



# Processing Enhancement and Virtualization for Cyber-Physical Computations

Dionisio de Niz and Bjorn Andersson

Next Generation Operating Systems for CPS Workshop

April 15<sup>th</sup>, 2019

Copyright 2019 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

**NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.**

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

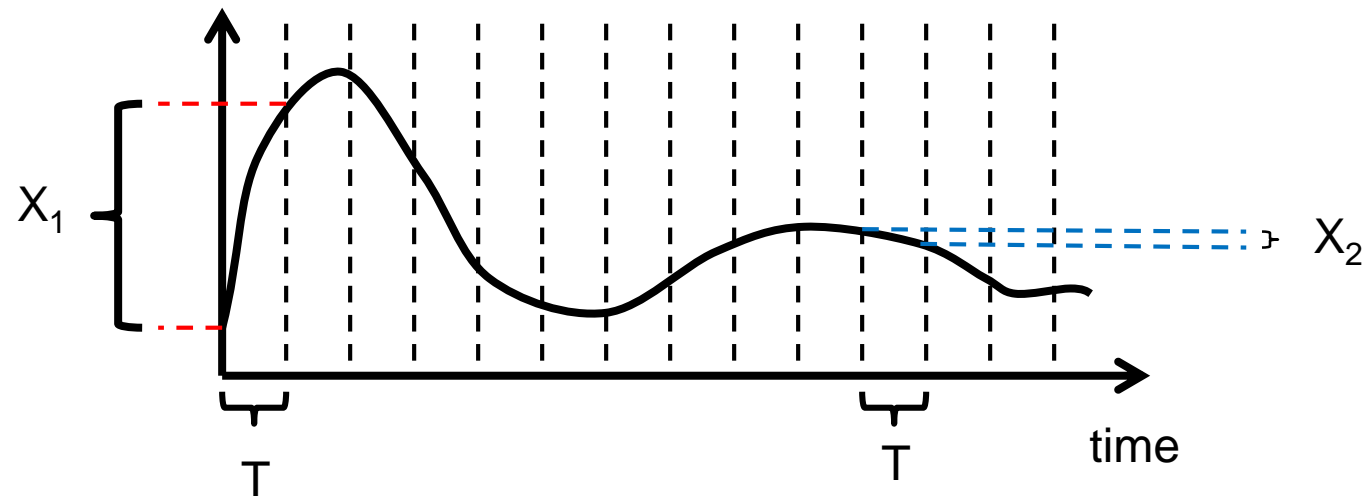
Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM19-0388



# Synchronize cyber processes with physical processes

Traditionally done by fixing a “sampling” period



Drawbacks:

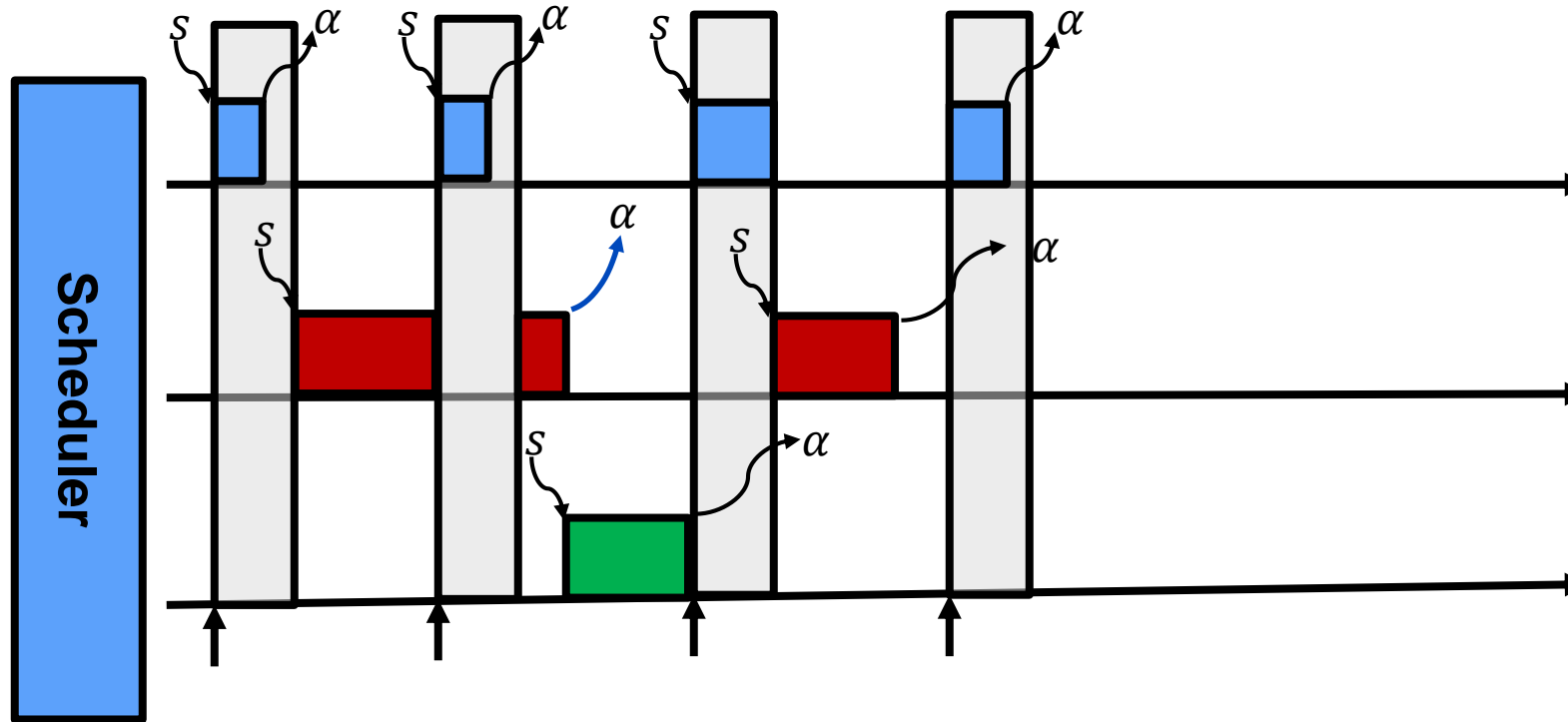
Variation in the evolution of physical process can be large ( $X_1 \gg X_2$ )

Need to force minimum period

Pessimistic resource utilization for guaranteed deadlines



# Budget Enforcement

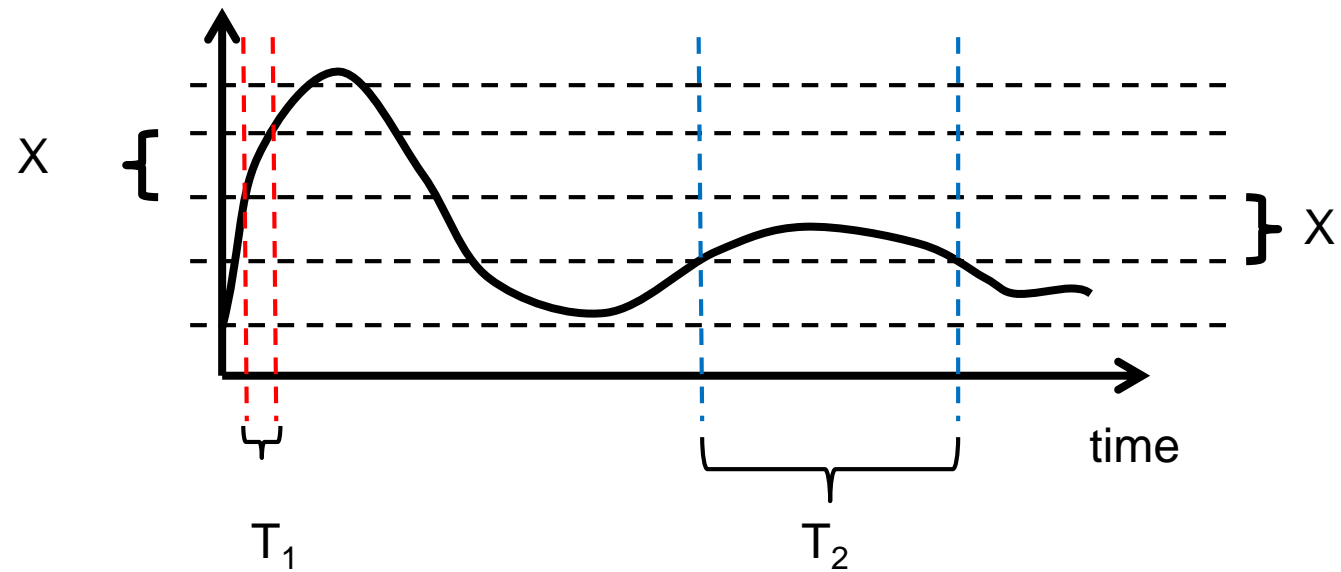


Icons credit: <http://www.doublejdesign.co.uk>



# Synchronize cyber processes with physical processes

Approach: Let the physical process drive the computation

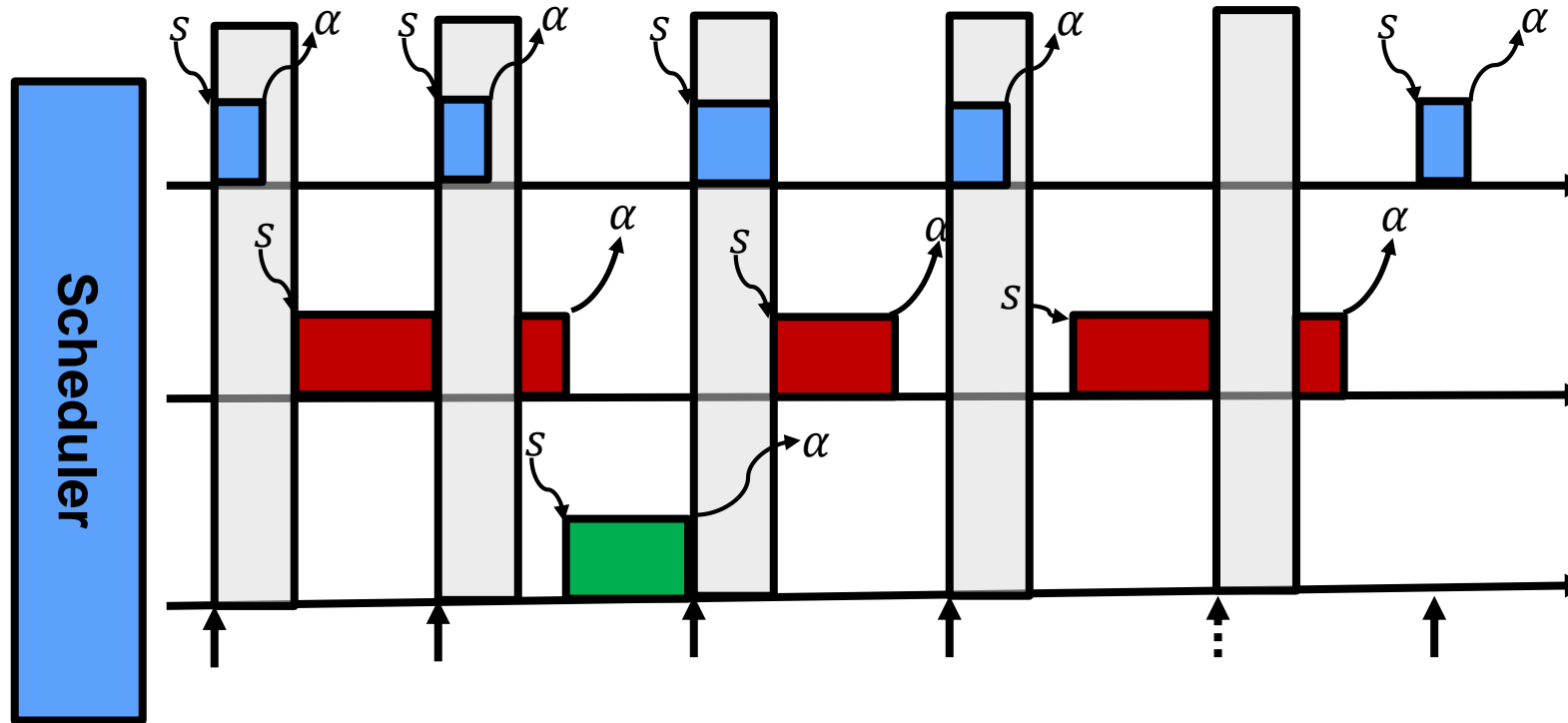


Alternative approaches:  
Event-based control  
Self-triggered control

Improved resource utilization for guaranteed deadlines



# Budget Enforcement Non-Periodic Arrivals



Icons credit: <http://www.doublejdesign.co.uk>



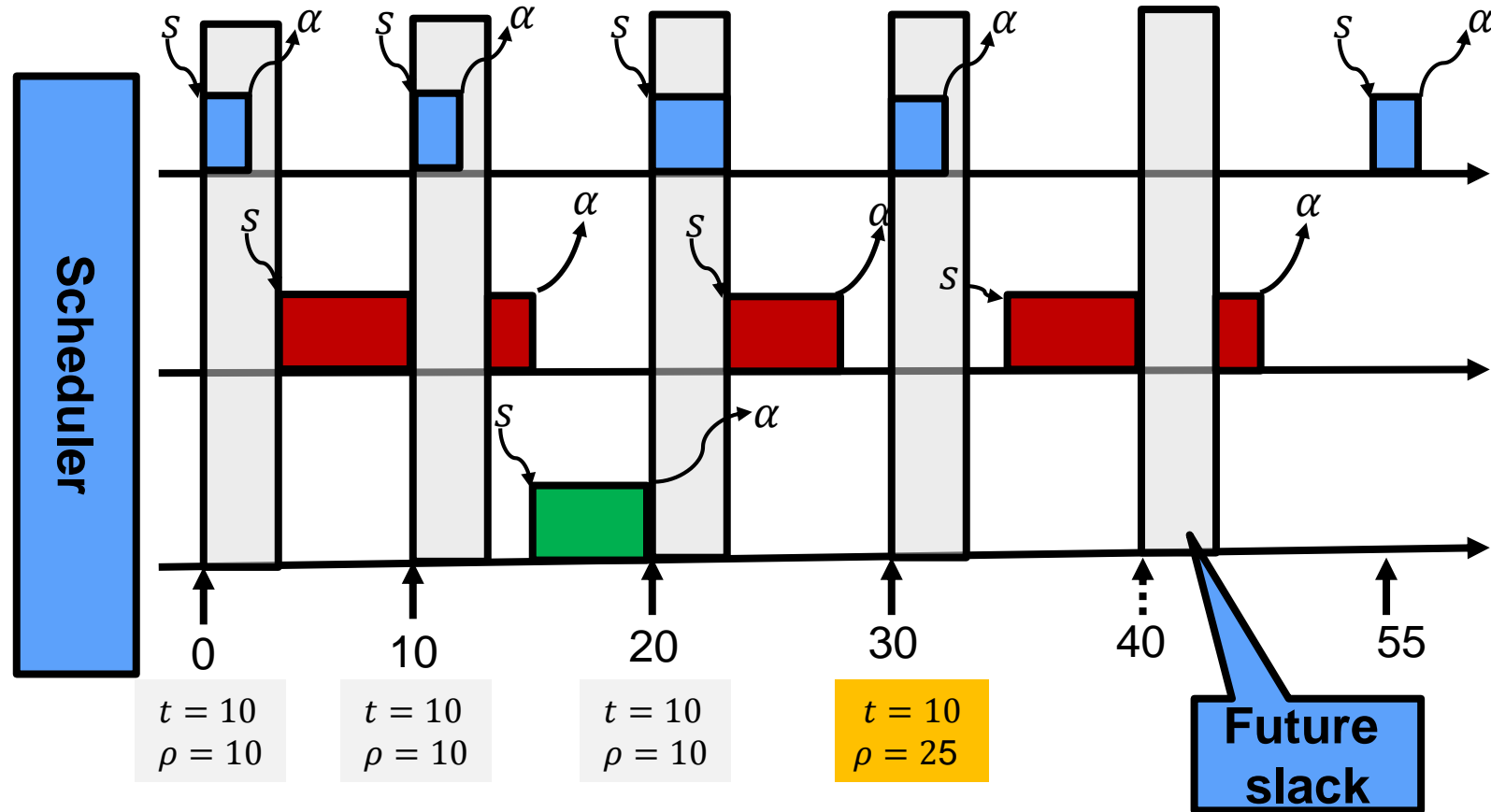
# Temporal-Physical Clocks

Infinite sequence of temporal-physical ticks:

- $\psi = (t, \rho)$ 
  - $t$  : traditional minimum inter-arrival time for a task (fixed at design time)
  - $\rho$ : physical tick that defines time to the next job arrival (driven by physical evolution)
    - Can change every tick
    - $\rho \geq t$
- At design time:
  - $t = T$  is used for schedulability :  $U = \sum_i \frac{C_i}{T_i} < bound$



# Recover Future Slack



Icons credit: <http://www.doublejdesign.co.uk>





# Distributed Physical-Temporal Clocks

Add maximum instantaneous speed of change of physical variable  $x$ .

- $\hat{x}$

Distributed Physical-Temporal Tick:

- $\delta\psi = (t, \rho, \hat{x})$

Bounding maximum physical disagreement

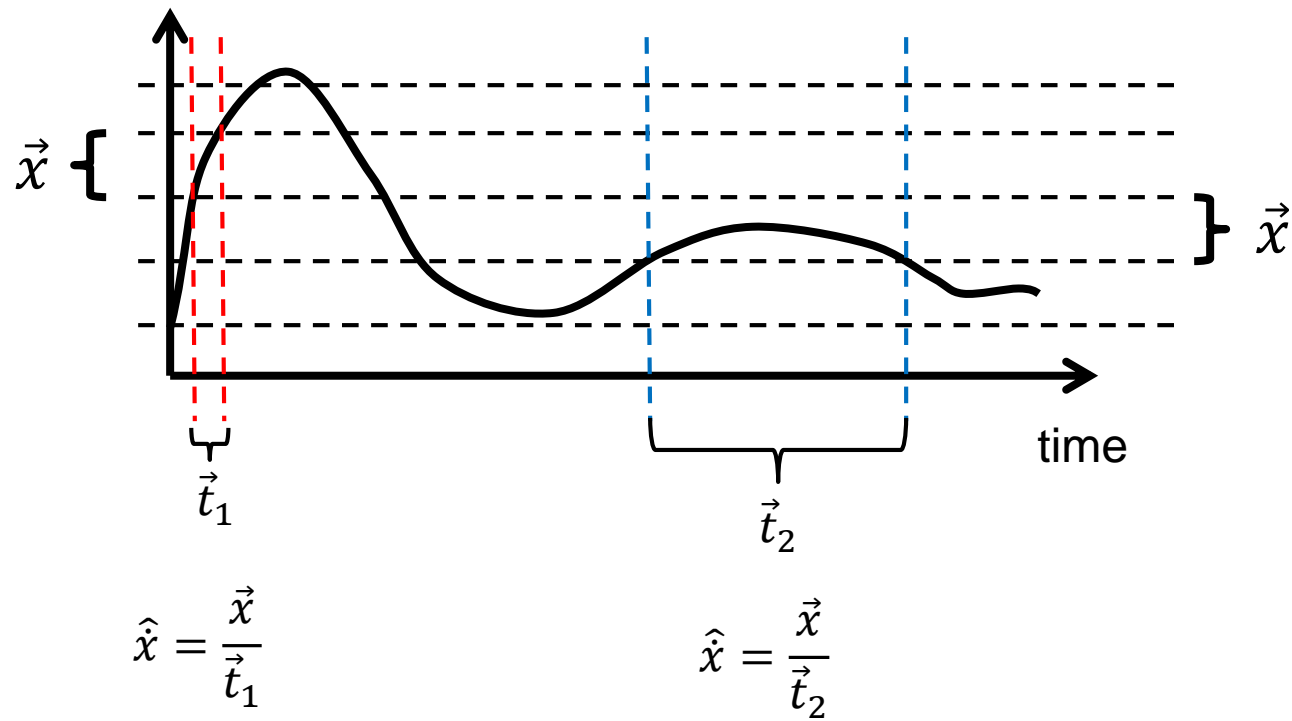
- Given maximum disagreement bound:  $\vec{x}$
- Calculate maximum out of sync interval:

$$- \vec{t} = \frac{\vec{x}}{\hat{x}}$$

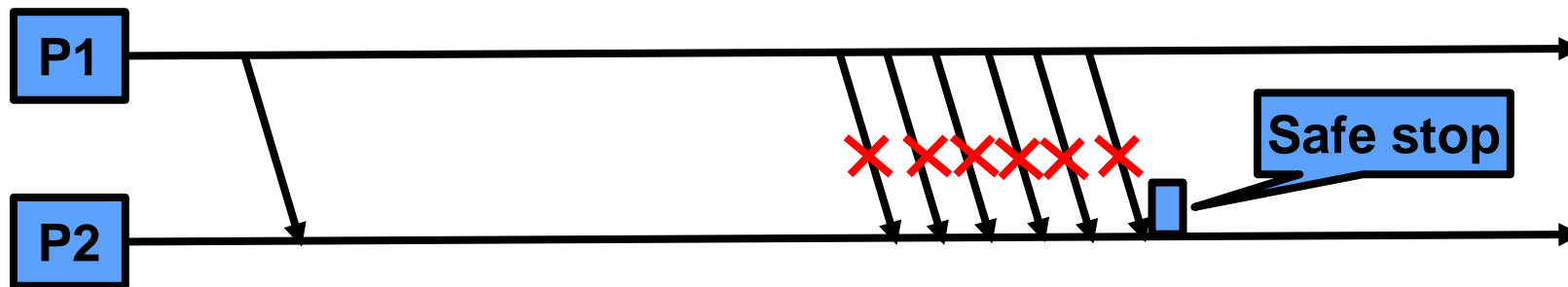
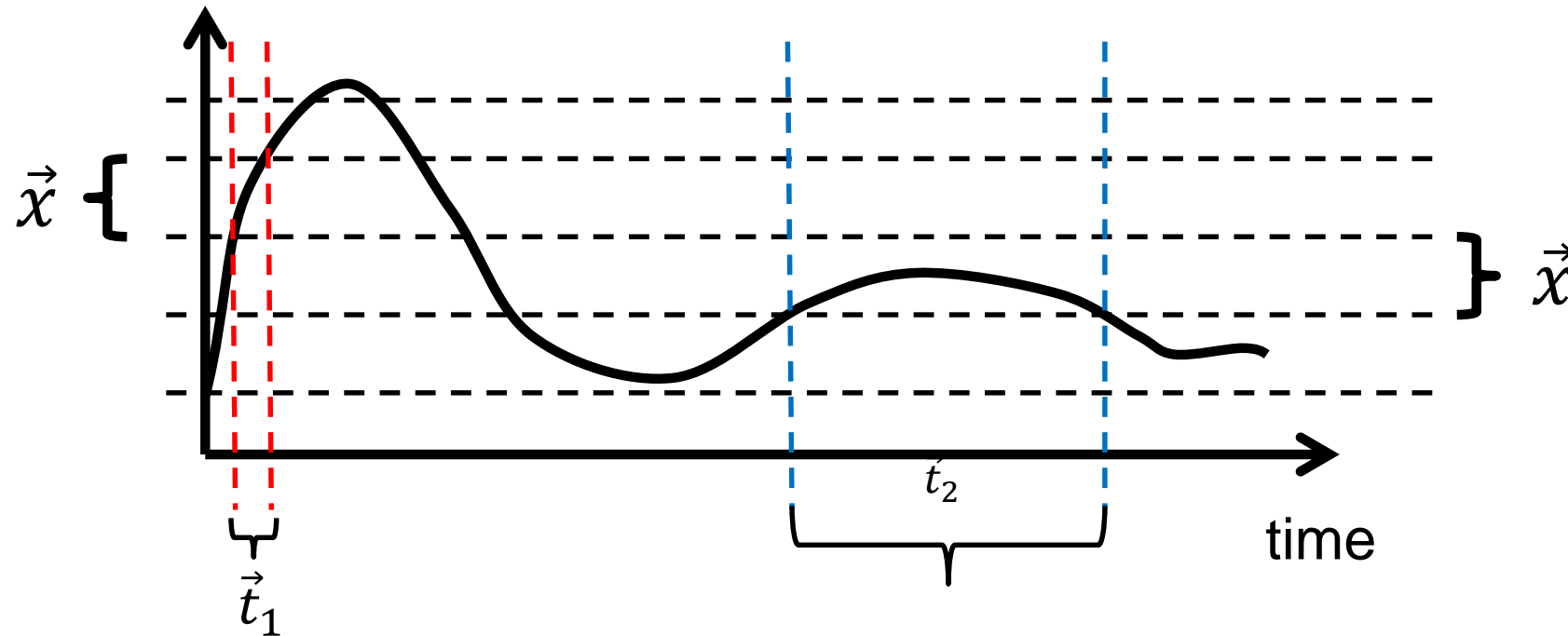
Use  $\vec{t}$  to adaptively timeout and take corrective actions



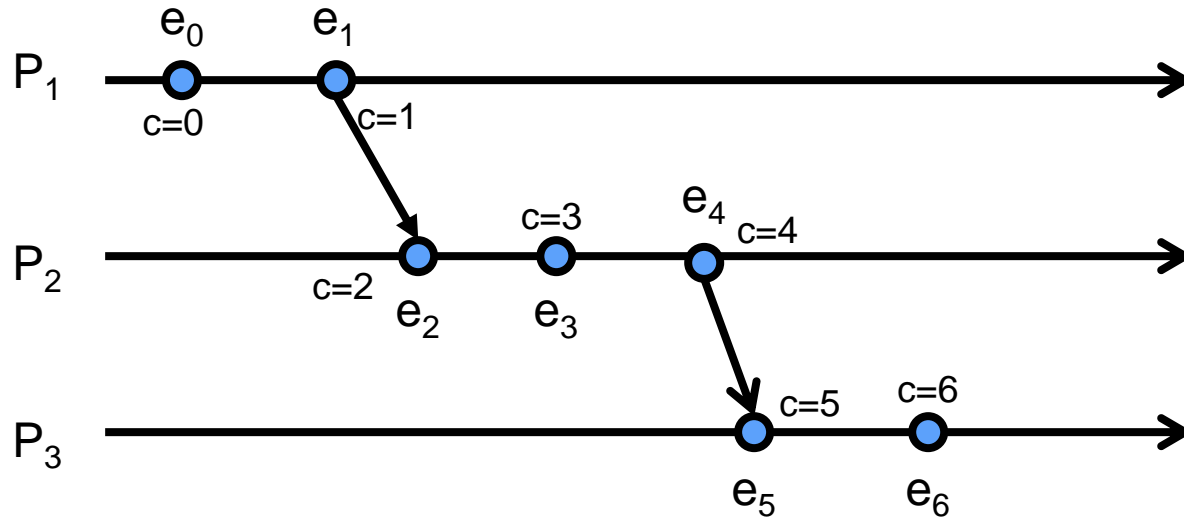
# Physical Tick



# Dynamic Timeout and Safe Stop



# Traditional Virtual Clocks

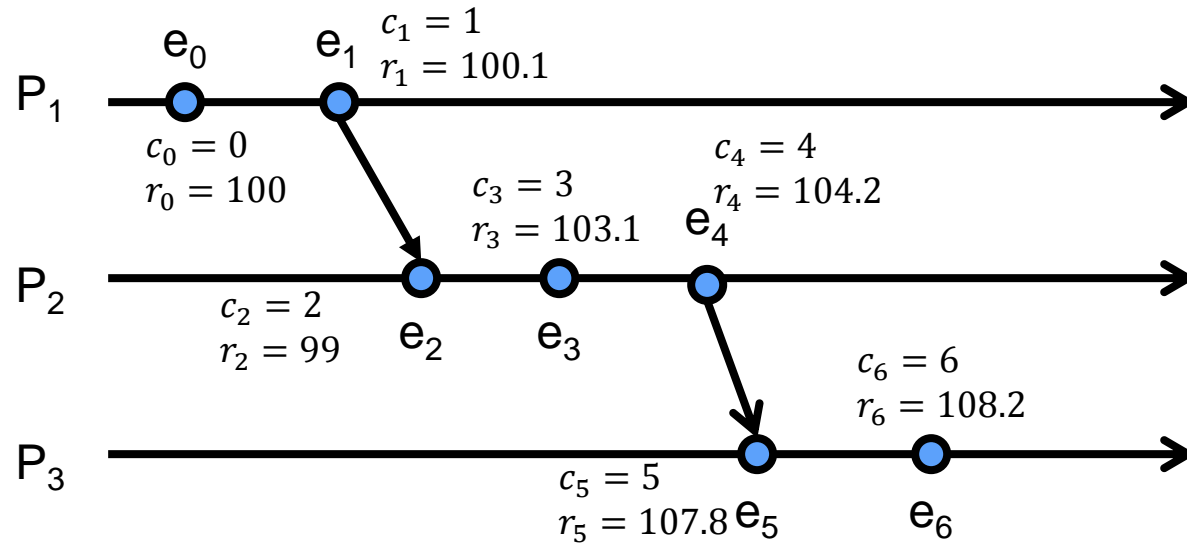


$e_i$  occurred before  $e_j$  if clock at  $e_i$  smaller than at  $e_j$

**BUT: No physical time relationship**



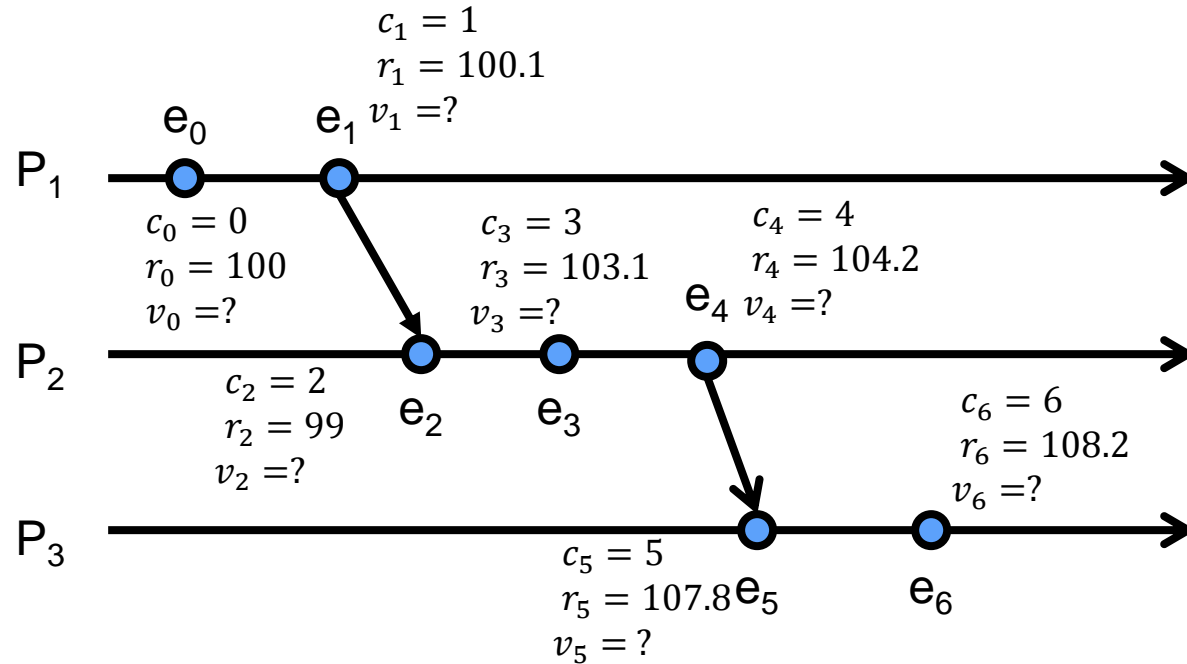
# Incorporating Physical Time



$e_i$  occurred before  $e_j$  if clock at  $e_i$  smaller than at  $e_j$



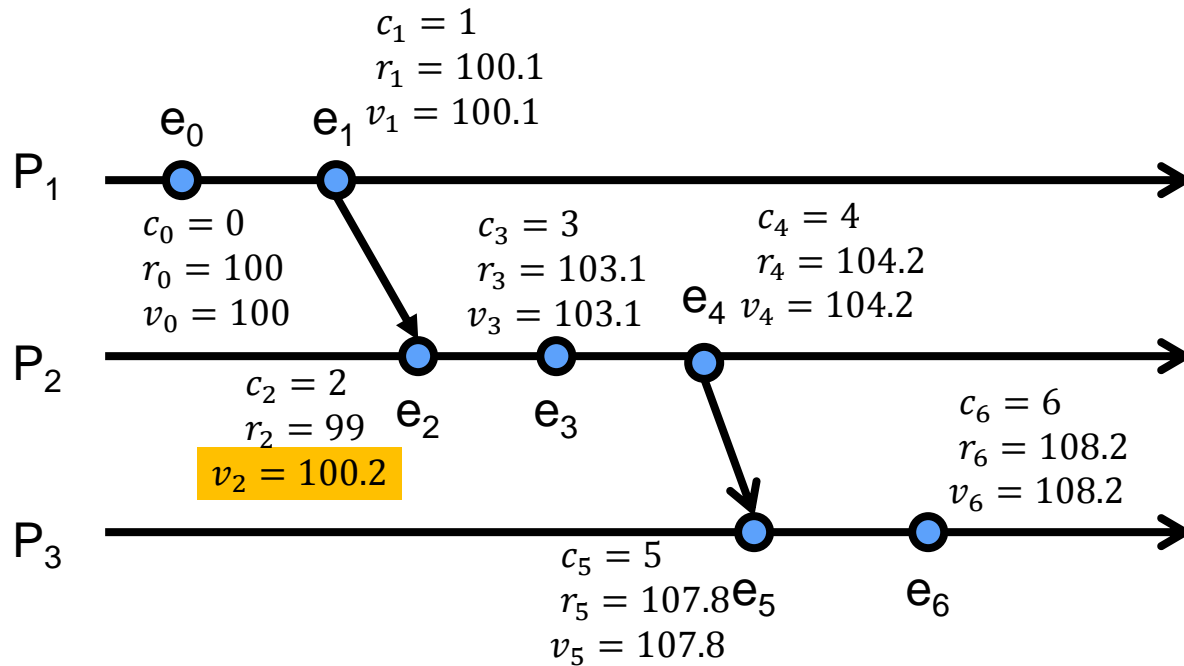
# Incorporating Physical Time



$e_i$  occurred before  $e_j$  if clock at  $e_i$  smaller than at  $e_j$



# Incorporating Physical Time



Minimize:

$$\max_i |v_i - c_i|$$

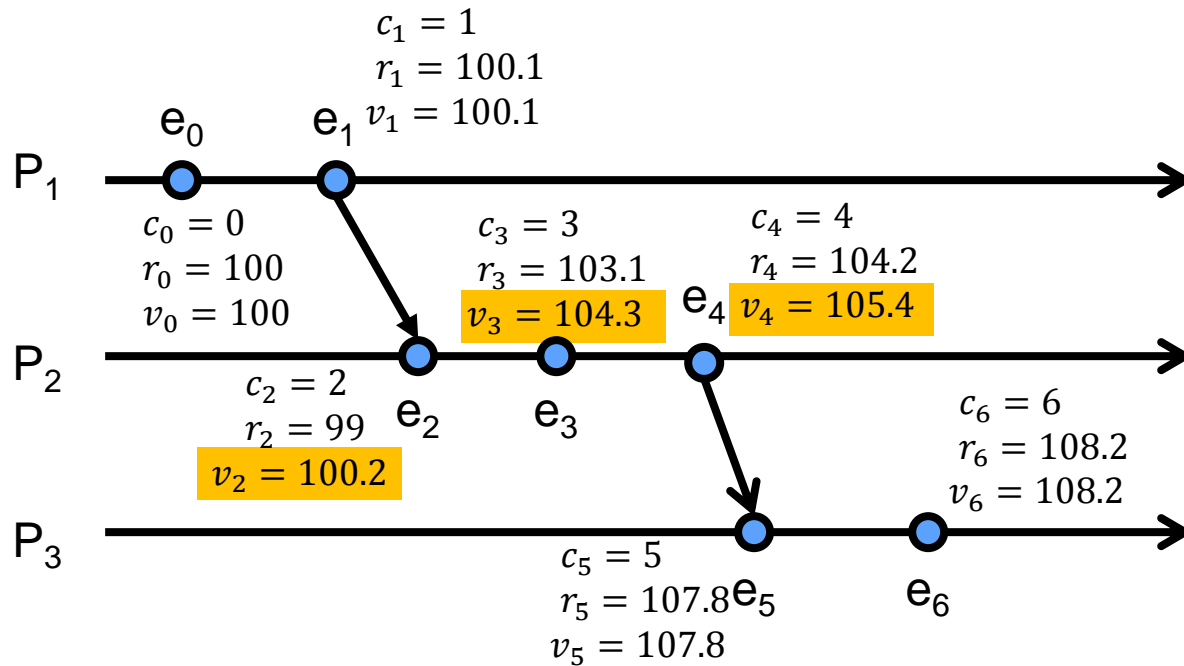
Subject to:

respecting event ordering

$e_i$  occurred before  $e_j$  if clock at  $e_i$  smaller than at  $e_j$



# Incorporating Physical Time



Minimize:

$$\max_{p \in \{1..m\}} \max_{e_i, e_j \in p} |(v_i - v_j) - (c_i - c_j)|$$

Subject to:

respecting event ordering

$e_i$  occurred before  $e_j$  if clock at  $e_i$  smaller than at  $e_j$





# Concluding Remarks

## Resource Optimization

- Exploiting new non-periodic control algorithms to minimize resource consumption

## Distributed Agreement Optimization

- Exploit knowledge of physical state to minimize synchronization

## Deriving Physical Timestamps from Agreement

- Assigning timestamps to events such that timestamps can be used to find event order and the timestamps mimic physical time.

