Differentiation Between Short and Long TCP Flows

Konstantin Avrachenkov, Urtzi Ayesta, Patrick Brown and Eeva Nyberg

diffusion libre

D1 - 09/04/2002

Introduction

- Mice and elephants: 80% of the flows are short, 5% of largest flows make up for 95% of the load.
- TCP point of view, short connections are more vulnerable to losses.
- Queuing theory point of view: Average number of users can be reduced with an appropriate scheduling.
- Contribution of this work is two fold:
  - Mathematical model to evaluate the performance of large connections in size based scheduling schemes.
  - Stateless threshold based approach to implement the size based differentiation.

Outline of the Talk

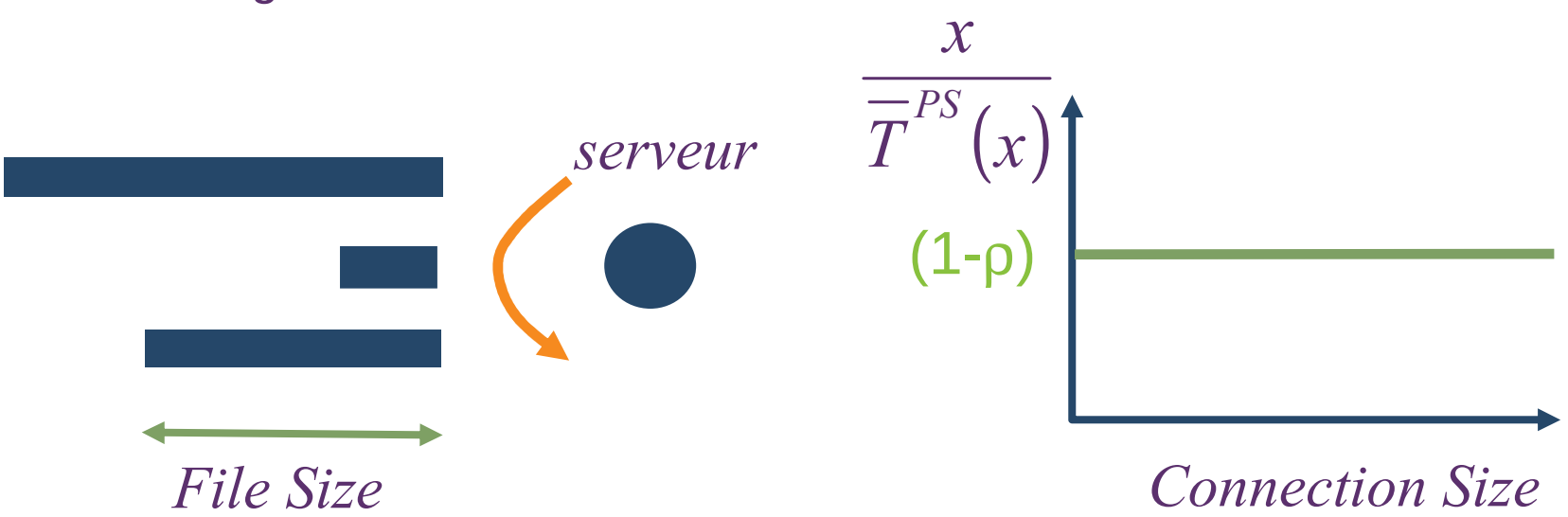
- Introduction.
- Scheduling review.
- Mathematical analysis.
- TCP implementation.
- Simulation results.
- Conclusion.

Scheduling review (I)

- Two important set of disciplines depending on whether or not the size of jobs is known.
- The size is known: Shortest-Remaining-Processing-Time SRPT is optimal with respect to the average response time of the system.
- The size is not known, but we know the *attained service* of jobs. The most appropriate scheduling discipline depends on the service time distribution characteristics.

Scheduling review (II)

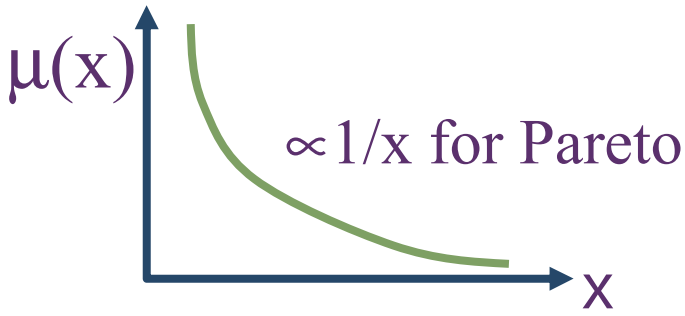
- Processor-Sharing (PS): All present jobs in the system get a fair share of service. If there are N connections, each one gets served at rate 1/N.
- An acceptable model for the current Best Effort TCP/IP network at high load.



Scheduling review (III)

- Hazard rate of a distribution function. $\mu(x)=P[x< \text{size of the job} \leq x+dx \mid \text{size of the job} > x]$

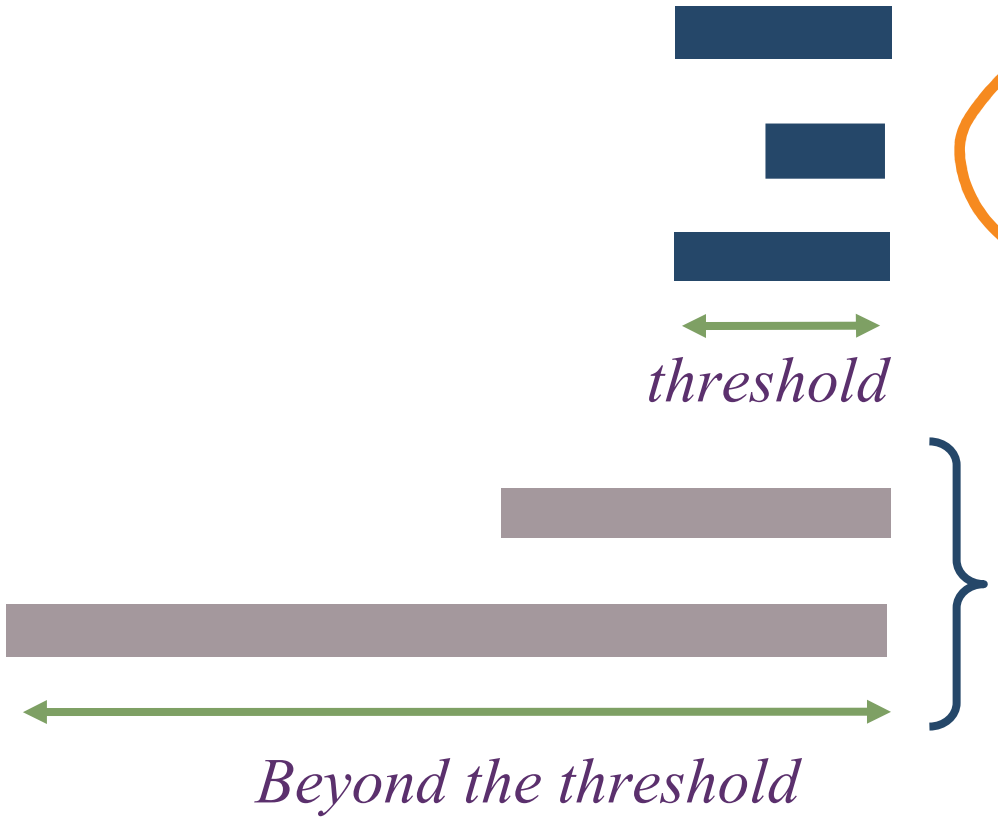
$$\mu(x)=\frac{f(x)}{1-F(x)}$$



- Least-Attained-Service (LAS): The job(s) who has attained the least amount of service is served. LAS is optimal with respect to the average response time in the system when the hazard rate is decreasing.

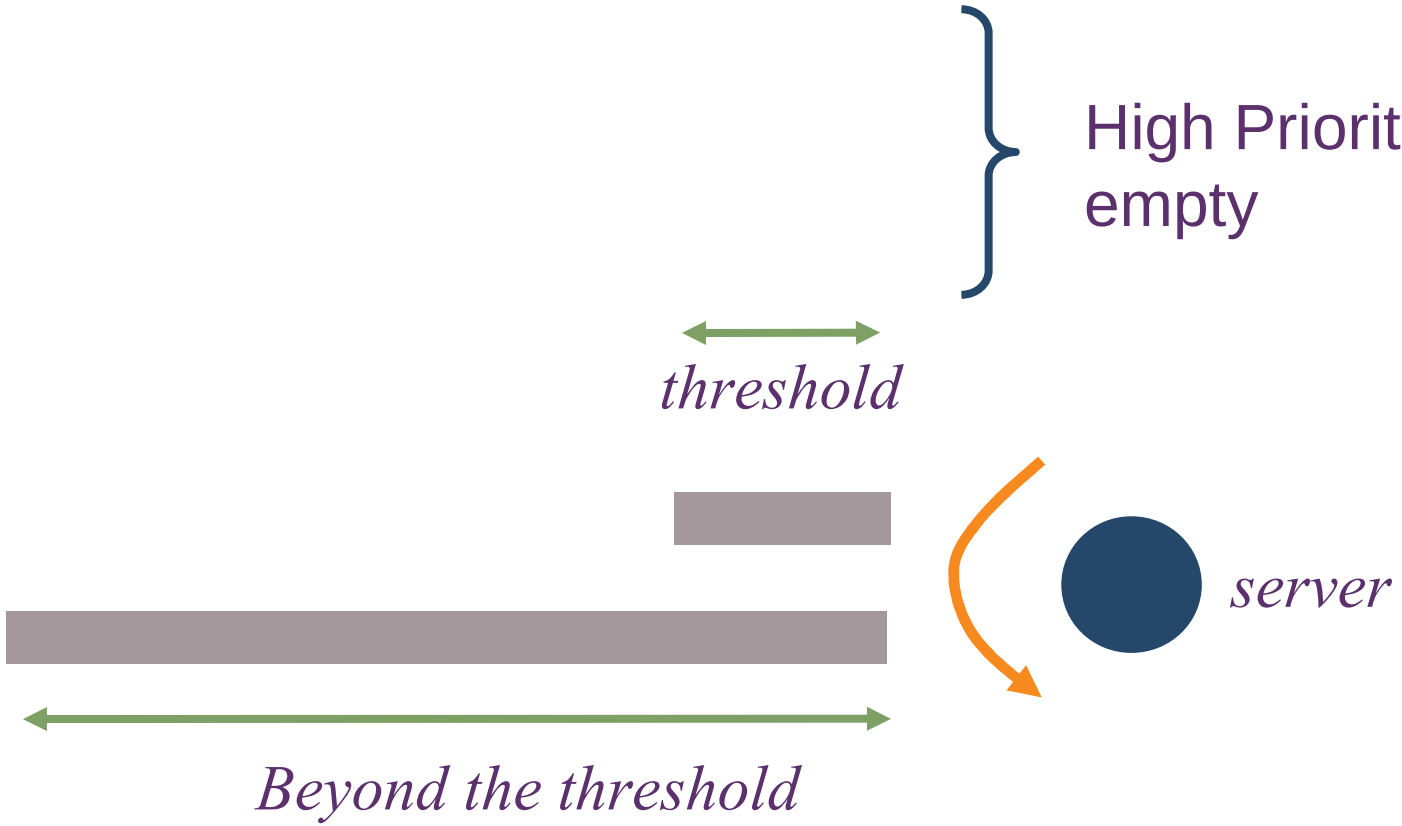
- PS+PS model. A particular case of the Multilevel-Processor-Sharing systems introduced by Kleinrock.

Processor Sharing+ Processor Sharing PS+PS (I)

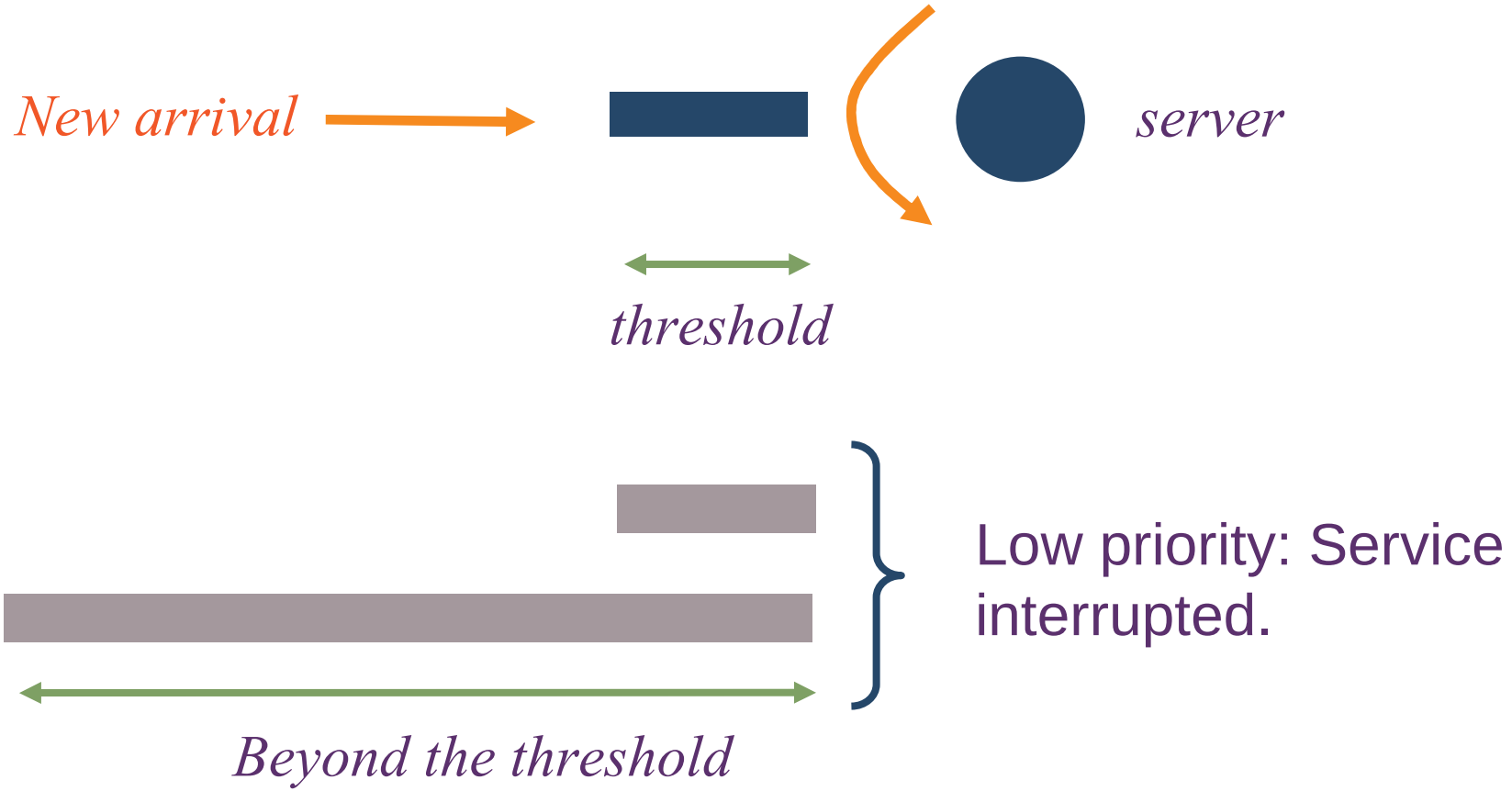


Low Priority: Connections that have already obtained threshold units of service.

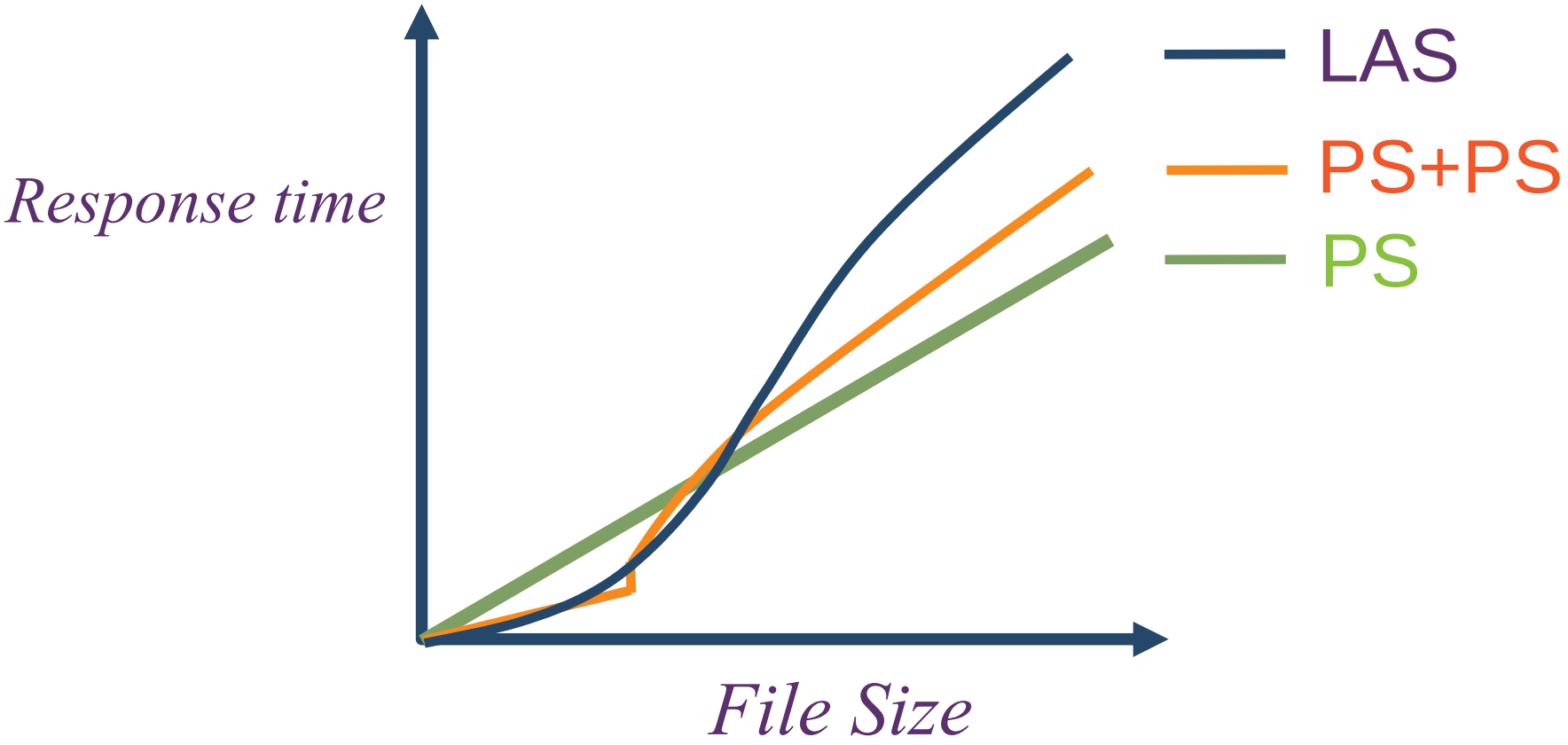
Processor Sharing+ Processor Sharing PS+PS (I)



Processor Sharing+ Processor Sharing PS+PS (I)

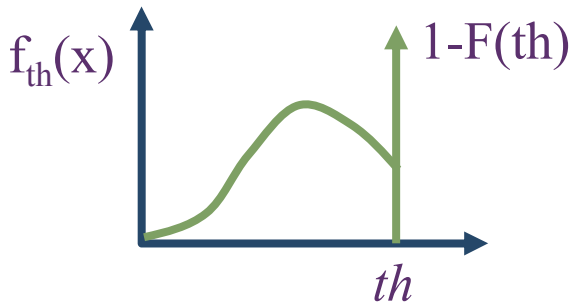


Comparison between PS et PS+PS and LAS

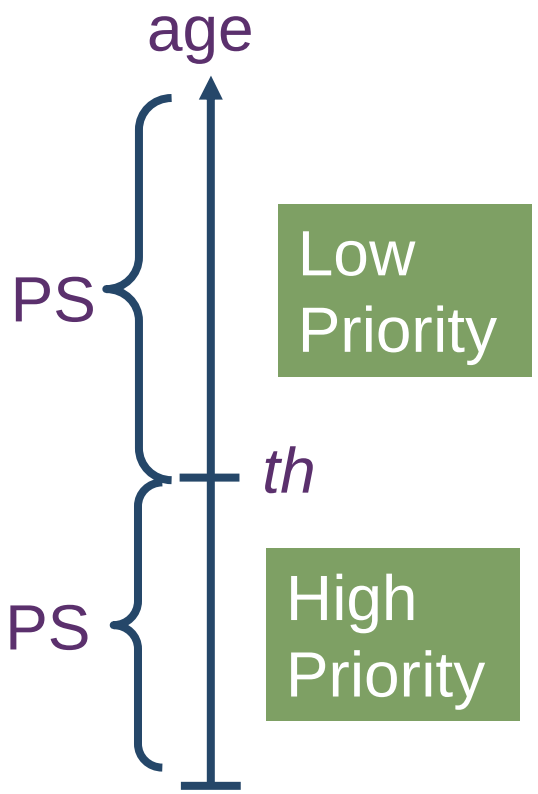


Mathematical Analysis of PS+PS

- The high priority system is processor sharing, with truncated random variable $X_{th}=\min\{X,th\}$.



- The low priority: The expected response time conditioned on the job size can be expressed as a function of the expected response time in a PS queue with Batch arrivals.



Expected response time of PS+PS

$$\overline{T}^{PS+PS}(x)=\begin{cases}\dfrac{x}{1-\rho_{th}} & if \quad x<th\\ f(th)+g(th)\overline{T}^{BPS}(x-th) & if \quad x\geq th\end{cases}$$

- Batch Processor-Sharing: Explicit expression for exponential file size distribution (Kleinrock et al.75, Rege and Sengupta 93).
- For a general distribution (Kleinrock et al. 1971)

$$\frac{\overline{T}^{BPS}(x)}{dx}=\lambda_B\int_0^{\infty}\frac{\overline{T}^{BPS}(y)}{dy}\overline{F}(x+y)dy+\lambda_B\int_0^{x}\frac{\overline{T}^{BPS}(y)}{dy}\overline{F}(x-y)dy+b\overline{F}(x)+1$$

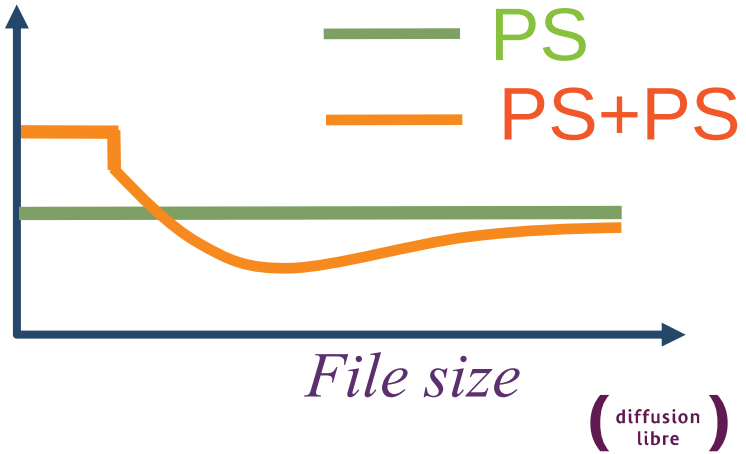
Asymptotic results

➔ PS+PS has an asymptote with slope 1/(1-ρ) and bias

$$\lim_{x\to\infty}\left(\overline{T}^{PS+PS}(x)-\frac{x}{1-\rho}\right)=const(th,E[X])$$

➔ The Slowdown for large file sizes is equal for the PS and PS+PS systems.

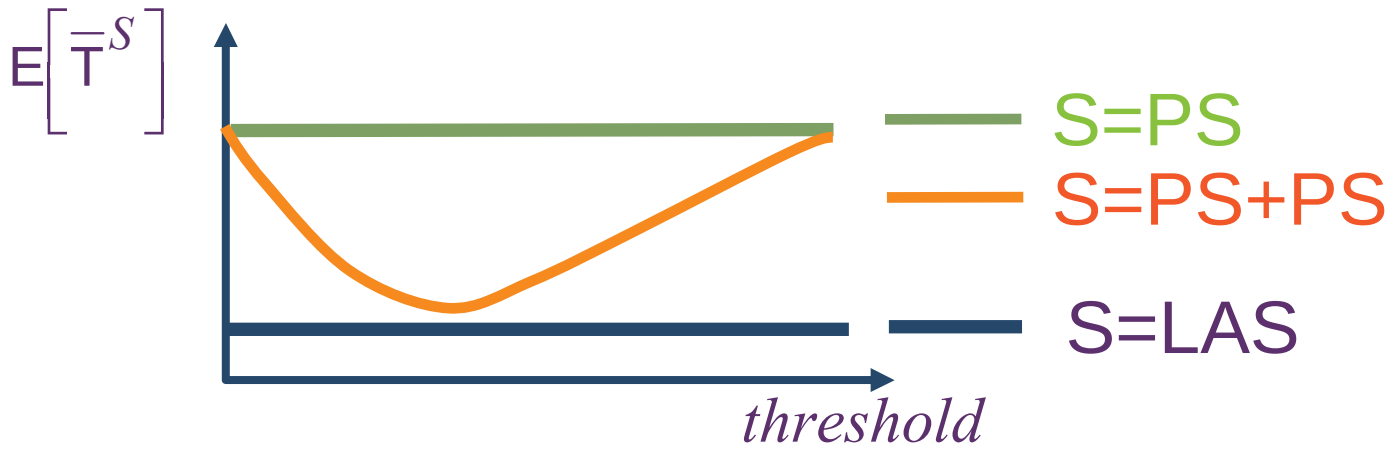
$$\lim_{x\to\infty}\frac{\overline{T}^{PS+PS}(x)}{x}=\frac{\overline{T}^{PS}(x)}{x}=\frac{1}{1-\rho}$$



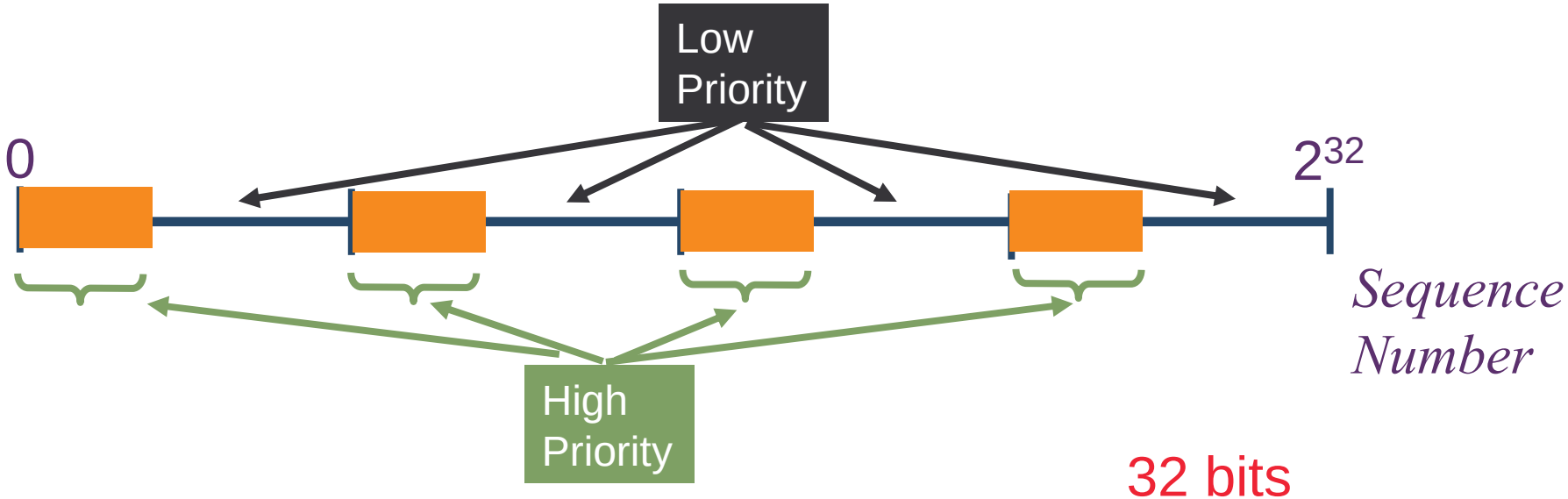
Expected unconditional response time reduced

- If the hazard rate of the distribution function is decreasing:

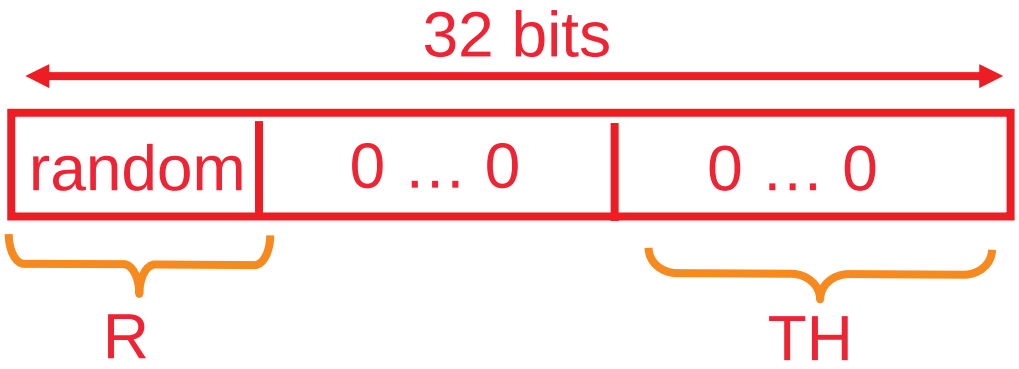
$$E\left[\overline{T}^{PS+PS}\right] \leq E\left[\overline{T}^{PS}\right]$$



RuN2c: Implementation of PS+PS



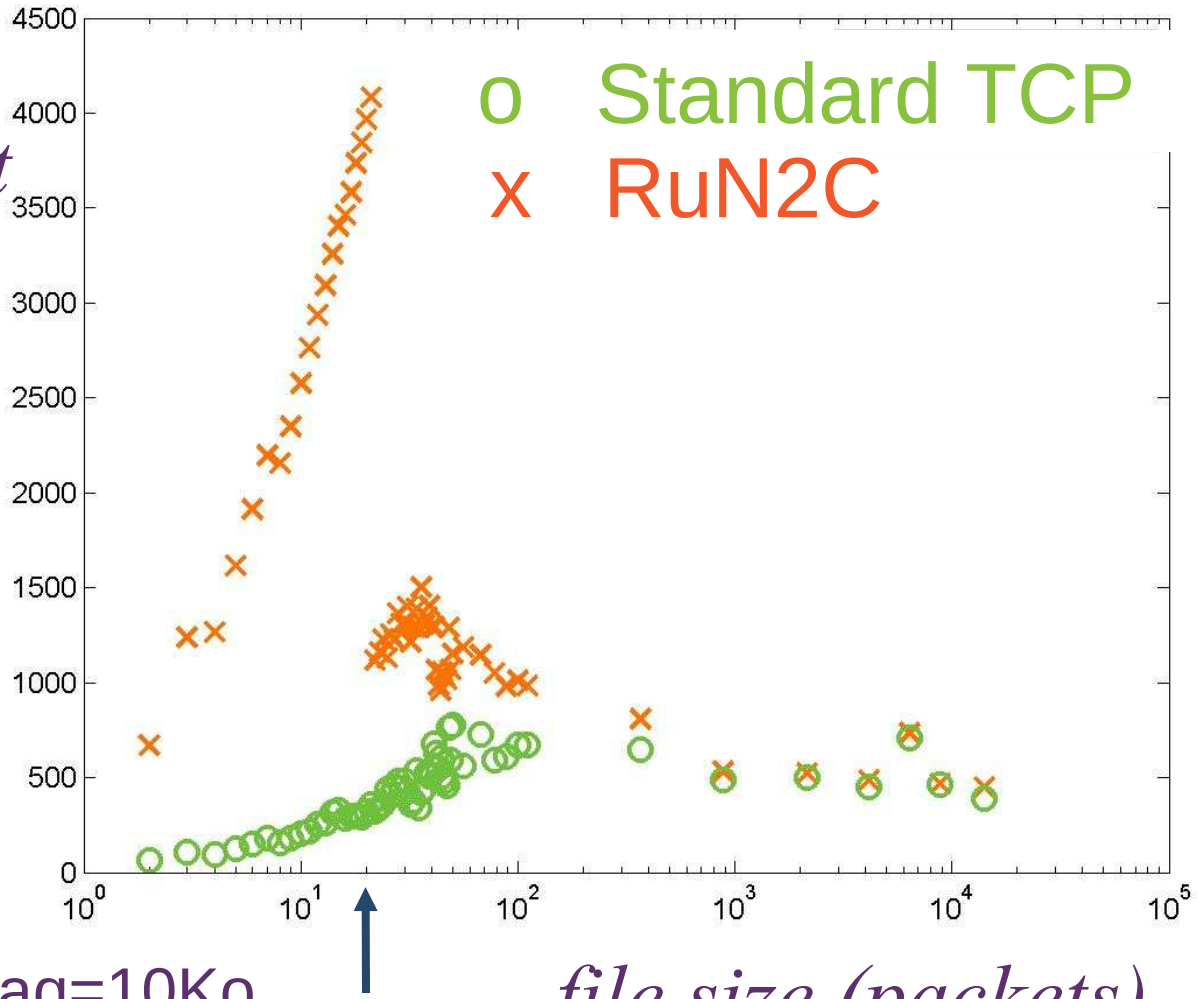
- 2^R initial sequence numbers.
- 2^{TH} bytes in high priority.



Priority detection by Mask comparison.

Simulations results

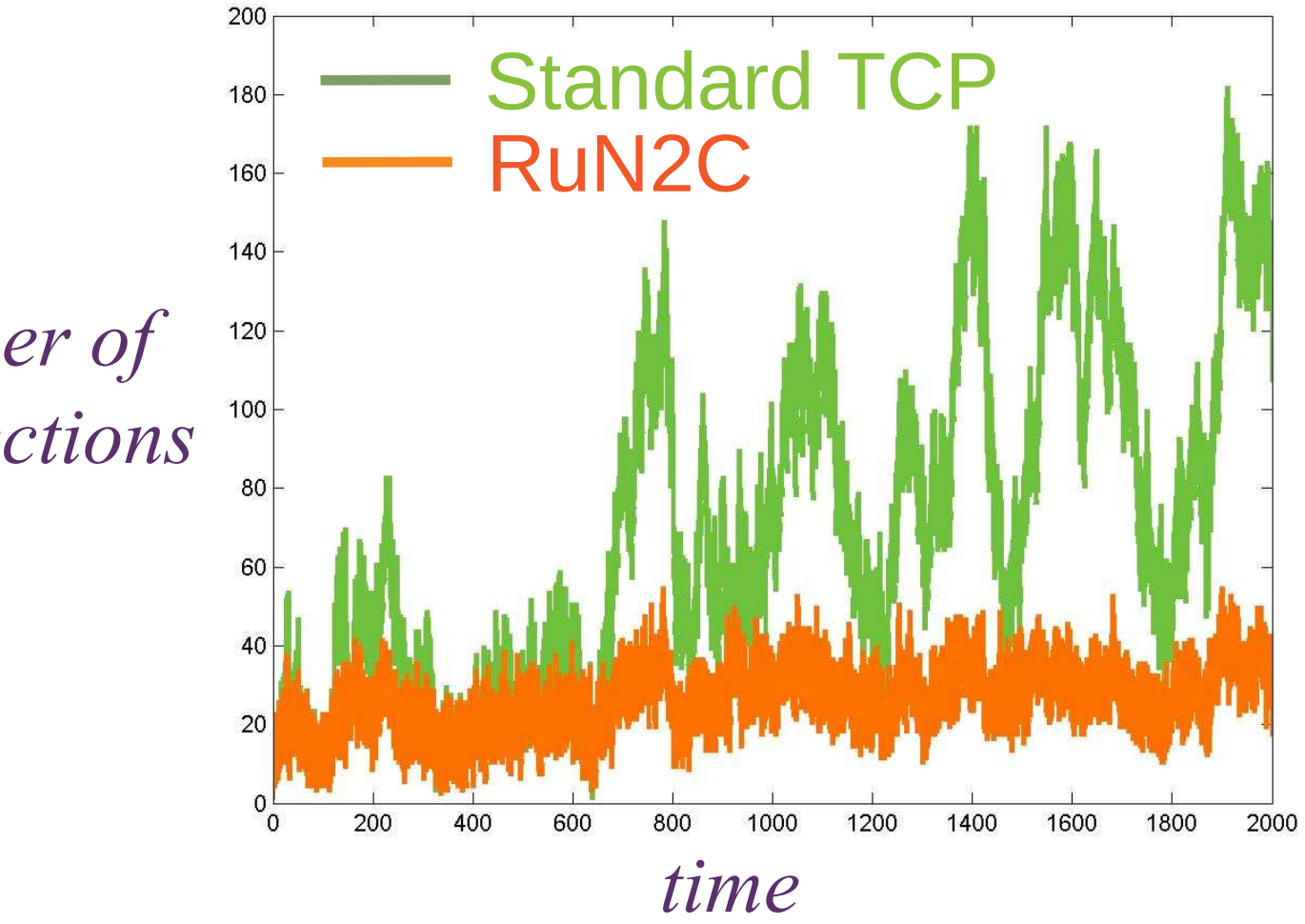
throughput



th=20paq=10Ko

file size (packets)

Number of connections



Conclusions

- We can provide better service to many short connections and do not deteriorate much the performance of long jobs.
- Giving preference to short flows is beneficial from the system point of view. Number of average connections is reduced.
- Stateless threshold based TCP proposal to perform the service differentiation.
- Future work:
  - Compatibility between RuN2C compliant and TCP standard connections.