

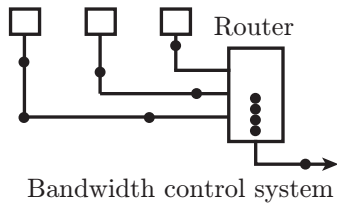
# A preliminary approach to feedback control of served-based real-time systems

Manel Velasco and Pau Martí

Automatic Control Department, Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain  
[manel.velasco, pau.marti]@upc.edu  
http://dcs.upc.edu

## Motivation

- Congestion control of TCP is an old problem with outstanding solutions
- Can we transfer these results to the real-time arena, for example to control CPU utilization?
- Our guess is YES



- A standard TCP congestion policy is as follows:
- Router senses outgoing bandwidth
  - Sender increments emitting rate if slack is available
  - Sender decrements emitting rate if overload is detected

This sounds like adjusting the number of jobs to be executed on a CPU.

Can we control it????

## Real-time formulation

Preliminary model borrowed from networking:

$$\begin{aligned} \dot{x}_1 &= N \frac{x_2}{W} - u_{max} \\ \dot{x}_2 &= \frac{a - \left(a + \frac{2b}{2-b}x_2\right) x_2 p}{W} \\ y &= x_1 - u_d \end{aligned}$$

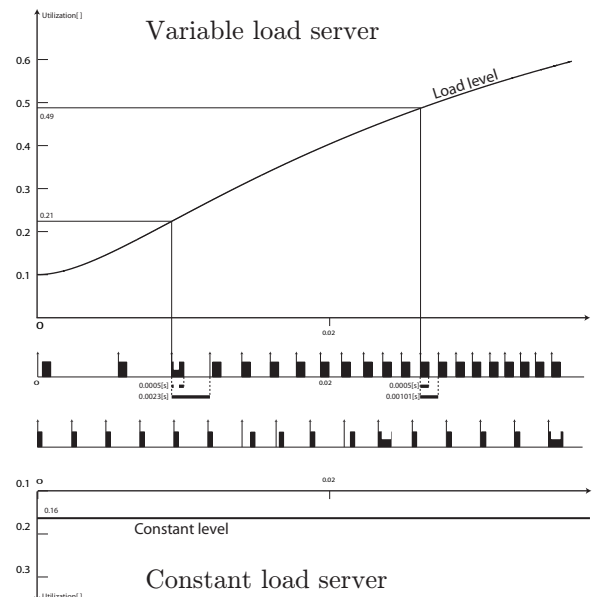
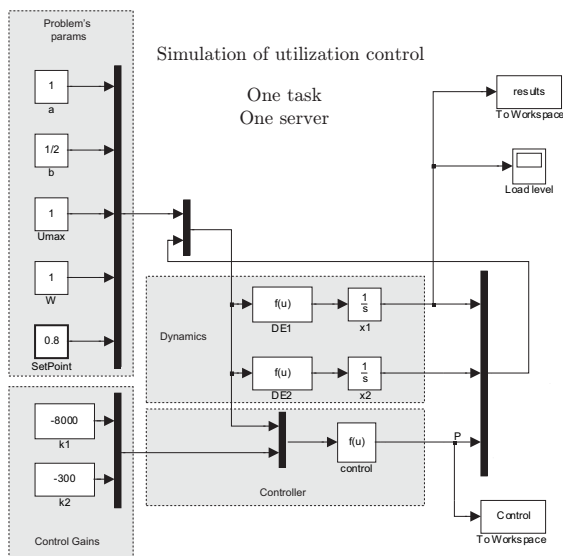
-  $x_1$  is the server utilization factor  
-  $x_2$  is the rate of progress of each task

Parameters:

- $N$ , number of tasks running on the server ( $N=1$ )
- $W$ , sliding window
- $U_{max}$ , maximum utilization
- $U_d$ , setpoint utilization
- $p$ , number of skipped jobs in  $W$
- $a$ , incremental policy of each task.
- $b$ , decremental policy of each task.

## Simulation and Results

Server control law 
$$p(t) = \frac{-8000(x_1 - u_d) - 300\left(\frac{N x_2}{W} - u_{max}\right) - \frac{N a}{W}}{-N\left(a + 2\frac{b x_2}{2-b}\right) x_2 W^{-1}}$$



## Conclusions:

- Simulations show that the load of 1 task in a server can be perfectly controlled
- The controller acts as load demand filter
- Overhead should be investigated
- What about  $N$  tasks in the server?
- Which is the relation between the load control and different scheduling policies?