Processing Enhancement and Virtualization for Cyber-Physical Computations

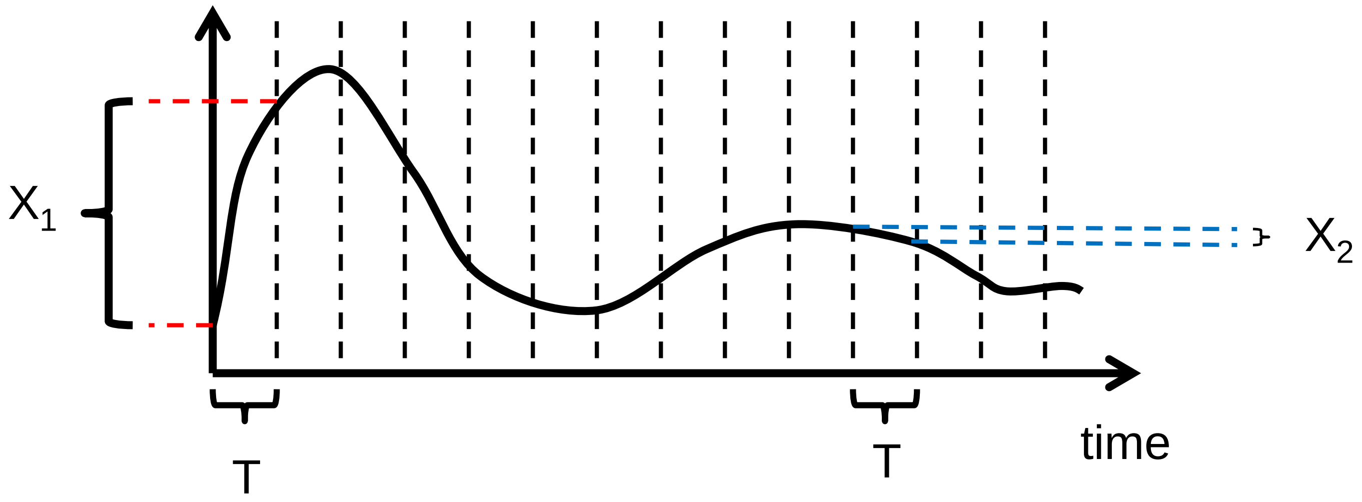
Dionisio de Niz and Bjorn Andersson

Next Generation Operating Systems for CPS Workshop

April 15th, 2019

Synchronize cyber processes with physical processes

Traditionally done by fixing a “sampling” period



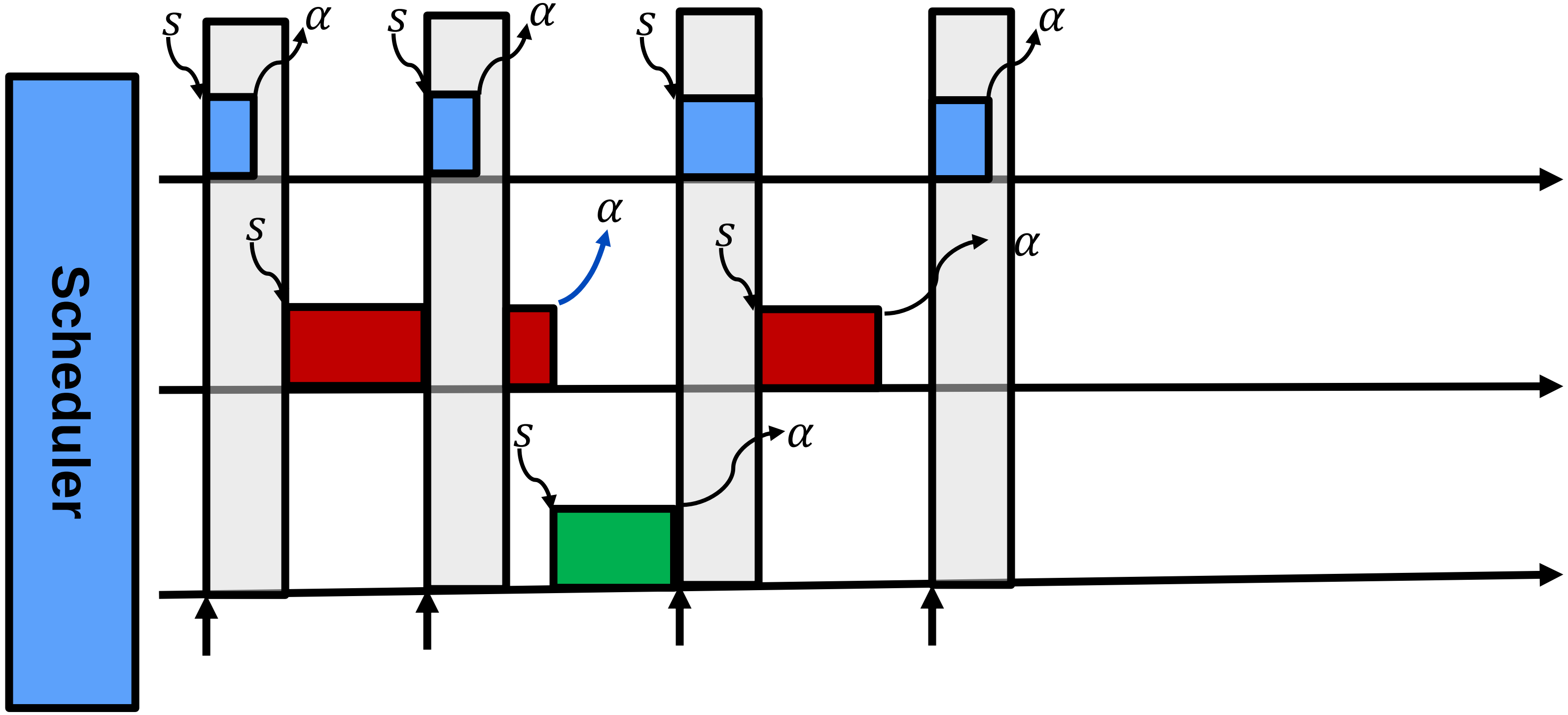
Drawbacks:

Variation in the evolution of physical process can be large (X_1>>X_2)

Need to force minimum period

Pessimistic resource utilization for guaranteed deadlines

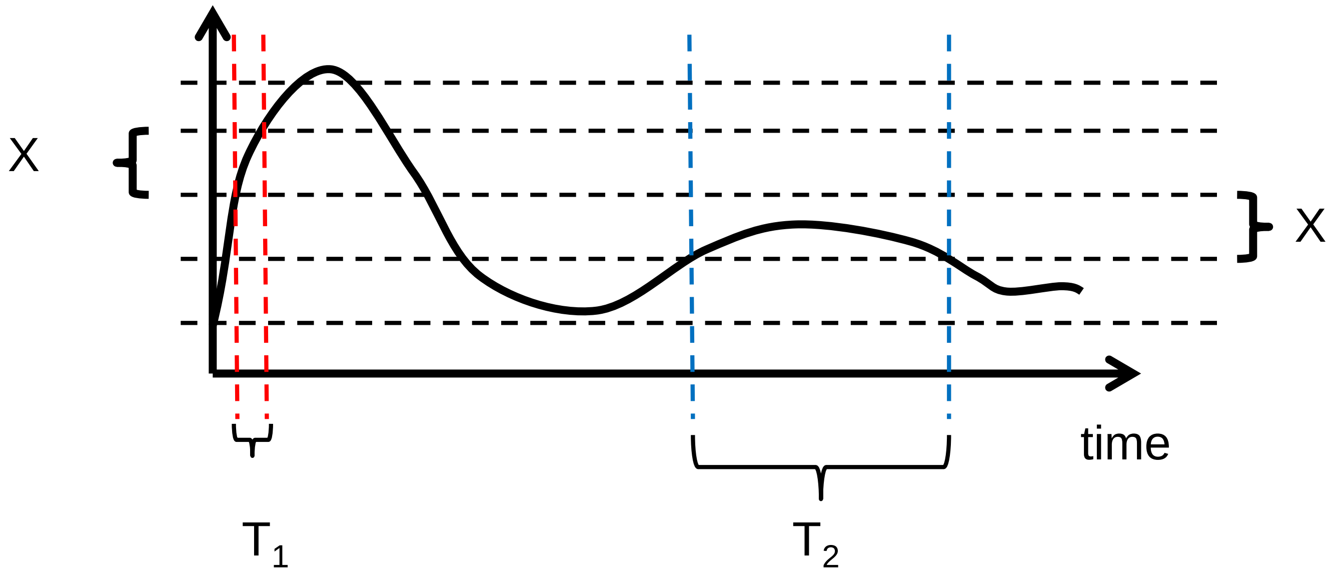
Budget Enforcement



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

Synchronize cyber processes with physical processes

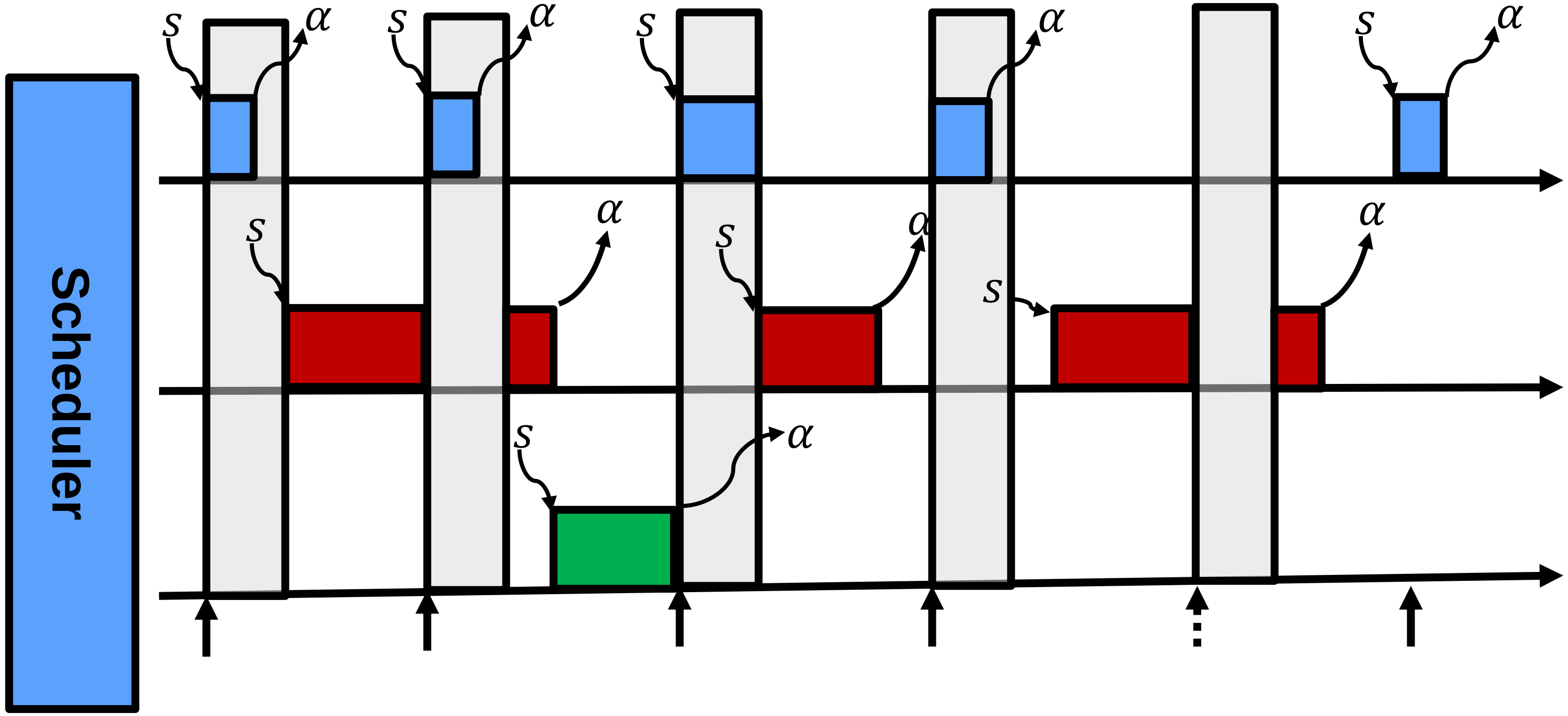
Approach: Let the physical process drive the computation



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

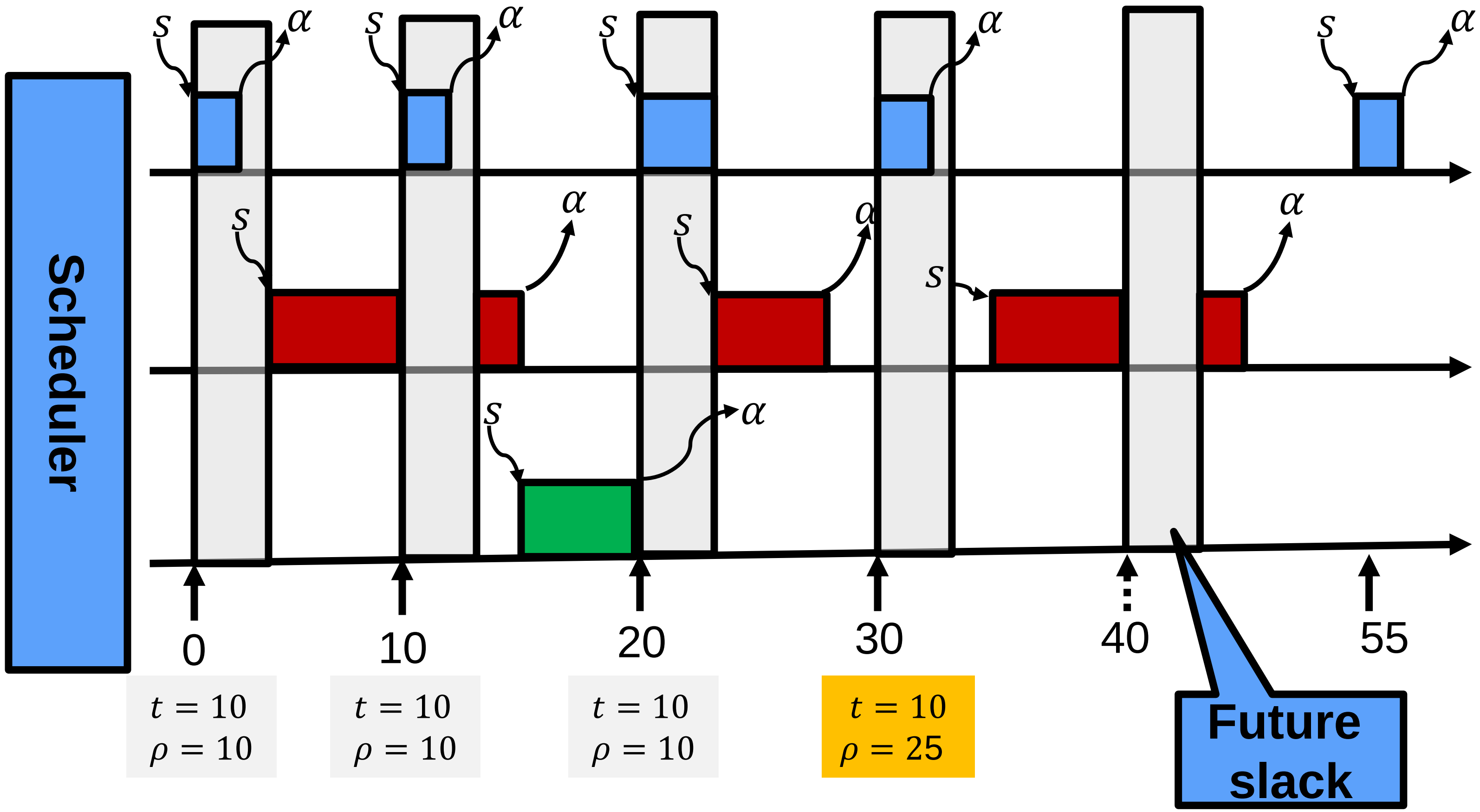
Temporal-Physical Clocks

Infinite sequence of temporal-physical ticks:

- $\psi = (t, \rho)$
  - t : traditional minimum inter-arrival time for a task (fixed at design time)
  - ρ: 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.

- $\hat{\dot{x}}$

Distributed Physical-Temporal Tick:

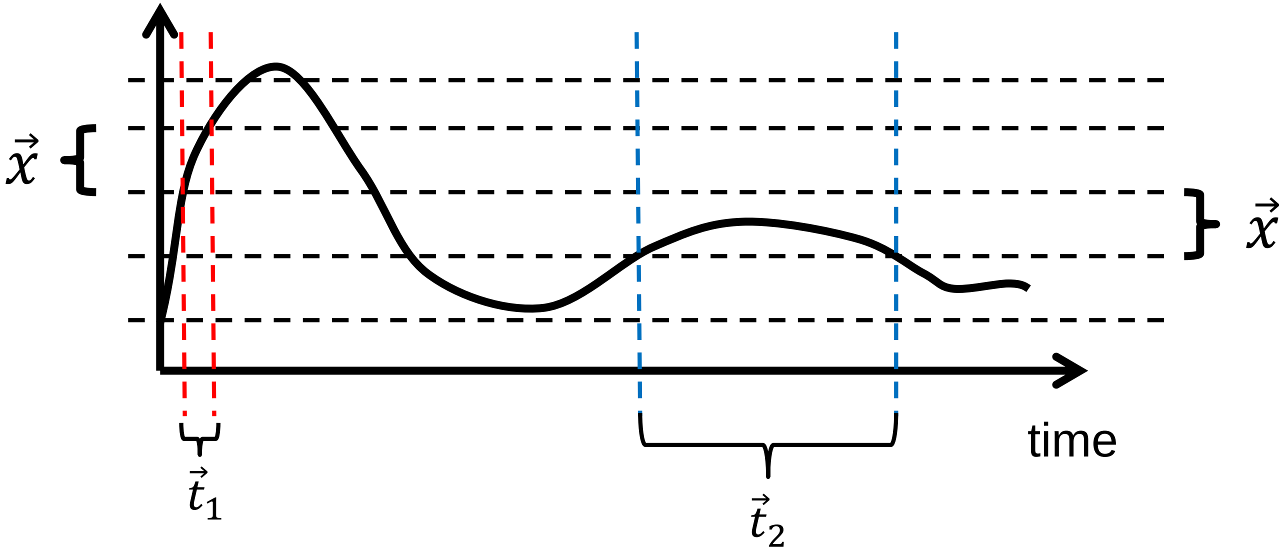
- $\delta\psi = (t, \rho, \hat{\dot{x}})$

Bounding maximum physical disagreement

- Given maximum disagreement bound: $\vec{x}$
- Calculate maximum out of sync interval:
  - $\vec{t} = \frac{\vec{x}}{\hat{\dot{x}}}$

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

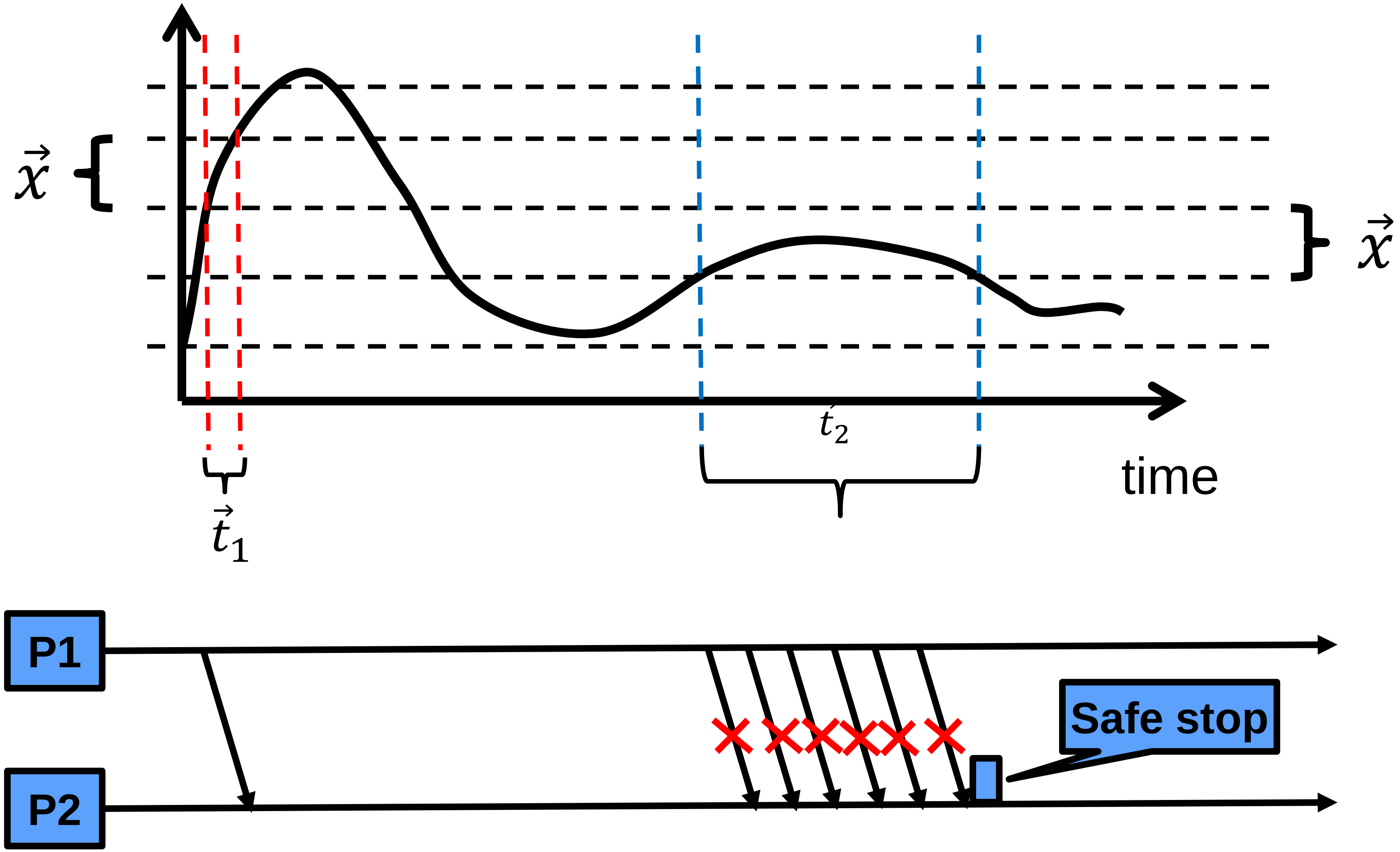
Physical Tick



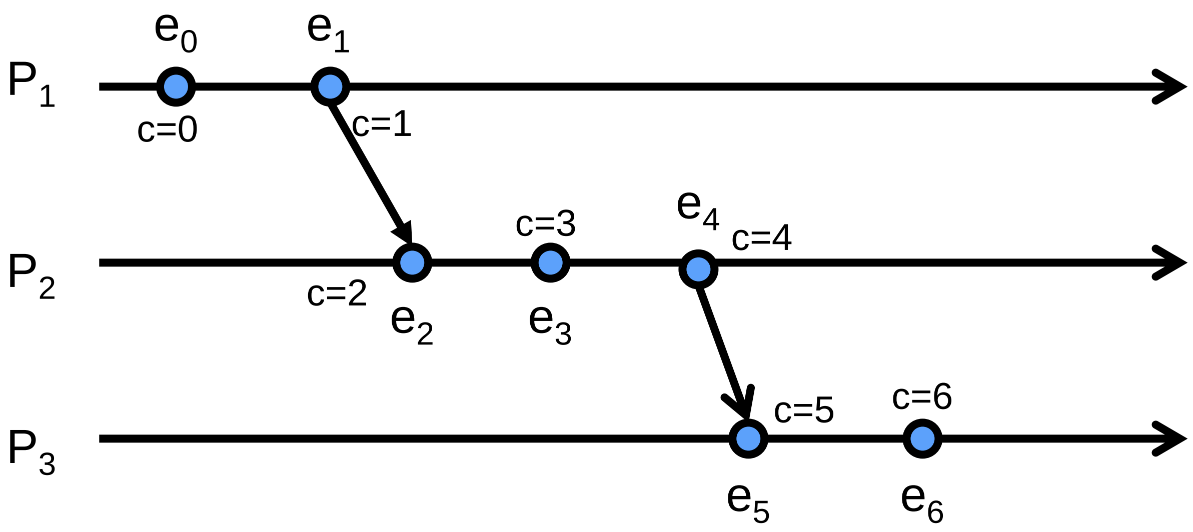
$$\hat{\dot{x}} = \frac{\vec{x}}{\vec{t}_1}$$

$$\hat{\dot{x}} = \frac{\vec{x}}{\vec{t}_2}$$

Dynamic Timeout and Safe Stop



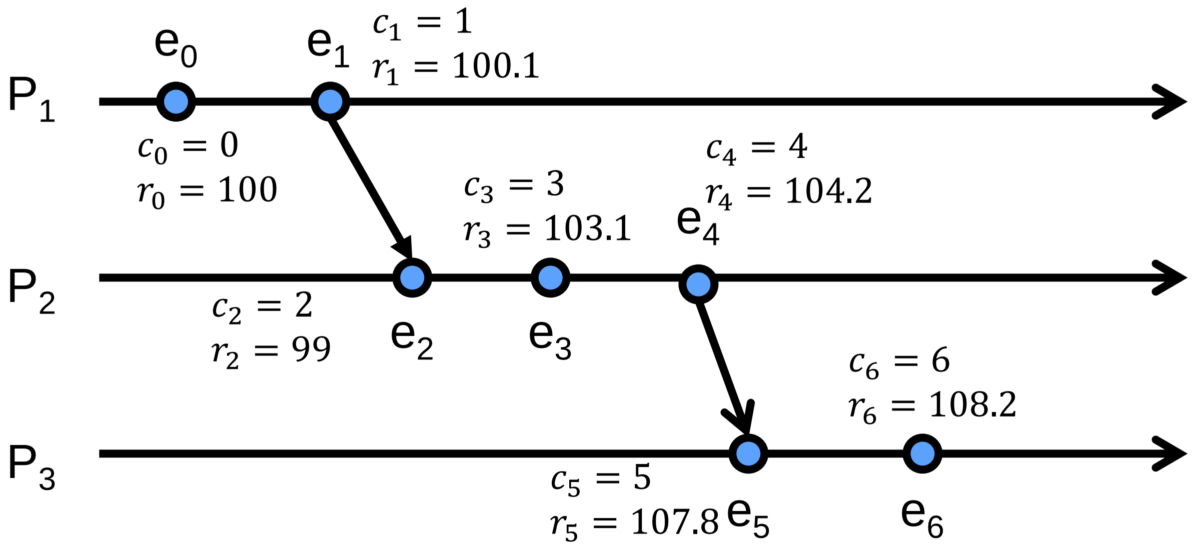
Traditional Virtual Clocks



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

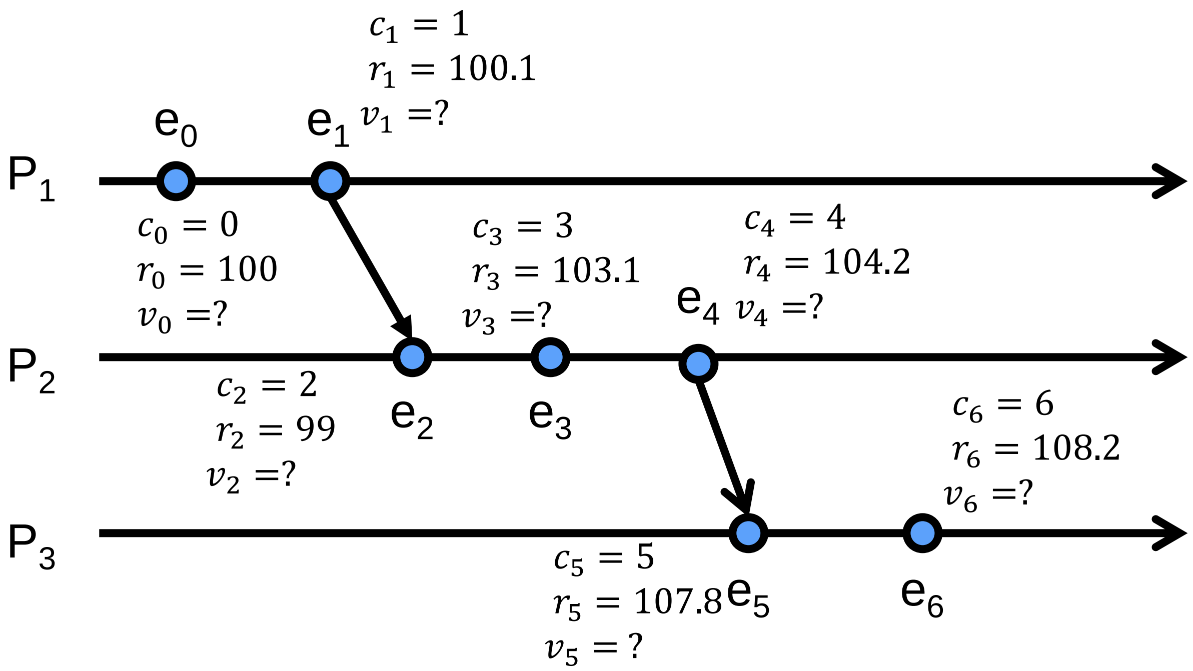
BUT: No physical time relationship

Incorporating Physical Time



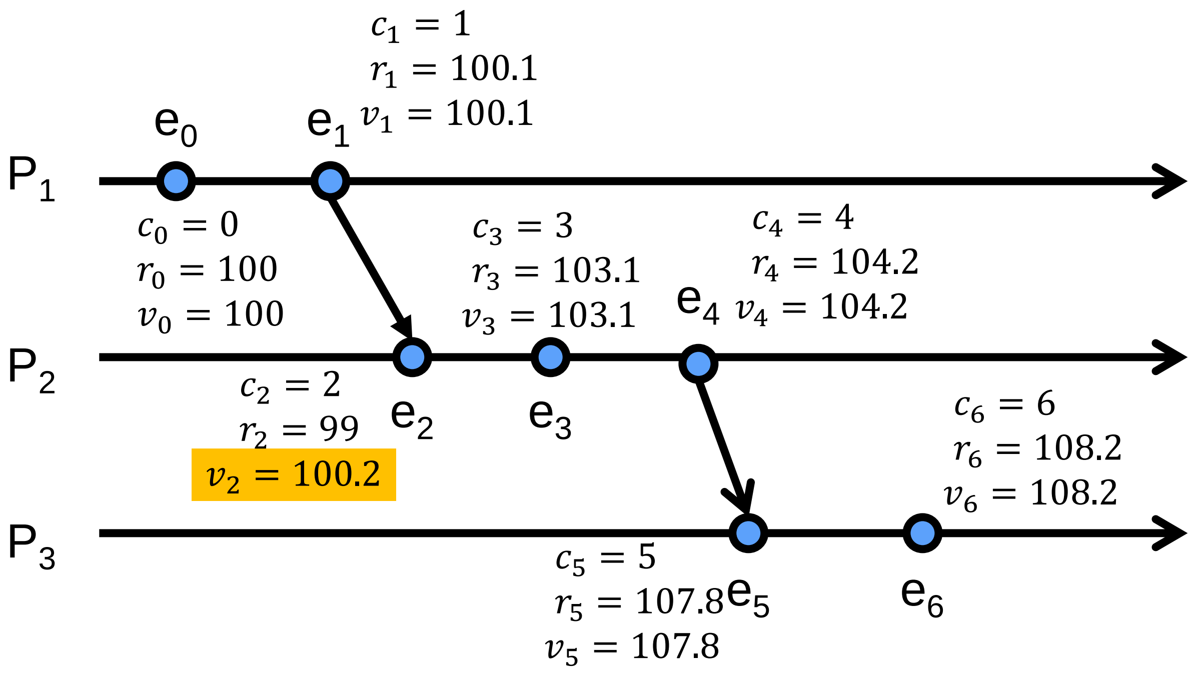
e_i occurred before e_j if clock at e_i smaller than at e_j

Incorporating Physical Time



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

Incorporating Physical Time



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

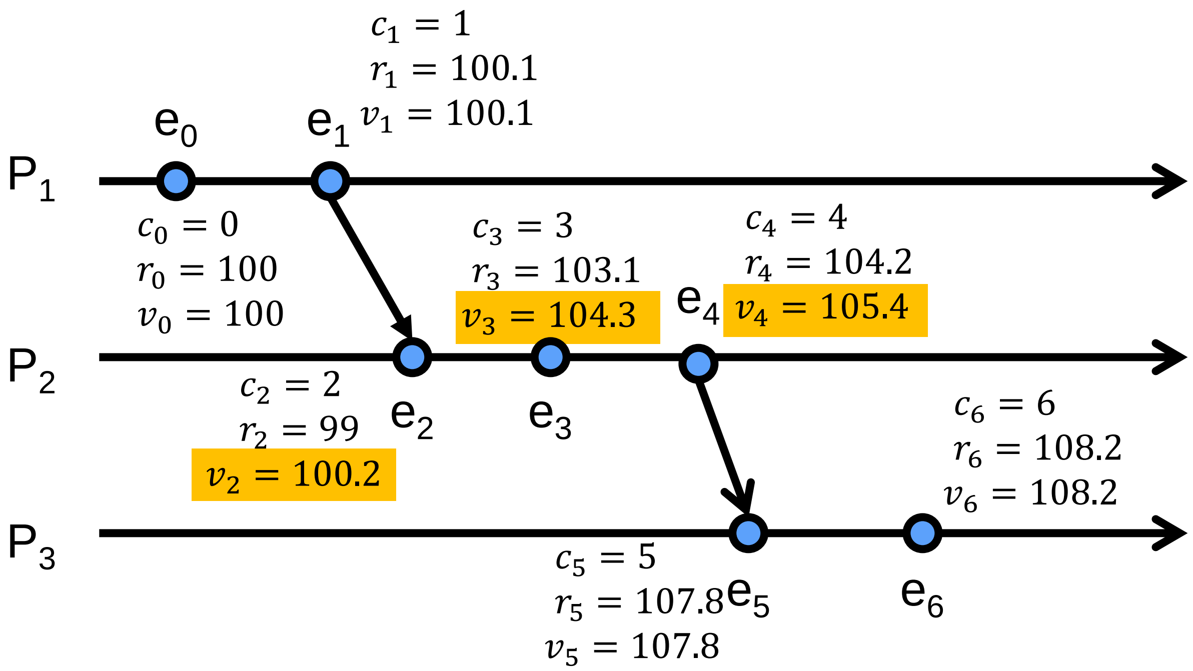
Minimize:

$$\max_i |v_i - c_i|$$

Subject to:

respecting event ordering

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.