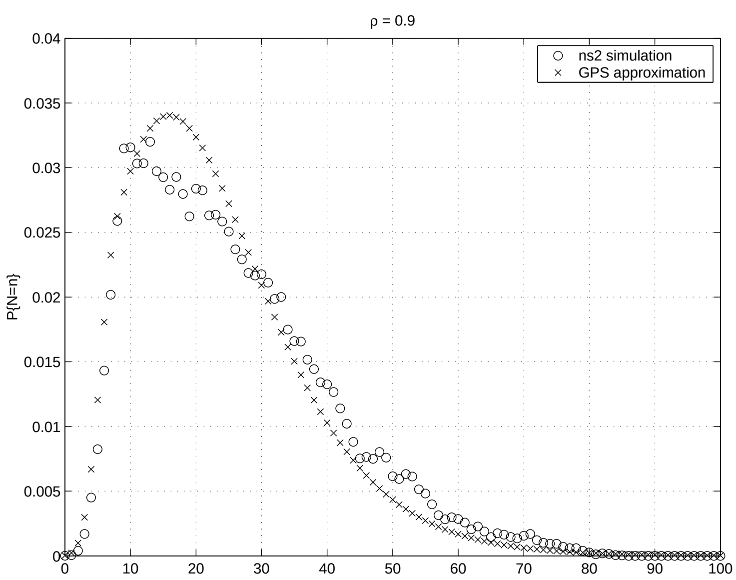



# Numerical results (distribution, small buffer)

$r = 1$ Mbps, $K = 10$, RTT $= 200$ ms, $\rho = \{0.7, 0.8, 0.9\}$

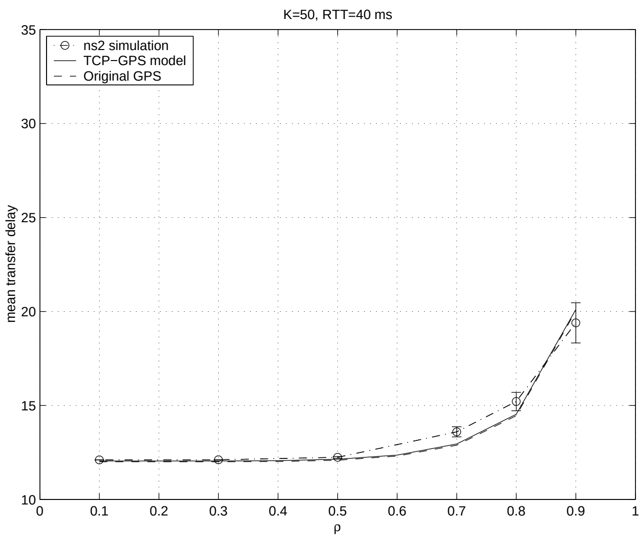


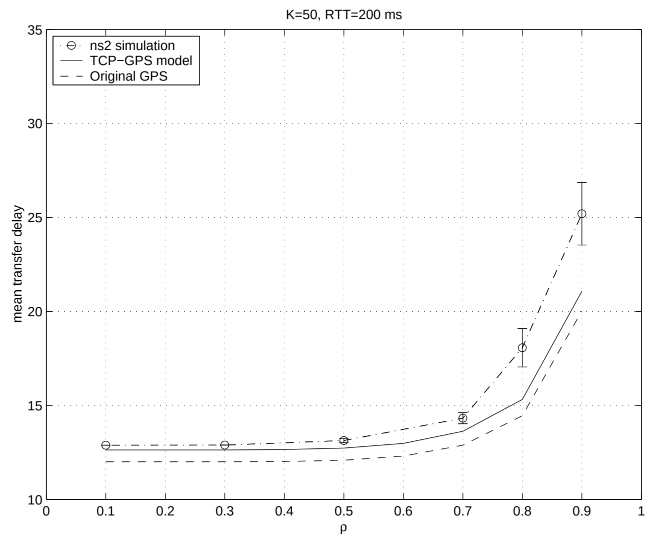



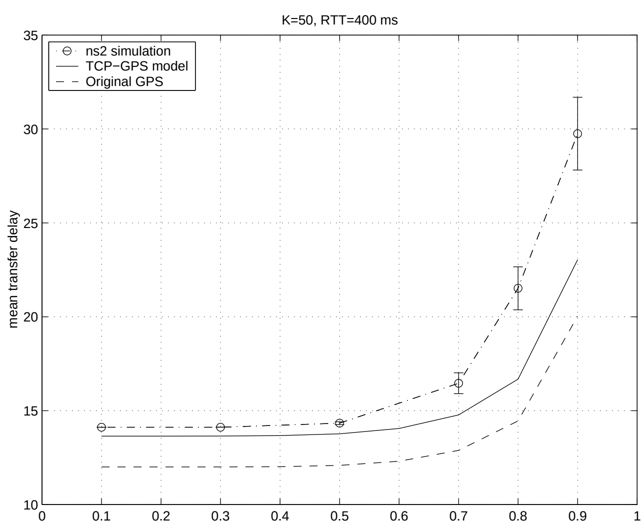



# Numerical results (mean delay, big buffer)

Buffer size $K = 50$, RTT $= \{40, 200, 400\}$ ms

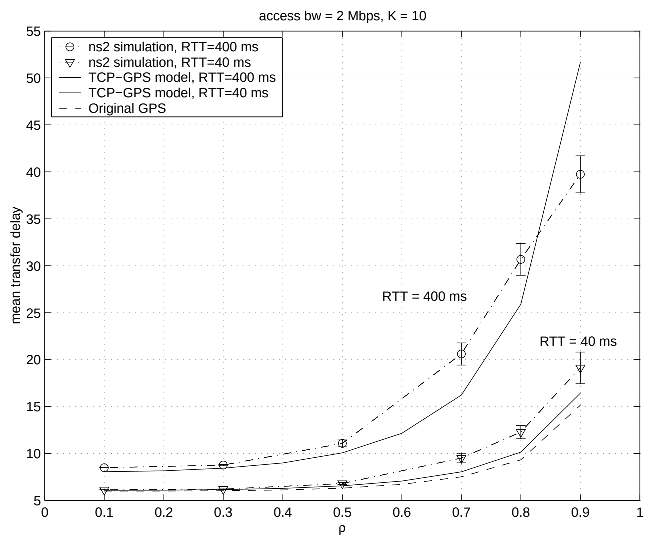


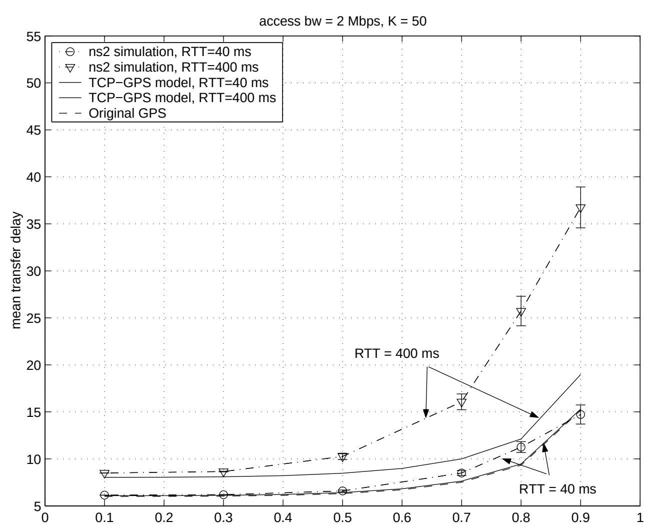



# Numerical results (larger access rate)

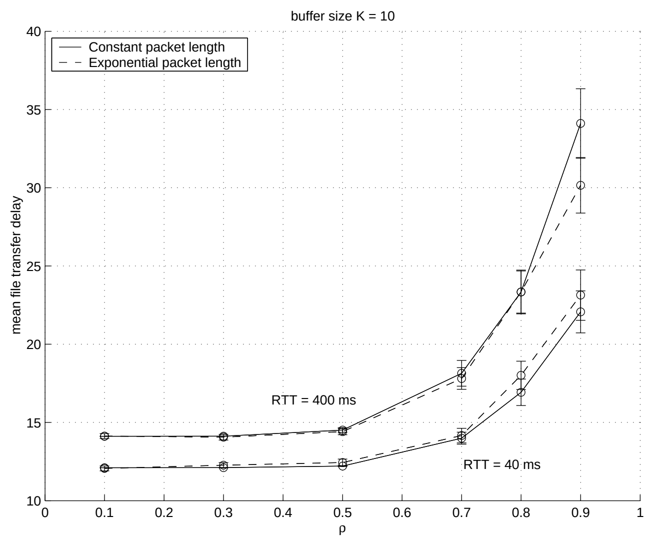
$r = 2$ Mbps, $K = \{10, 50\}$, RTT $= \{40, 400\}$ ms

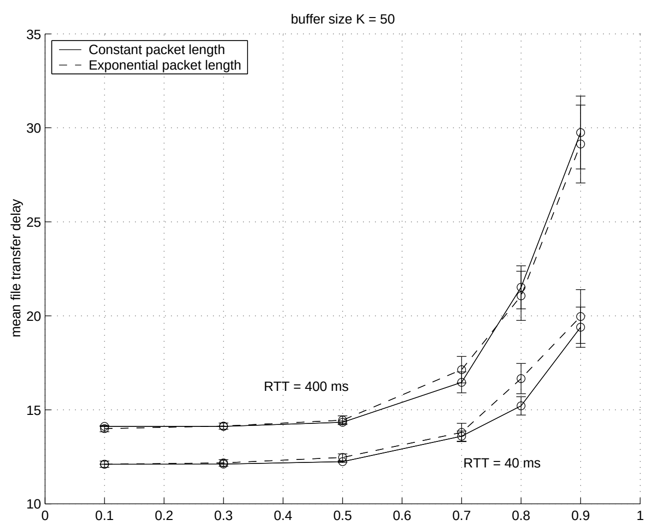


# Numerical results (insensitivity)

$r = 1$ Mbps, $K = \{10, 50\}$, RTT $= \{40, 400\}$ ms



# Conclusions and future work

- An extension of the traditional GPS model
  - \+ Captures qualitatively the effect of RTT and finite buffers on delay
  - \- Quantitatively, the parameters can be chosen to give good/bad correspondence with simulations
- Future work
  - – Generalization to networks of GPS queues (multiple congested links)
  - – Poisson assumption does not really work at packet level, a better packet level model is needed