Decentralized and Dynamic Bandwidth Allocation in Networked Control Systems

Ahmad T. Al-Hammouri, Michael S. Branicky, Vincenzo Liberatore
Case Western Reserve University

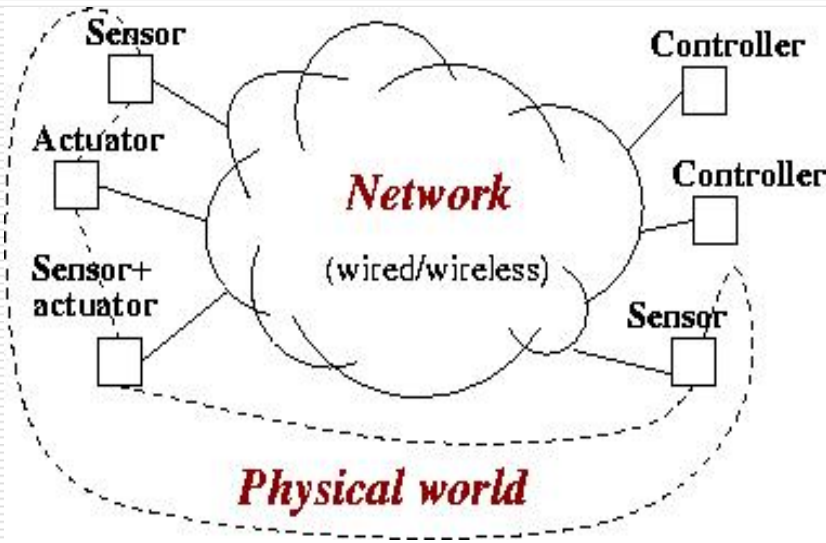
Stephen M. Phillips
Arizona State University

April 25, 2006

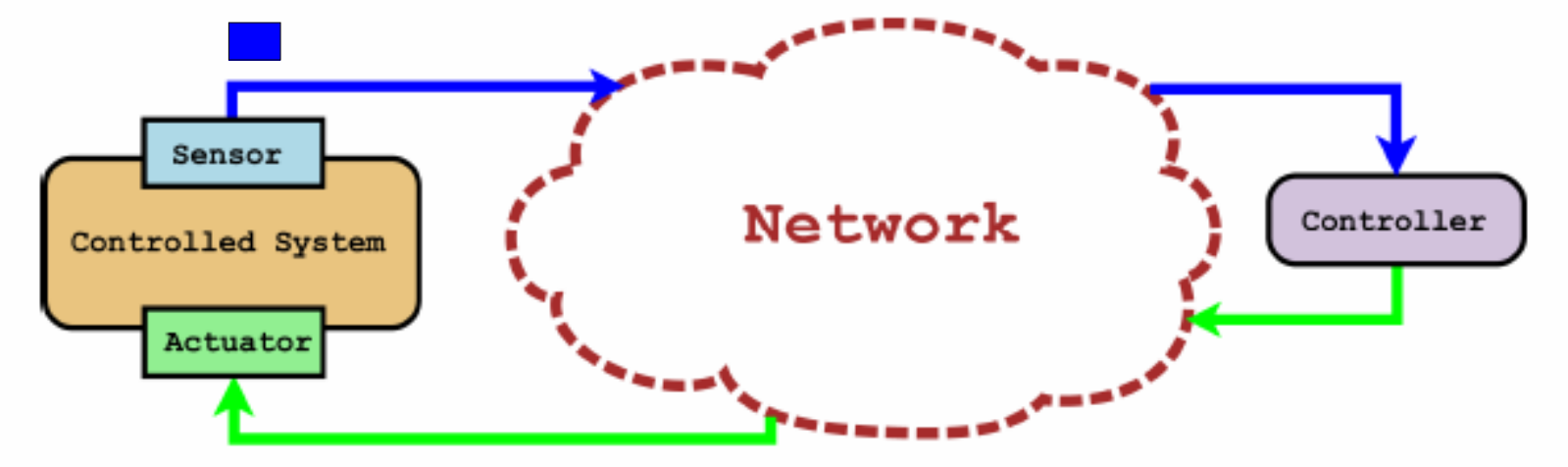
Support by: **NSF** CCR-0329910, **Department of Commerce** TOP 39-60-04003, **NASA** NNC04AA12A, and an **OhioICE** Training grant

Paper Overview

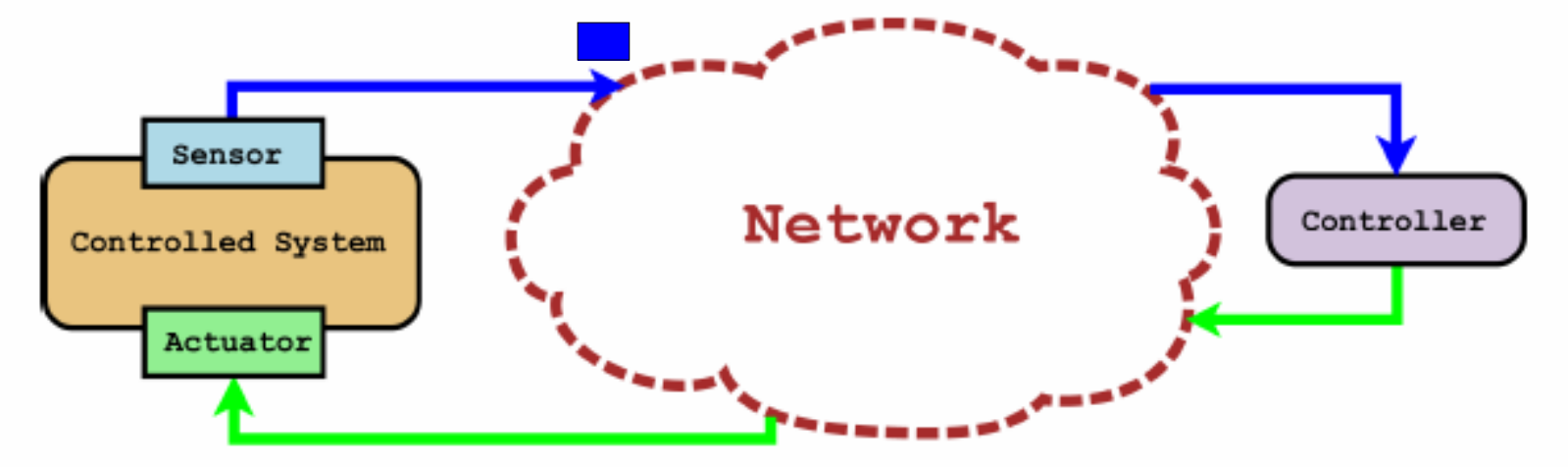
- Control ***over*** Networks
  - NCSs, DCSs, SANETs, …
- Control ***of*** Networks
  - Efficient BW allocation
    - Congestion control
  - Fairness
- We propose a "C***of*** N" scheme to better serve "C***over*** N"



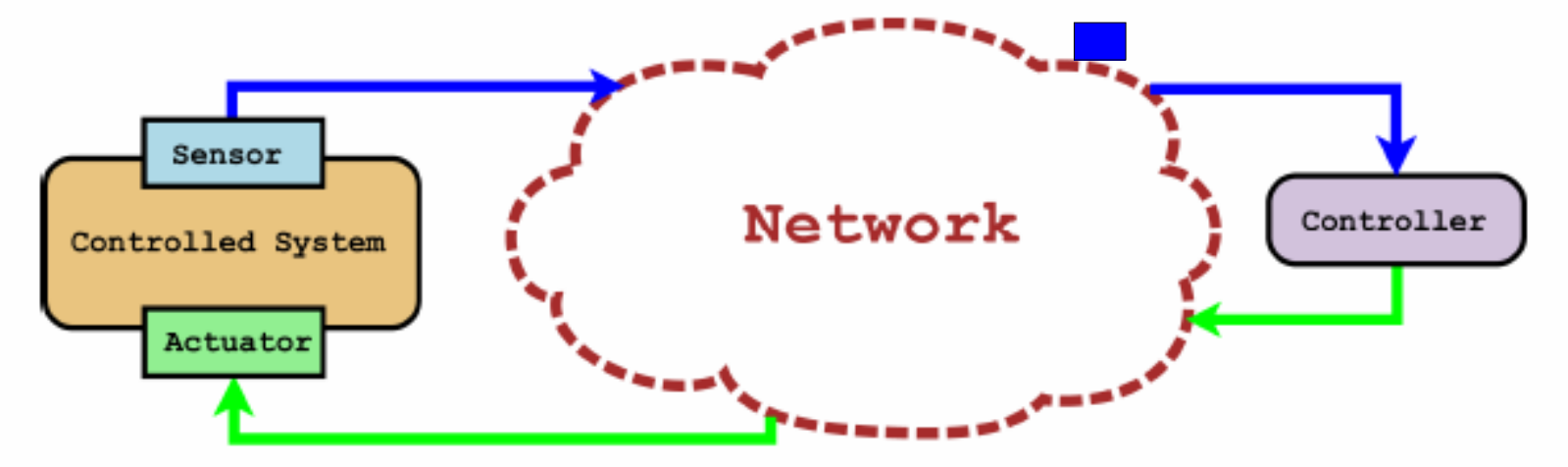
Control ***over*** Networks



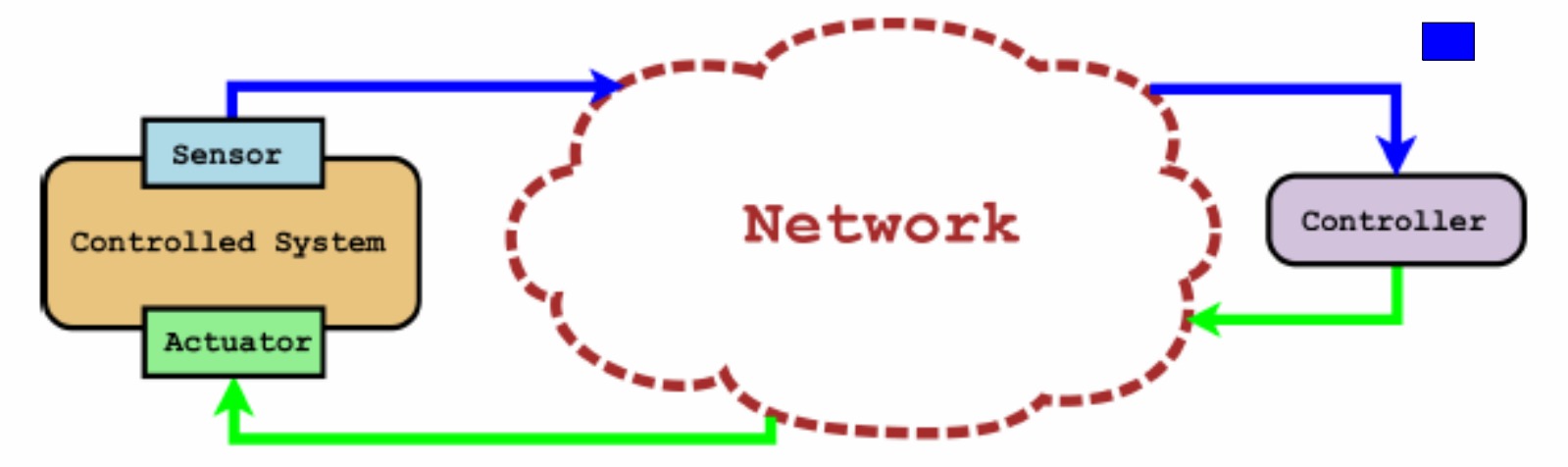
Control ***over*** Networks



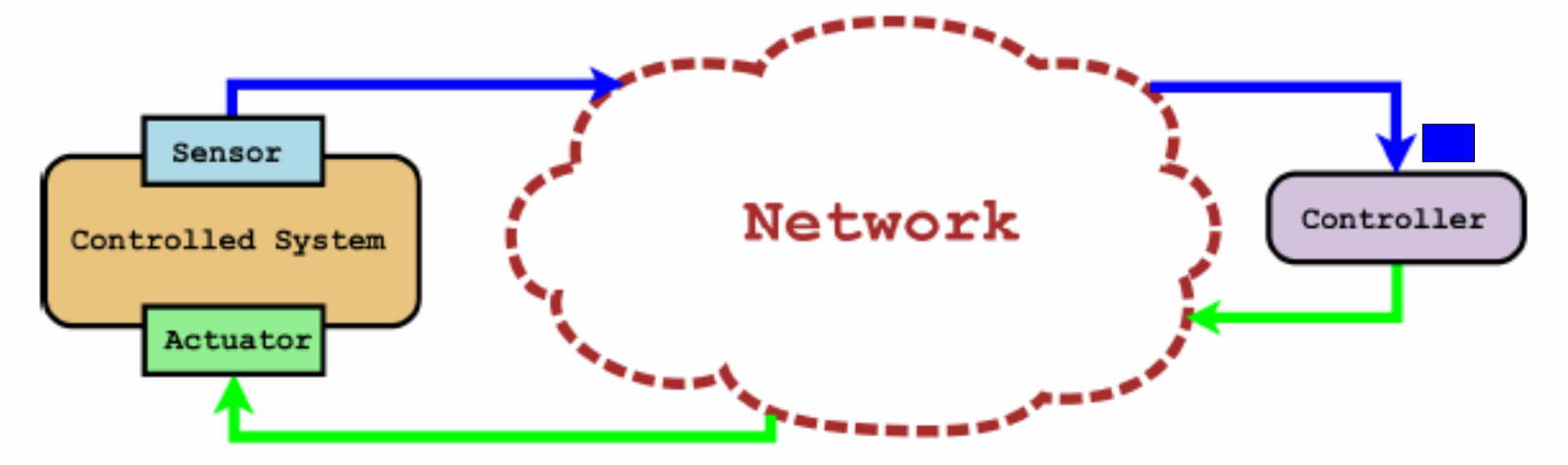
Control ***over*** Networks



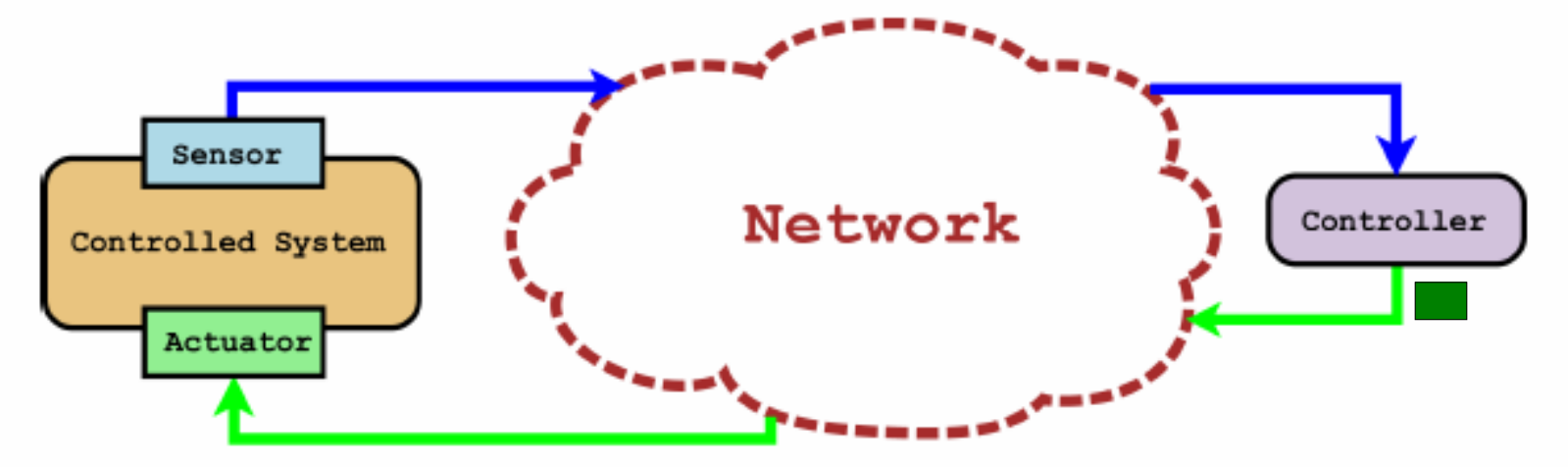
Control ***over*** Networks



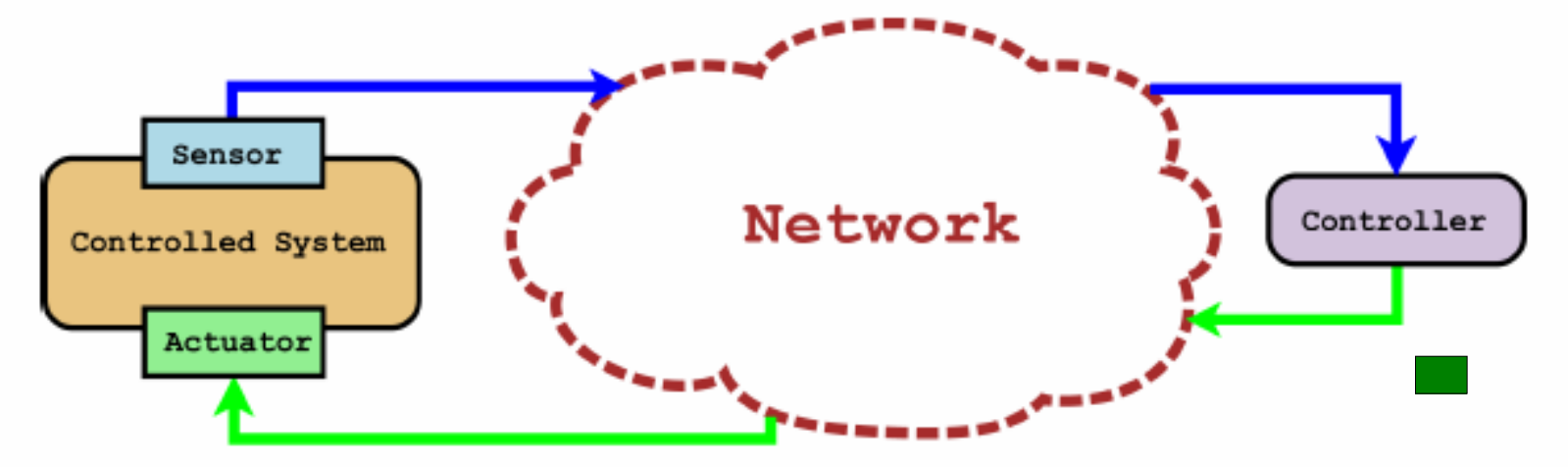
Control ***over*** Networks



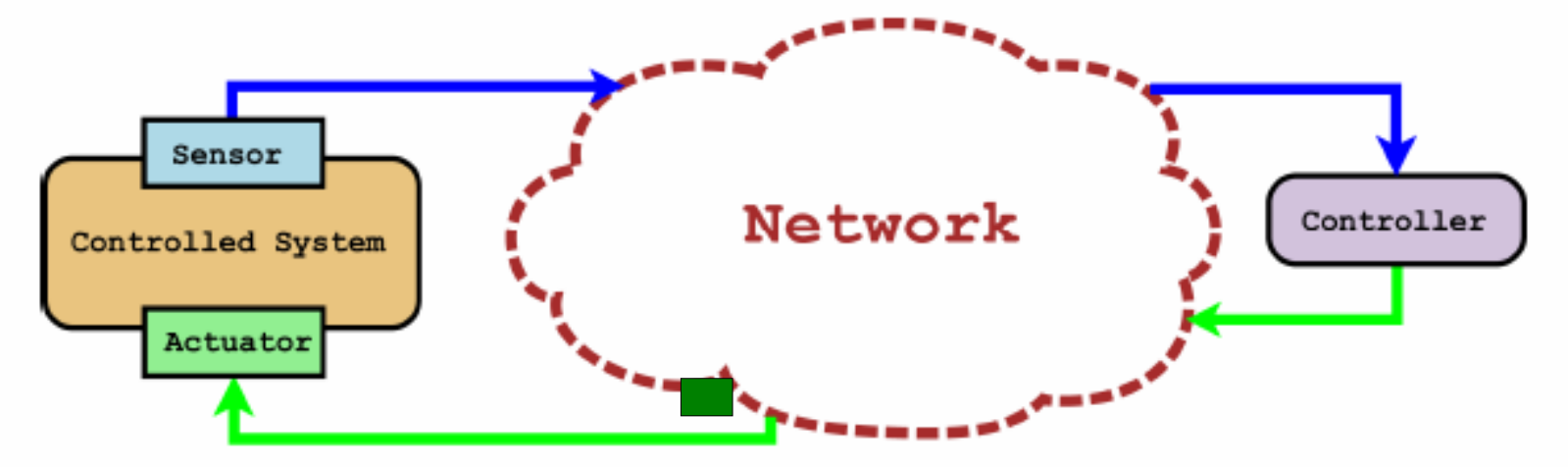
Control ***over*** Networks



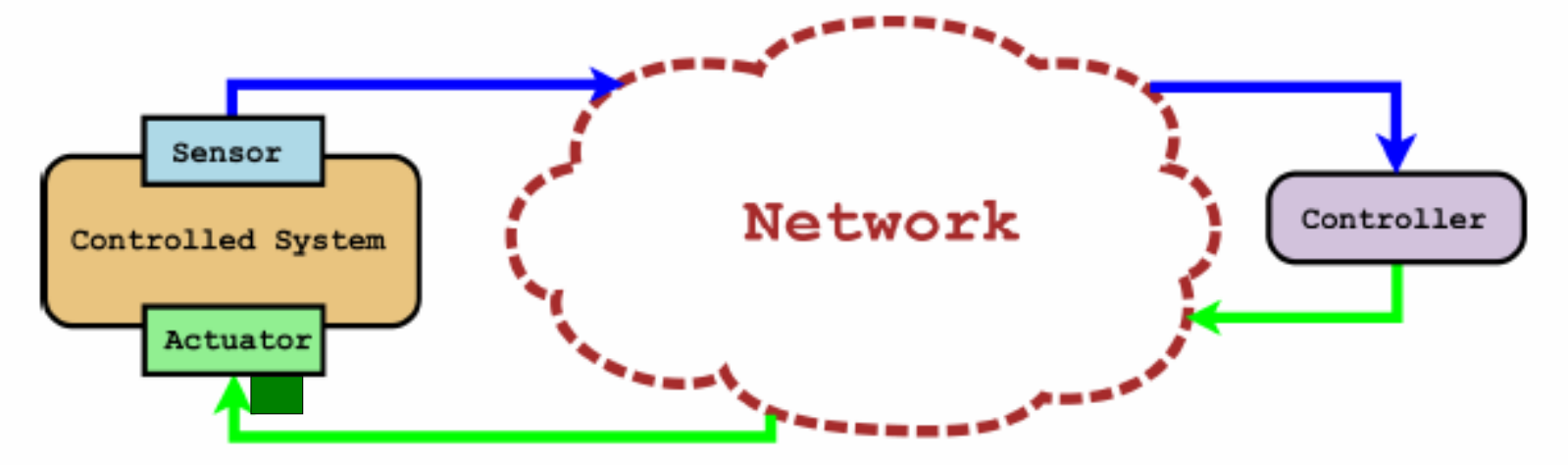
Control ***over*** Networks



Control ***over*** Networks



Control ***over*** Networks



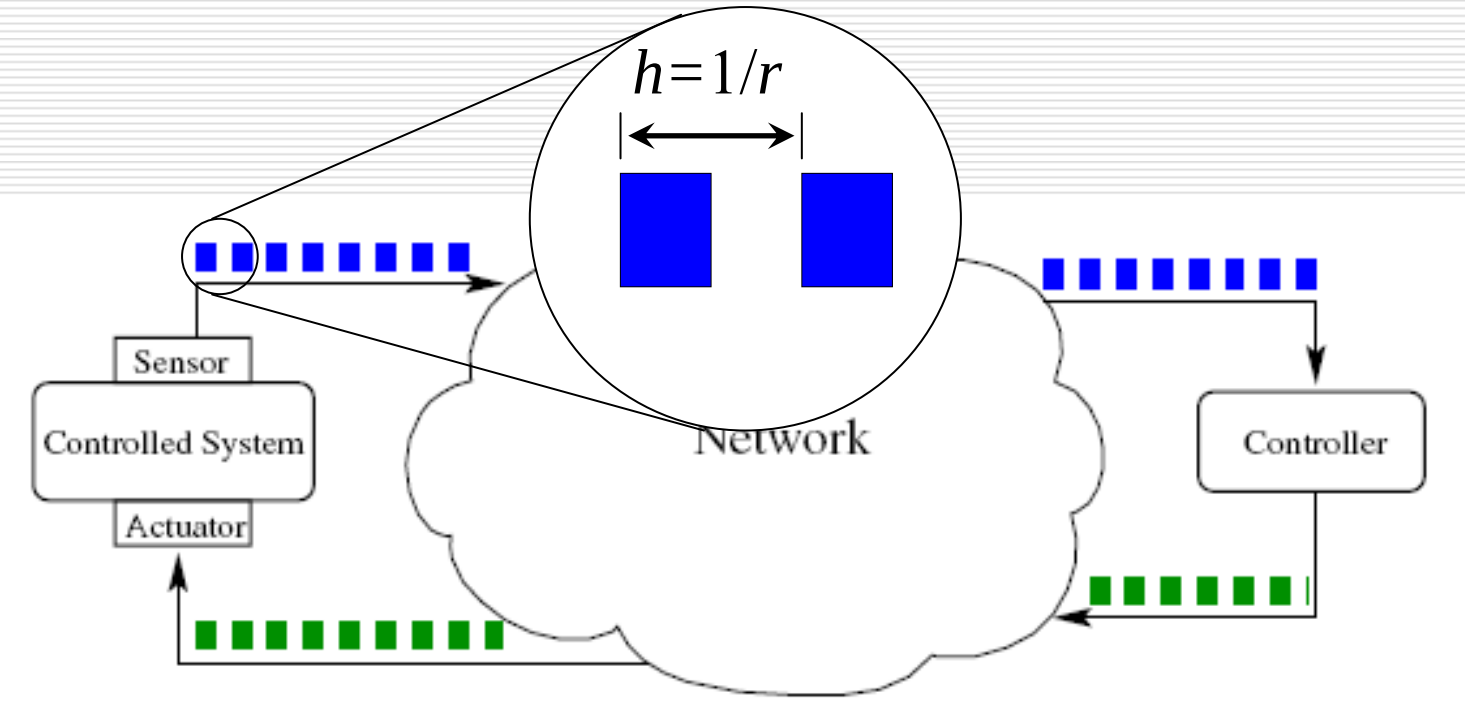
Control ***over*** Networks

- Remote interaction (monitoring & control) with the physical world
- Applications:
  - Industrial automation & process control
  - Space exploration, e.g., telerobotics
  - Smart homes
  - Medical sensing
    & surgical simulations
  - Automatic asset mgmt. (RFID)

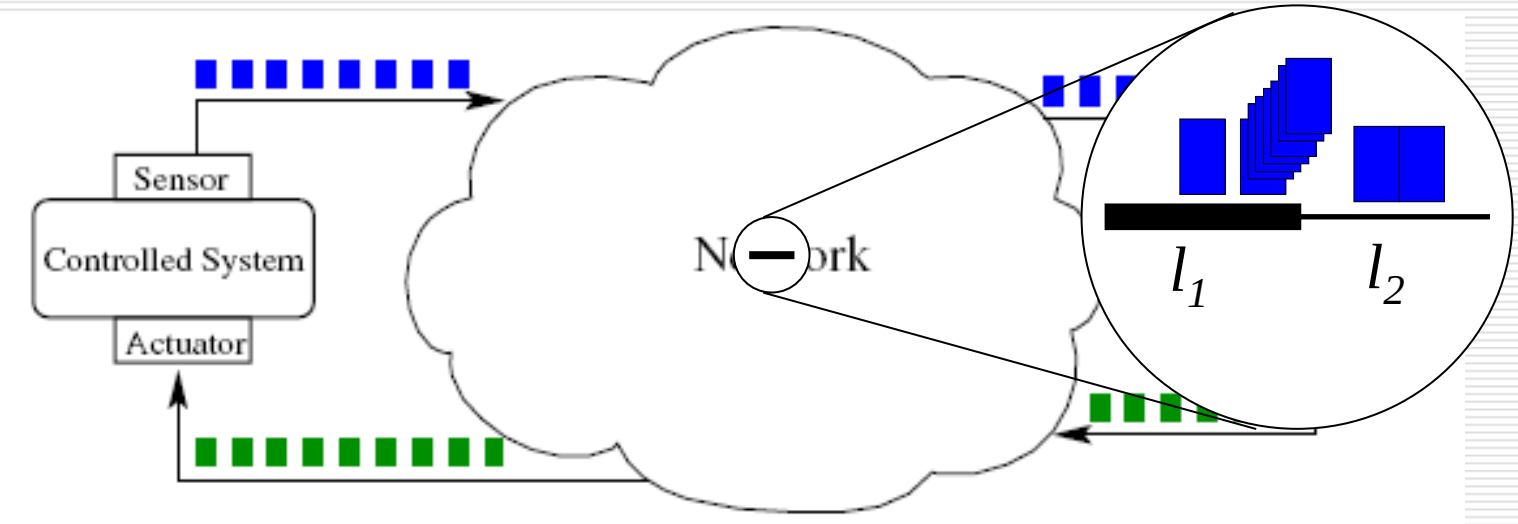
Control ***of*** Networks (Scope of the paper)

- A bandwidth allocation scheme
- Objectives:
  - Stability of control systems
  - Efficiency & fairness
  - Fully distributed, asynchronous, & scalable
  - Dynamic & self reconfigurable
- Formulating the scheme in CT
  - NCSs regulate $\boldsymbol{h}$ based on congestion fed back from the network

Sampling Rate & Network Congestion

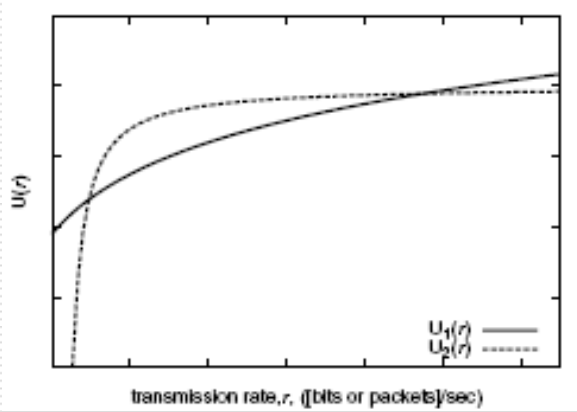


Sampling Rate & Network Congestion



Problem Formulation

- Define a utility fn $U(r)$ that is
  - Monotonically increasing
  - Strictly concave
  - Defined for $r \geq r_{\min}$

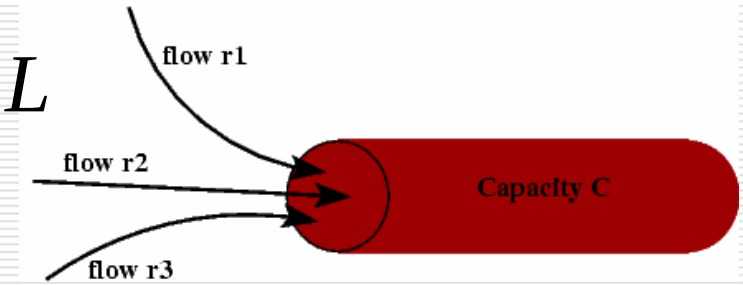


- Optimization formulation

$$\max \sum_i U_i(r_i)$$

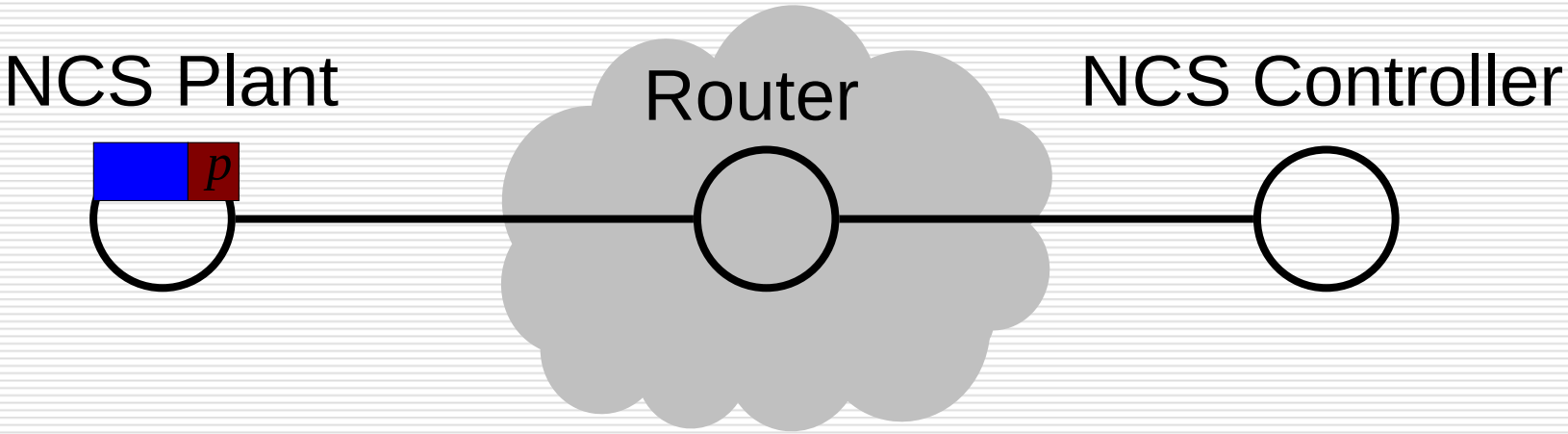
$$\text{s.t.} \sum_{i \in \mathbb{S}(l)} r_i \leq C_l, l = 1, \ldots, L$$

$$\text{and } r_i \geq r_{\min,i}$$



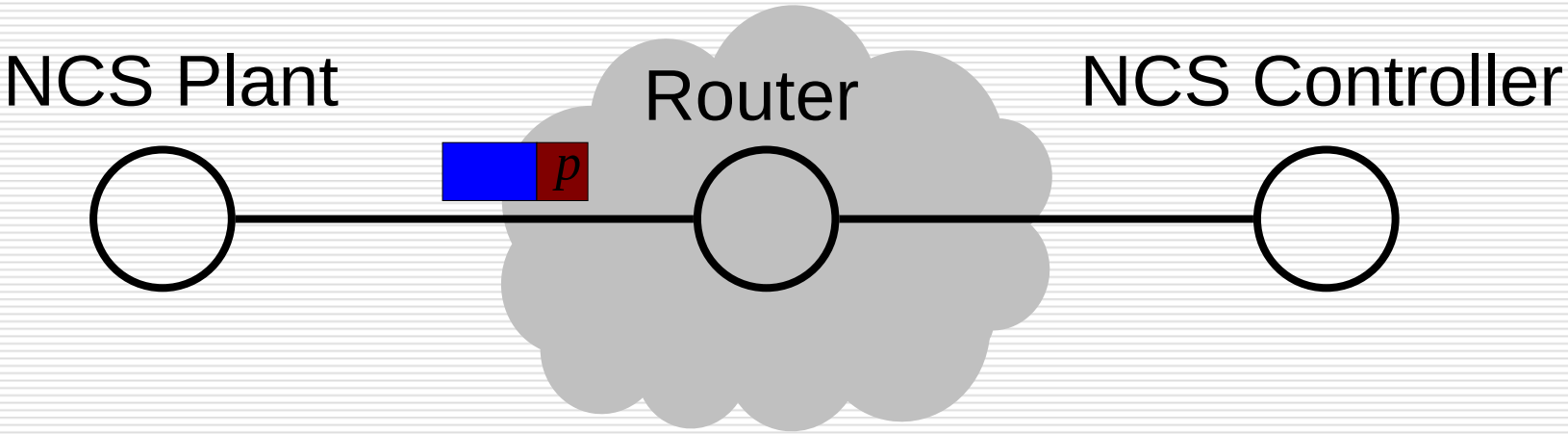
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



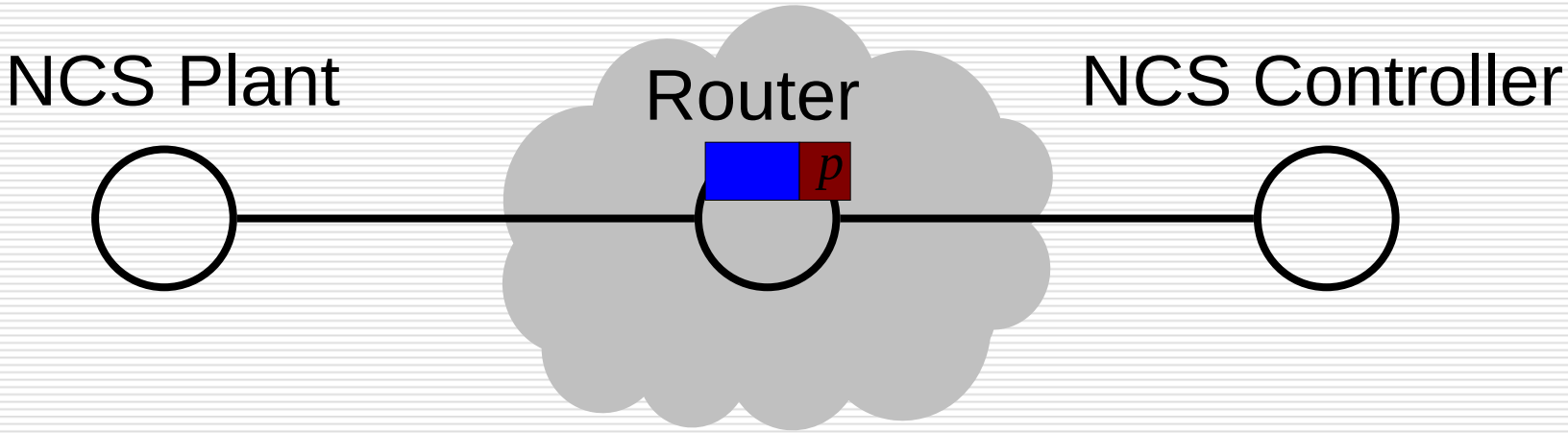
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



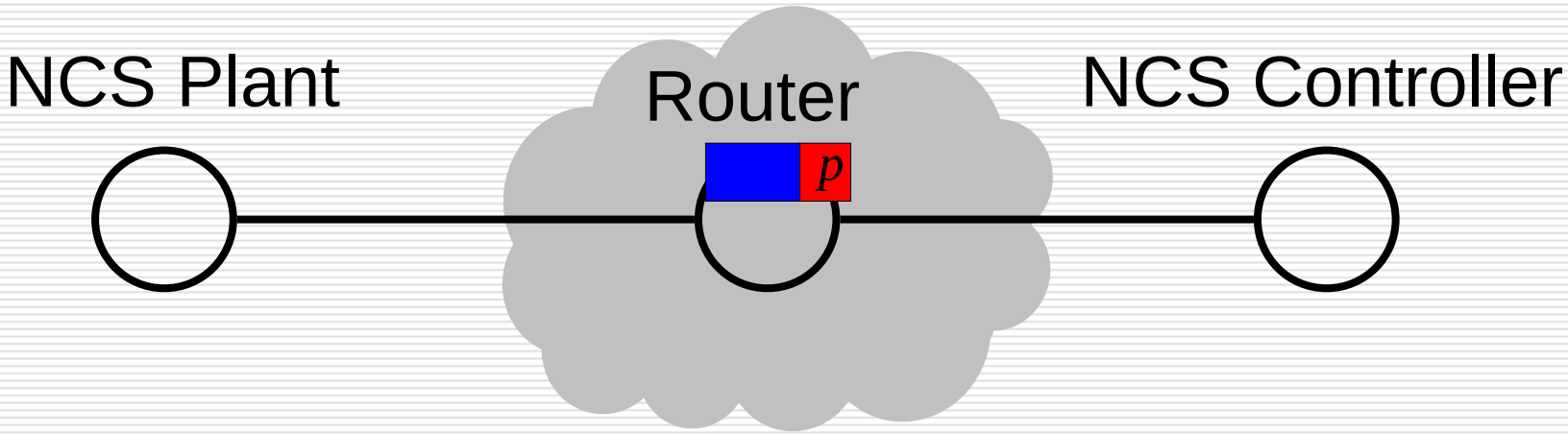
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



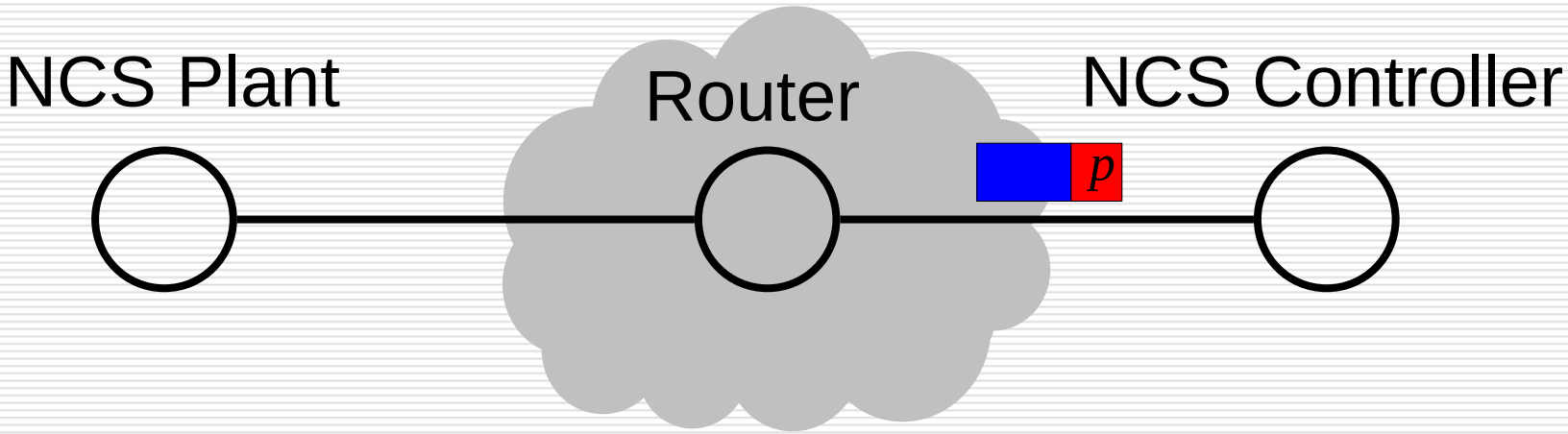
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



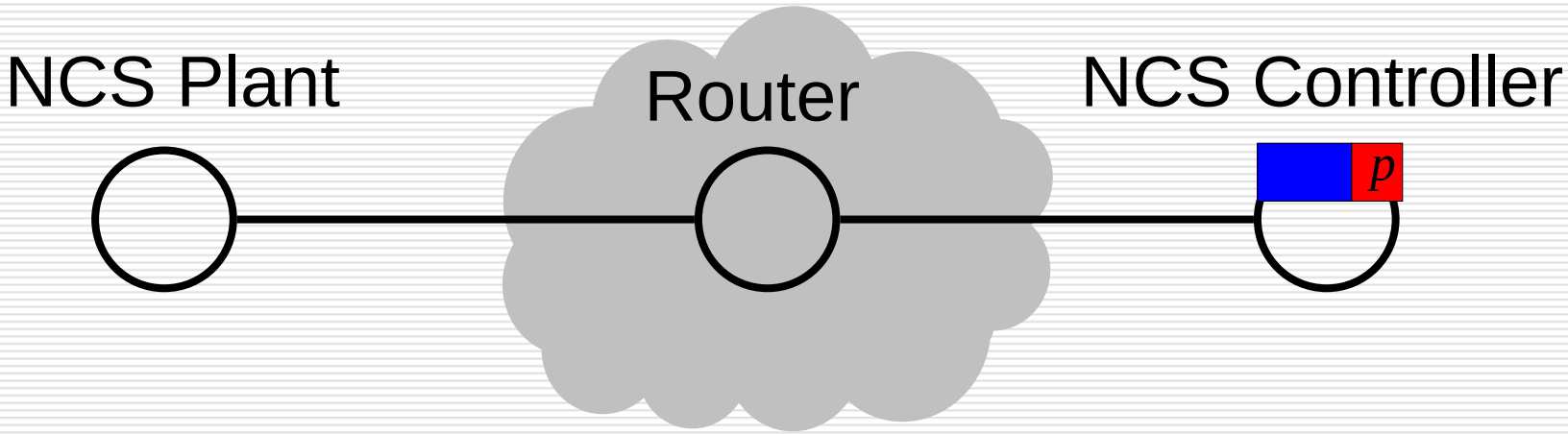
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



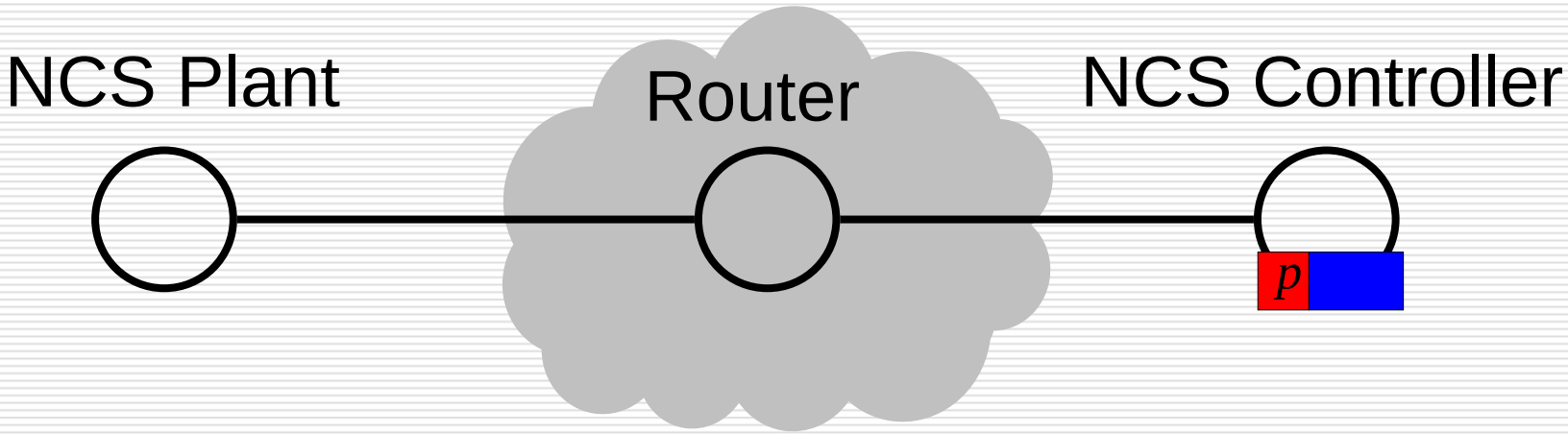
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



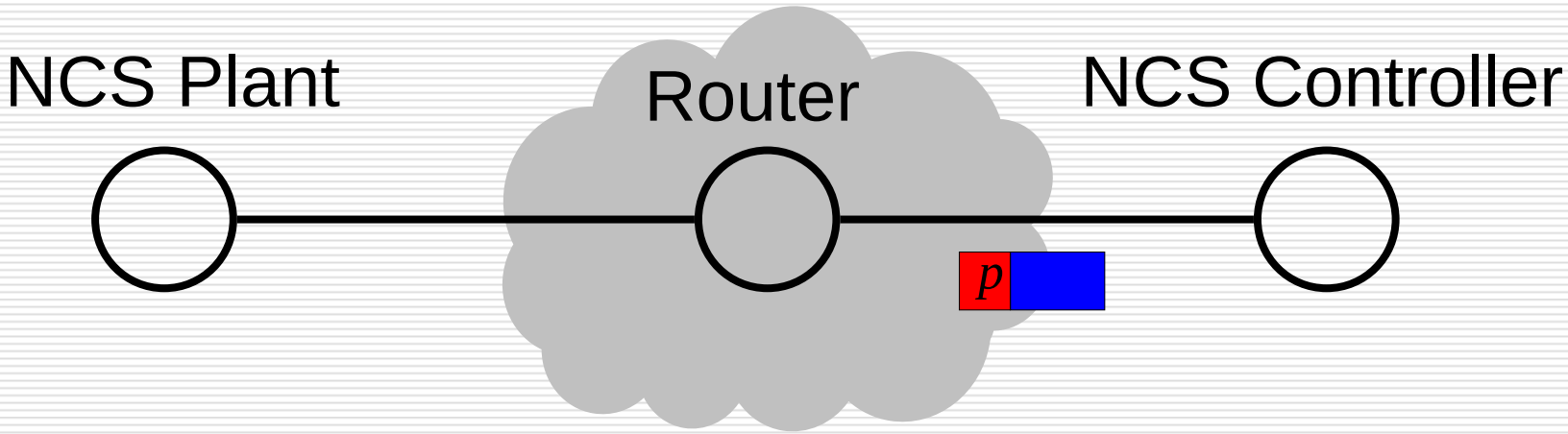
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



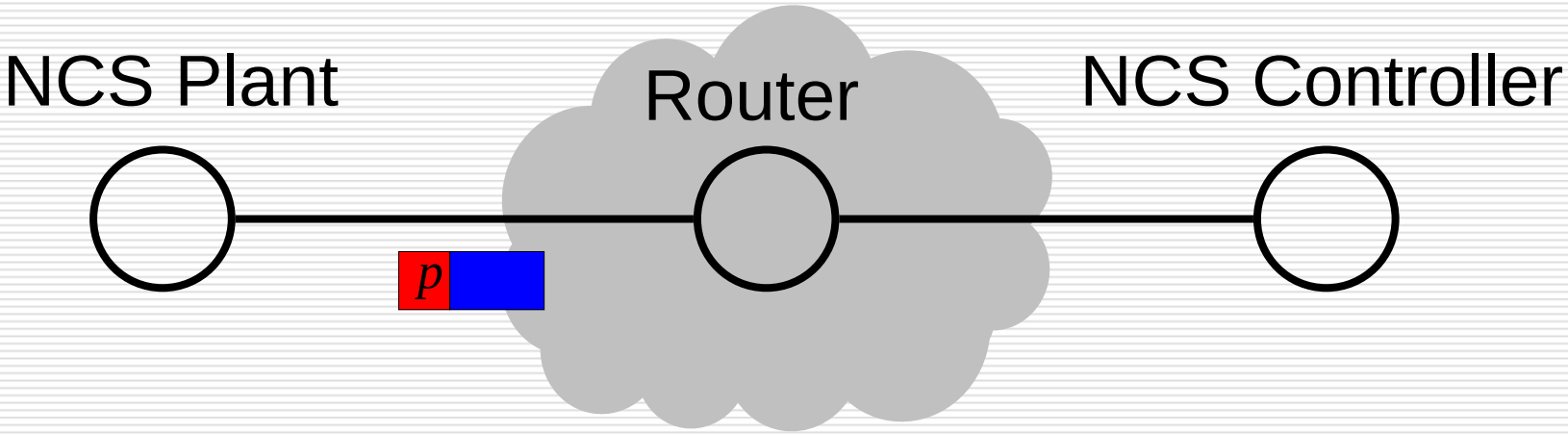
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



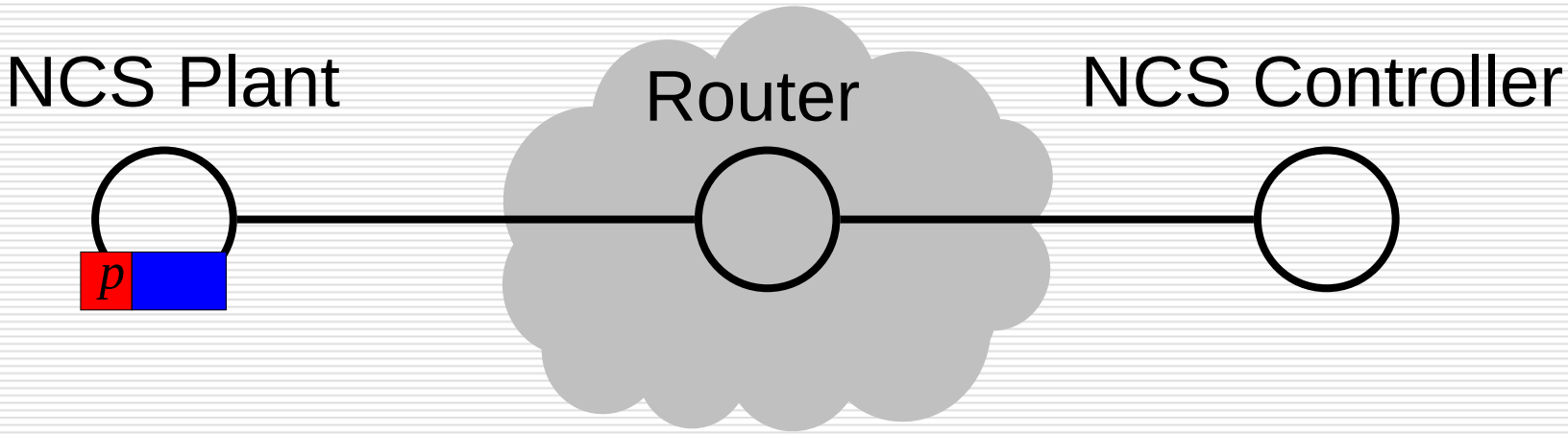
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



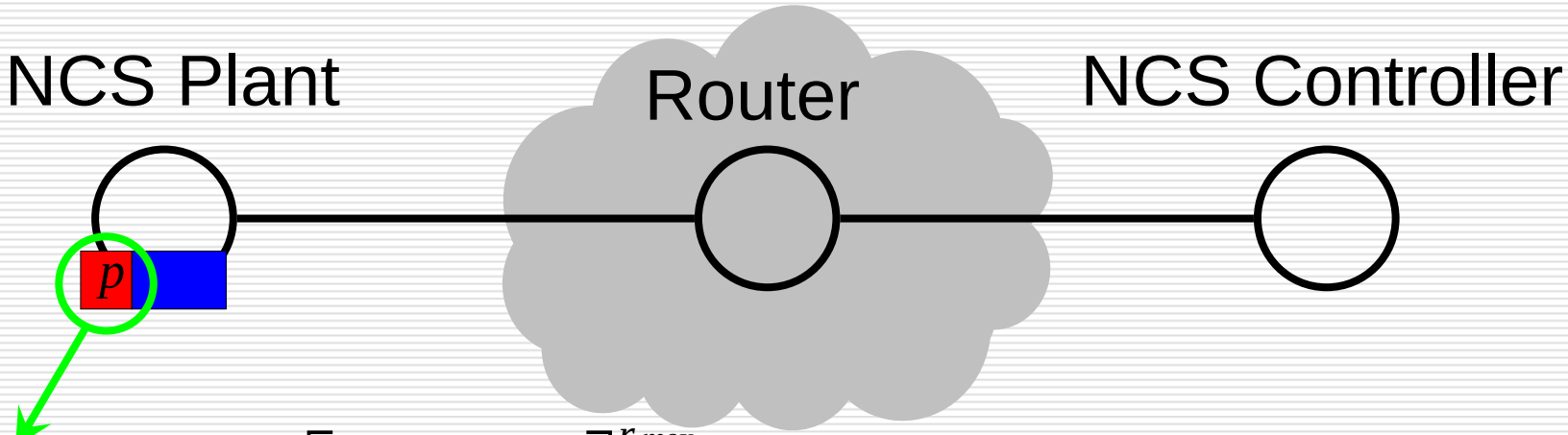
Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



Distributed Implementation

- Two independent algorithms
  - End-systems (plants) algorithm
  - Router algorithm (later on)



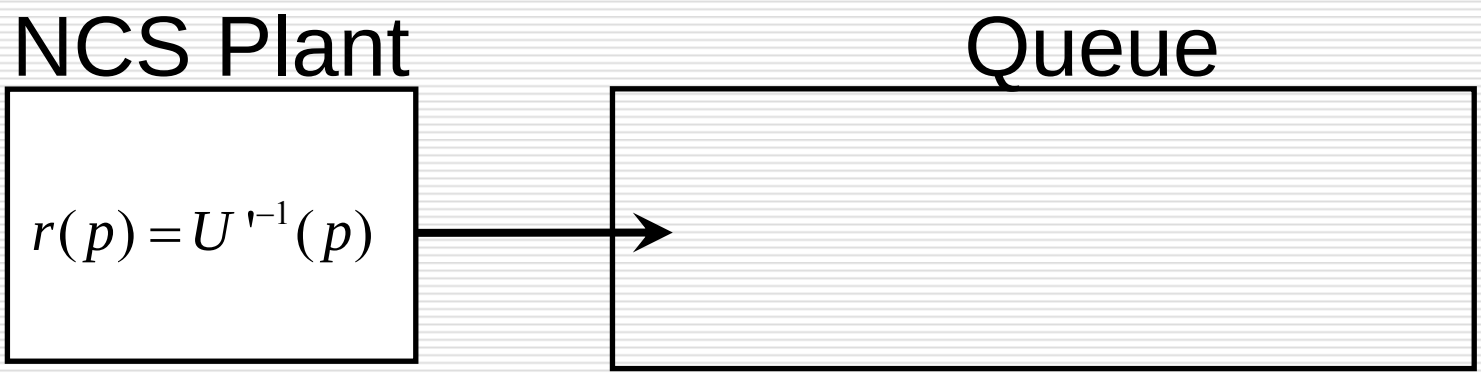
$$r(p_t) = 1/h = \left[U'^{-1}(p_t) \right]_{r_{\min}}^{r_{\max}}$$

NCS-AQM Control Loop

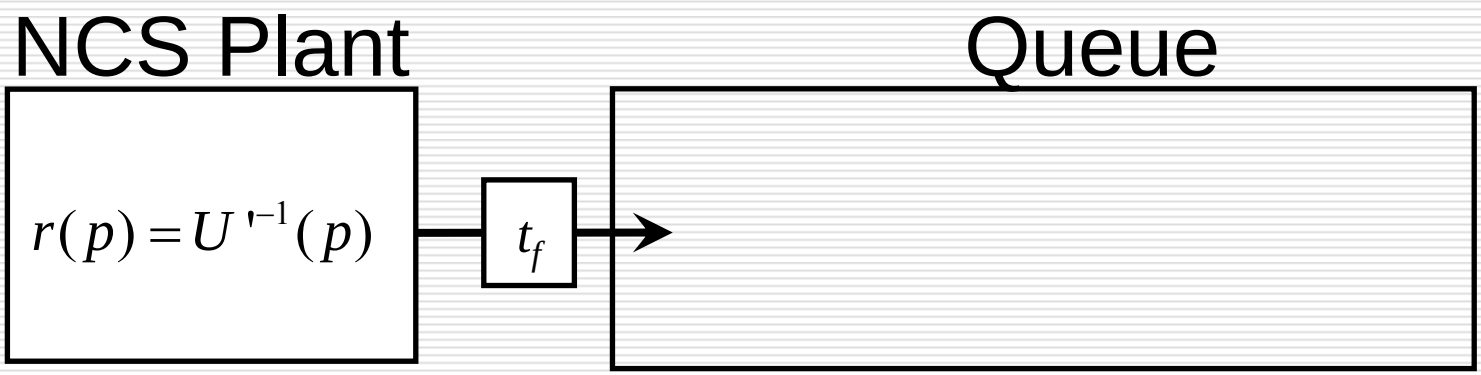
NCS Plant

$$r(p) = U'^{-1}(p)$$

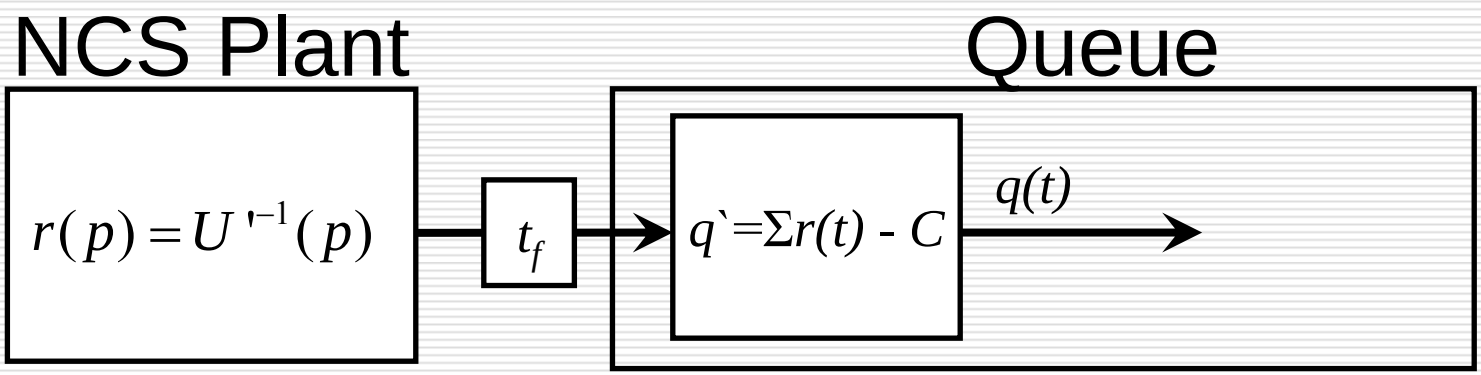
NCS-AQM Control Loop



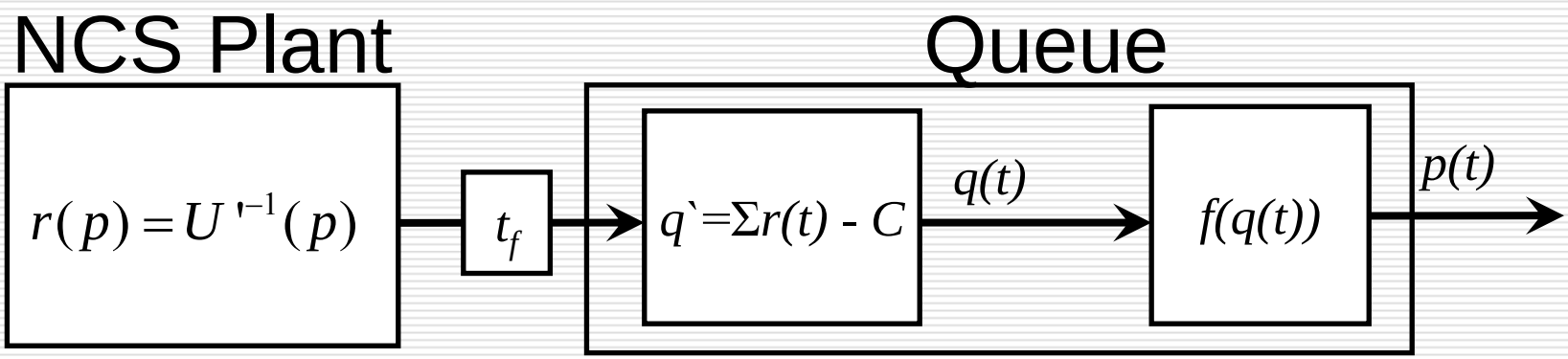
NCS-AQM Control Loop



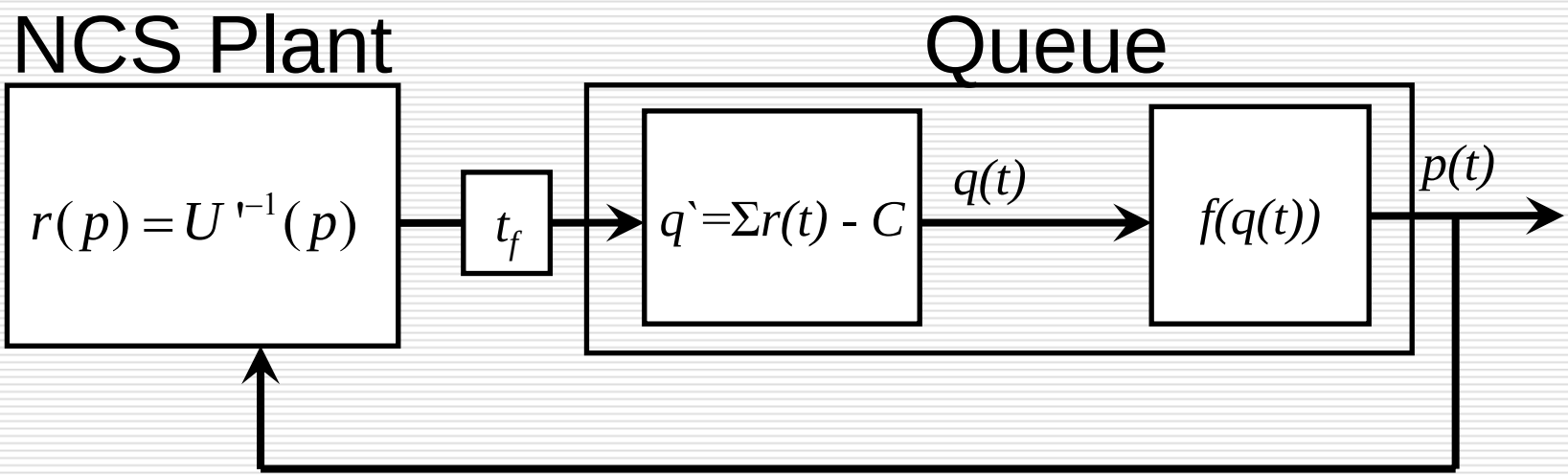
NCS-AQM Control Loop



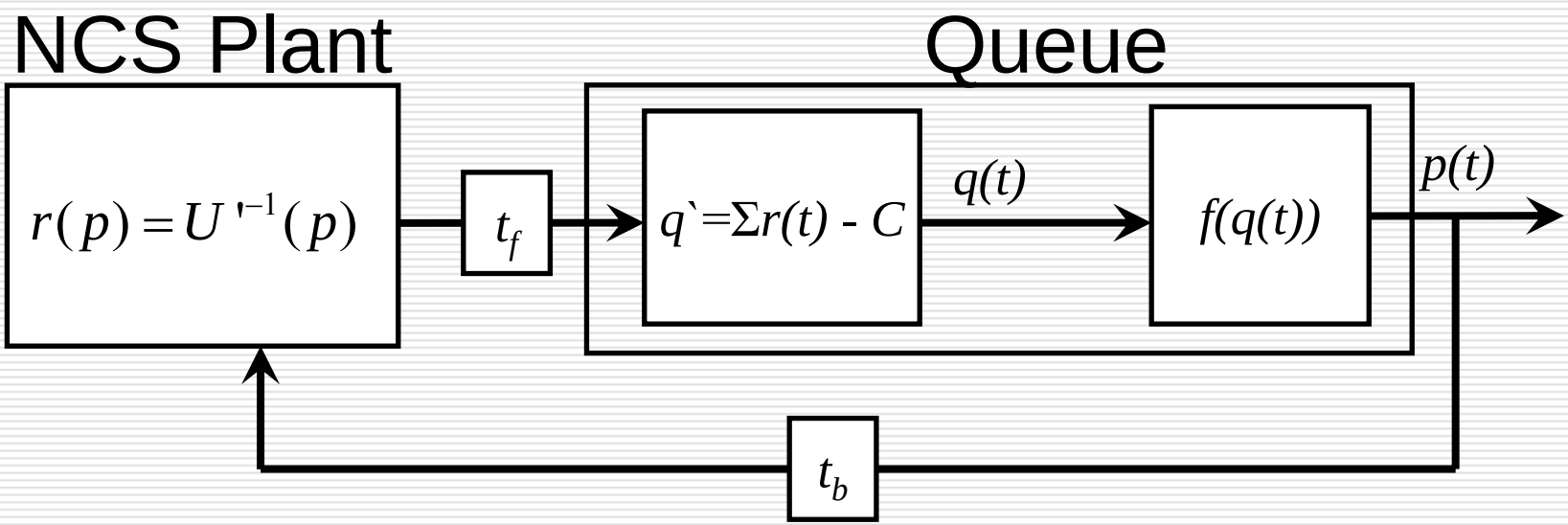
NCS-AQM Control Loop



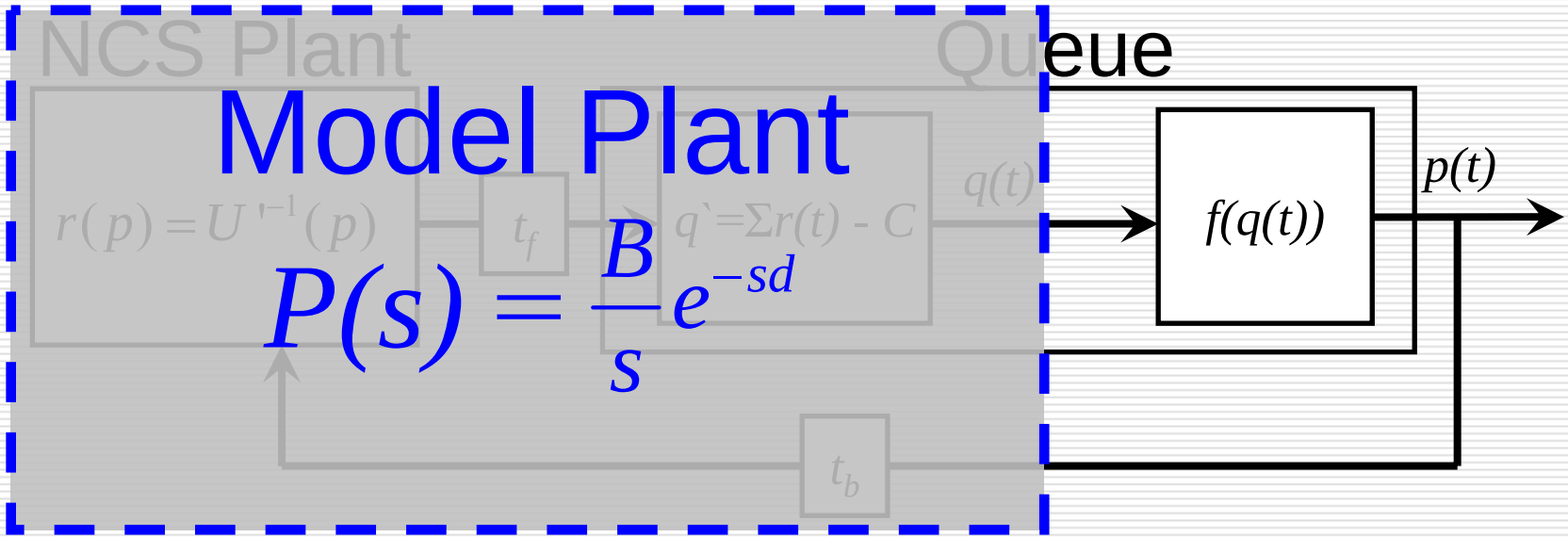
NCS-AQM Control Loop



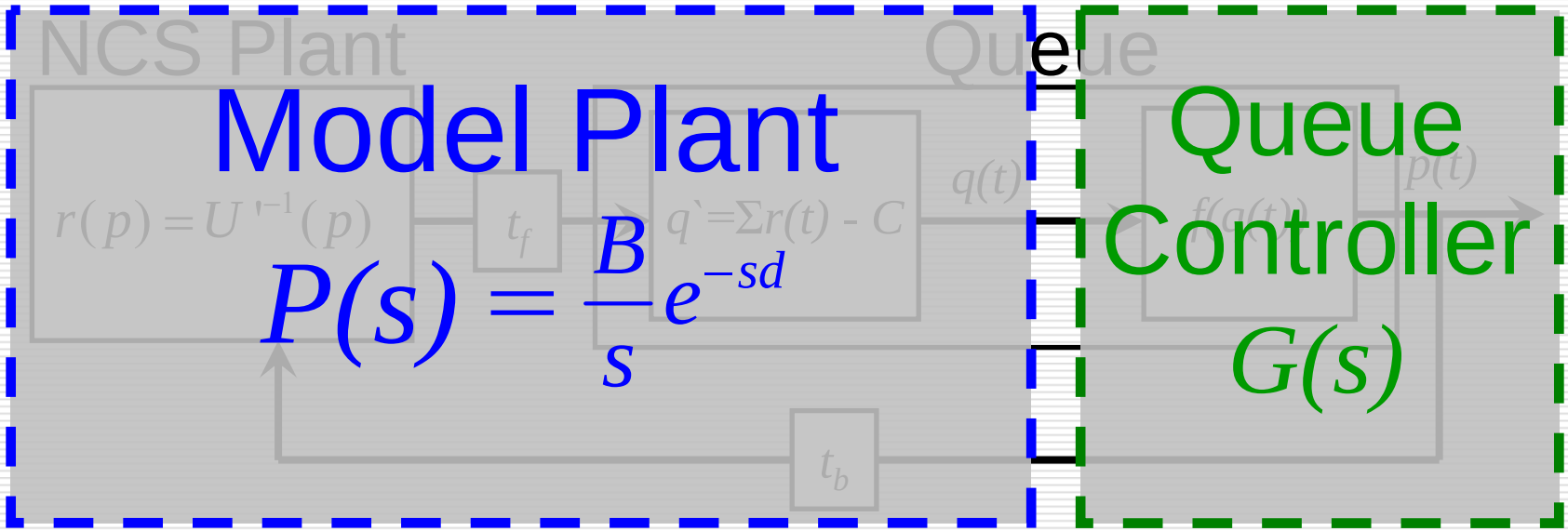
NCS-AQM Control Loop



NCS-AQM Control Loop

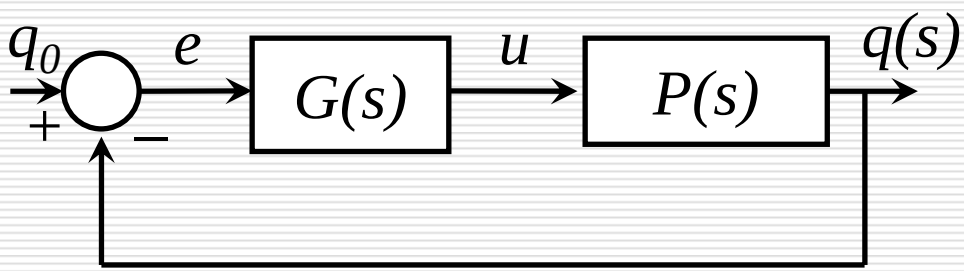


NCS-AQM Control Loop



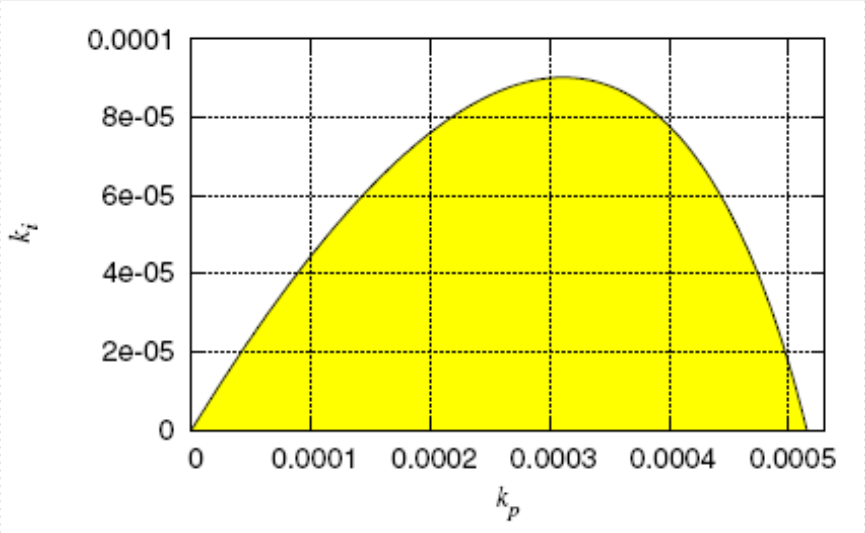
Queue Controller $G(s)$

- Proportional (P) Controller
  - $G_P(s) = k_p$
- Proportional-Integral (PI) Controller
  - $G_{PI}(s) = k_p + k_i/s$



Determination of k_p and k_i

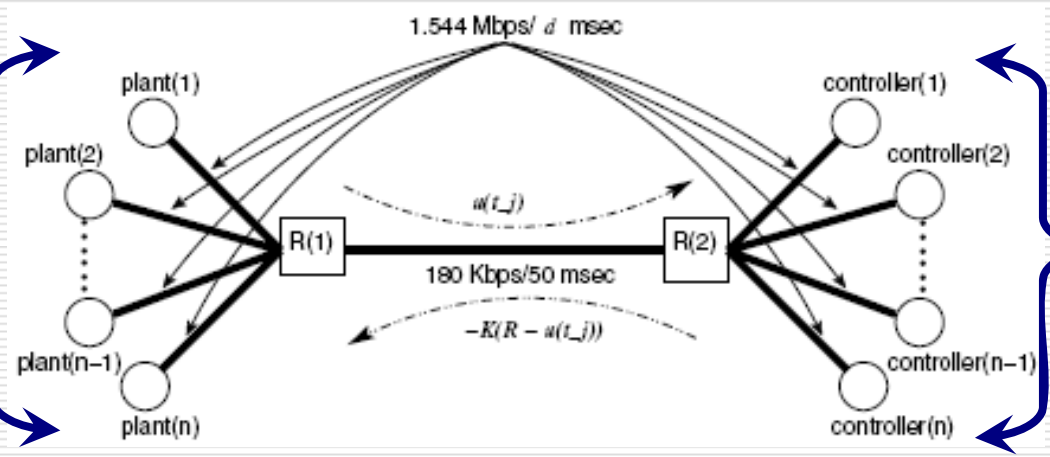
- Stability region in the k_i–k_p plane
  - Stabilizes the NCS-AQM closed-loop system for delays less or equal d
- Analysis of quasi-polynomials, $f(s,e^s)$



Simulations & Results

50 NCS Plants:

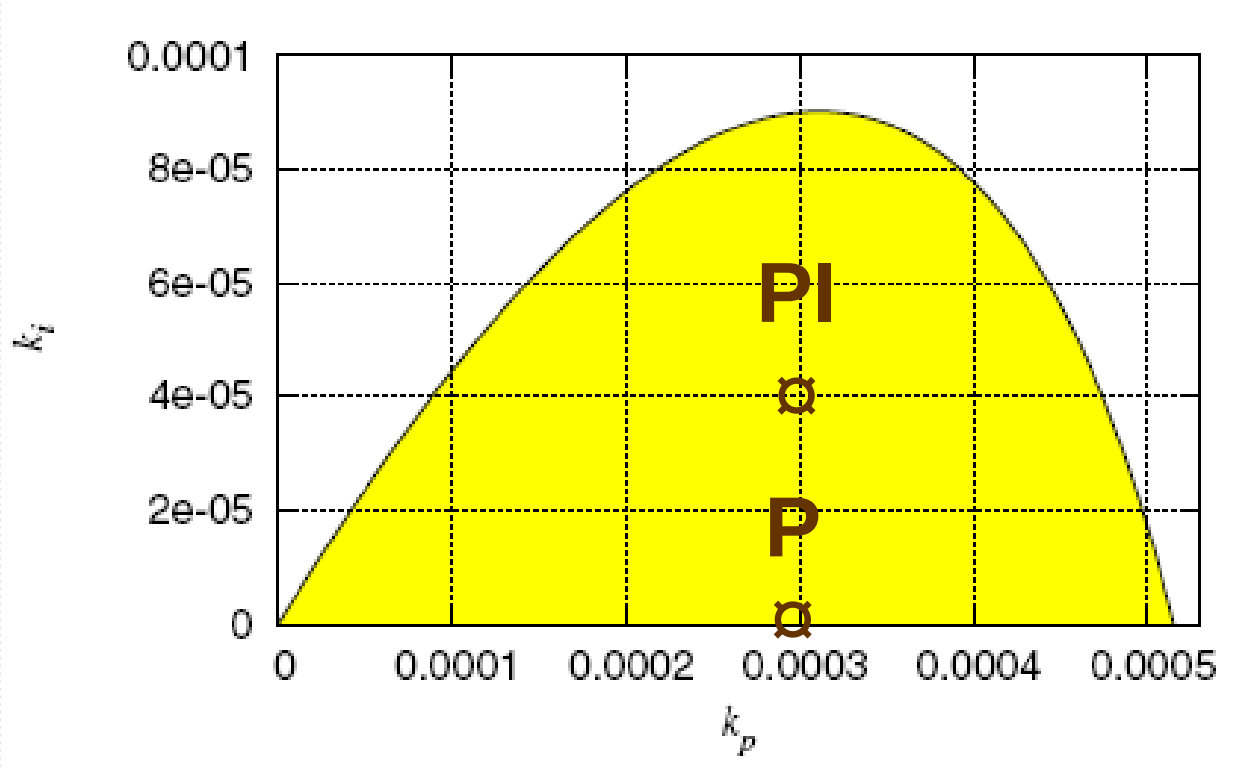
- $\frac{dx}{dt} = ax(t) + bu(t)$
- $U(r) = \frac{a - bK}{a} e^{a/r}$
- $r_{\min} = \frac{a}{\ln \frac{bK + a}{bK - a}}$



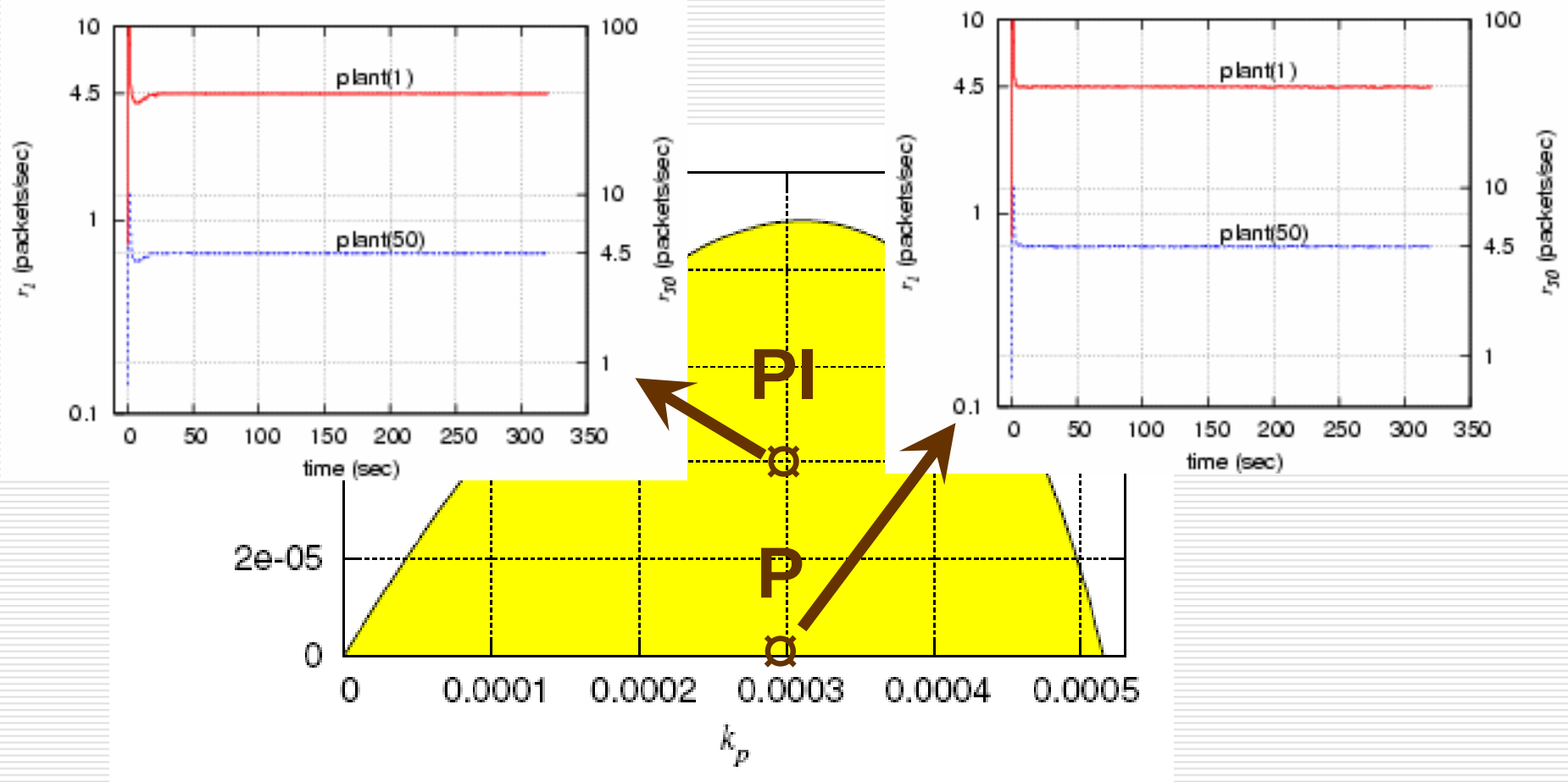
$u(t_j) = -K(R - x(t_j))$

- [Branicky et al. 2002]
- [Zhang et al. 2001]

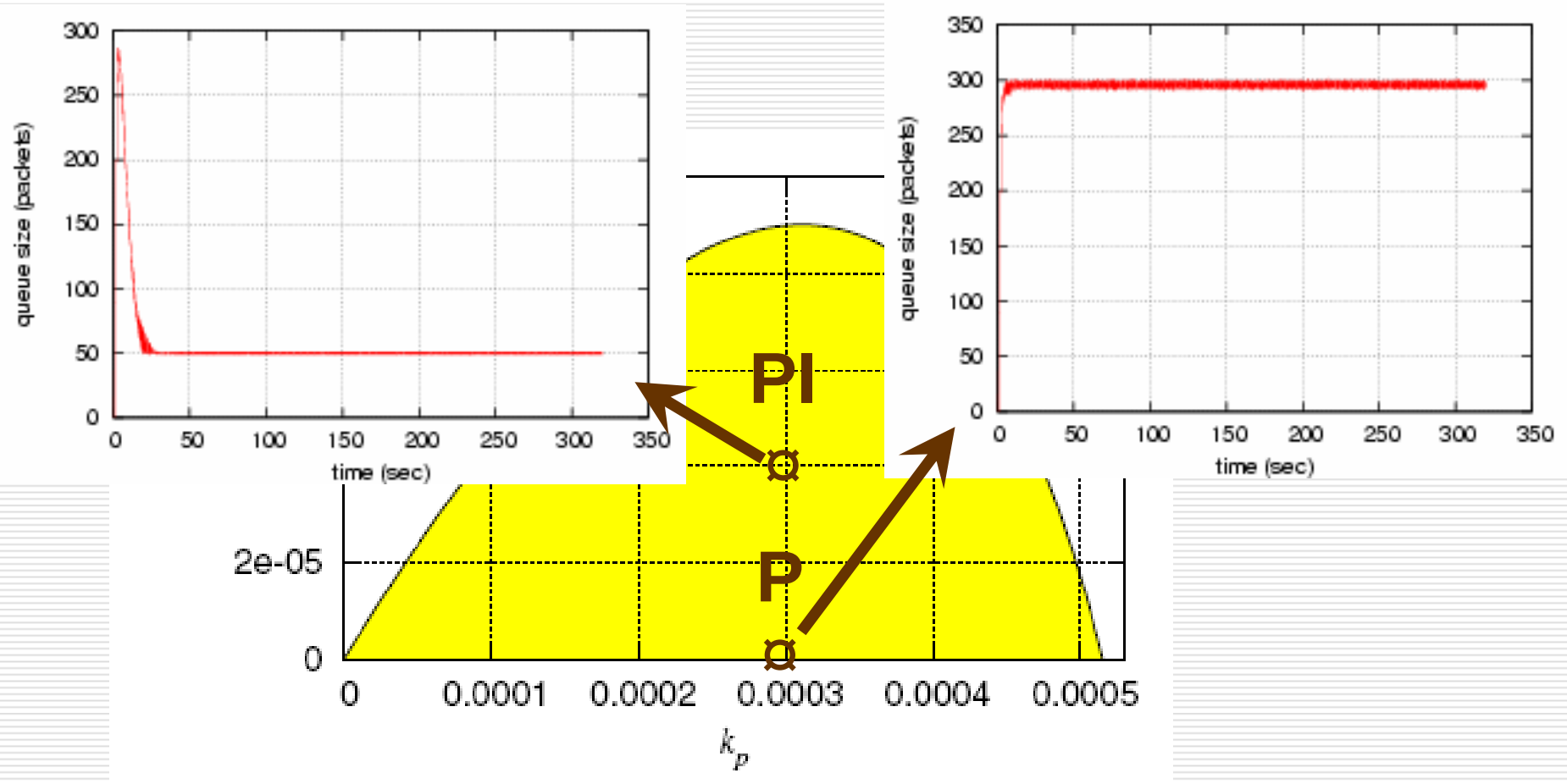
Simulations & Results (cont.)



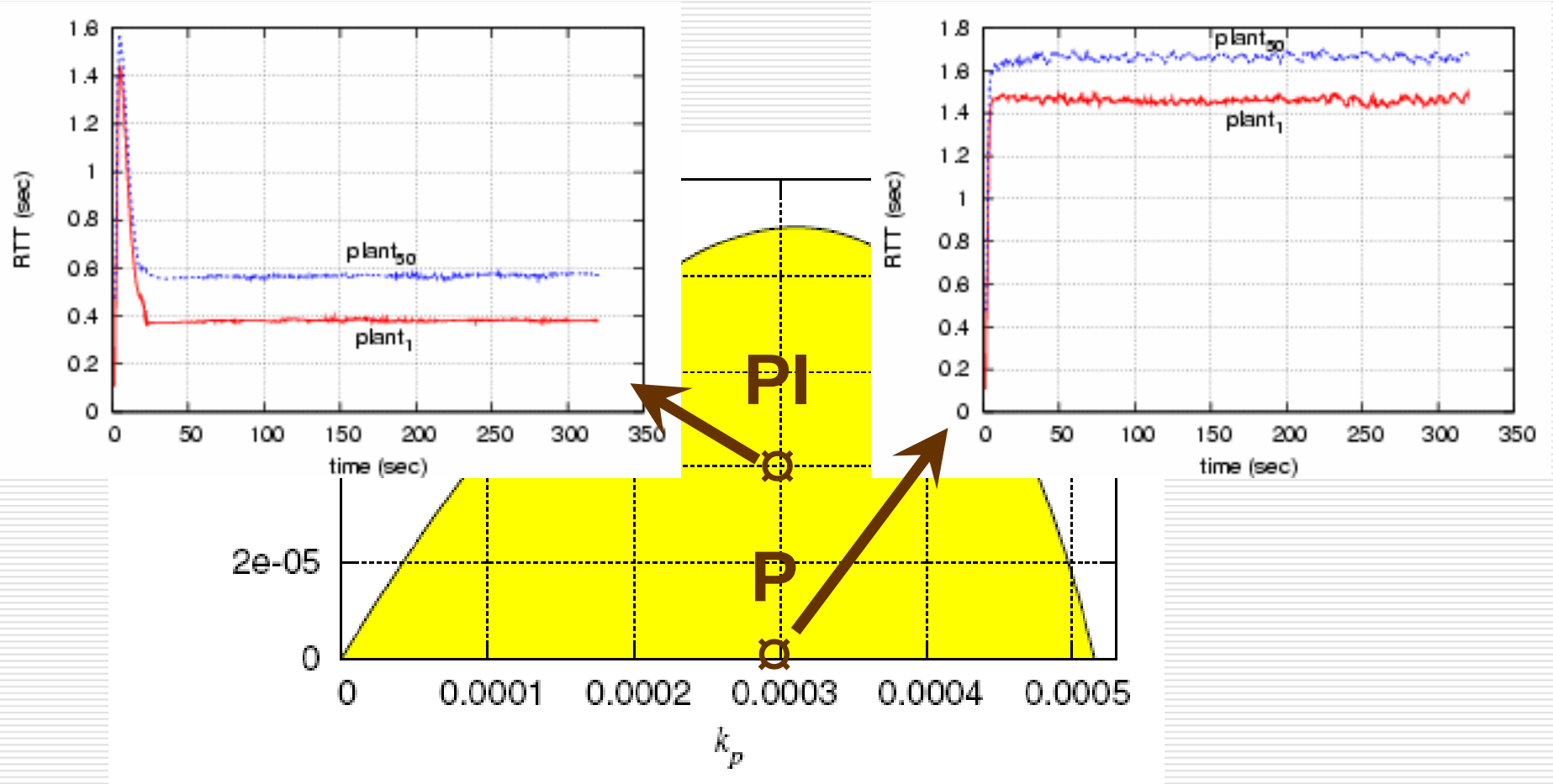
Simulations & Results (cont.)



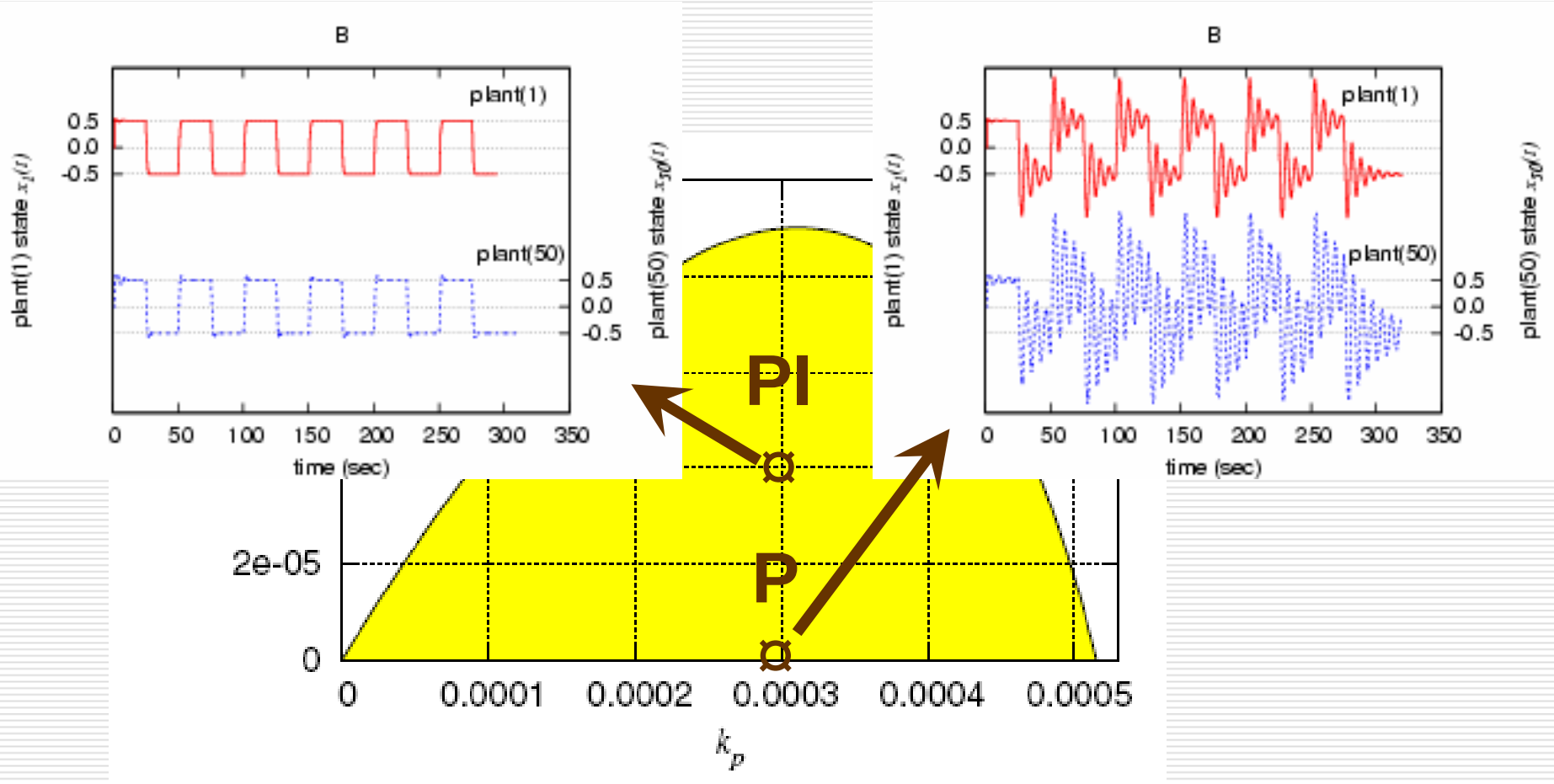
Simulations & Results (cont.)



Simulations & Results (cont.)



Simulations & Results (cont.)



Thank You

- Questions
- Comments

- Probing further:

`http://start.case.edu/~vxl11/NetBots`