An Efficient Fault-tolerant Scheduling Algorithm for Real-time Tasks with Precedence Constraints in Heterogeneous Systems

Xiao Qin Hong Jiang David R. Swanson

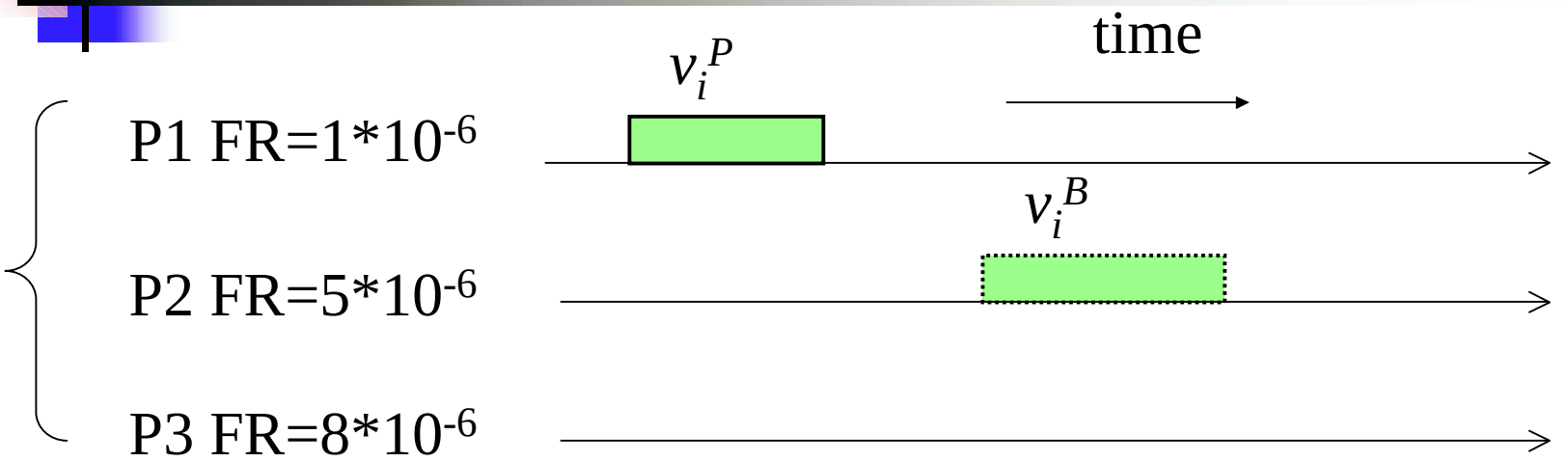
Department of Computer Science and Engineering

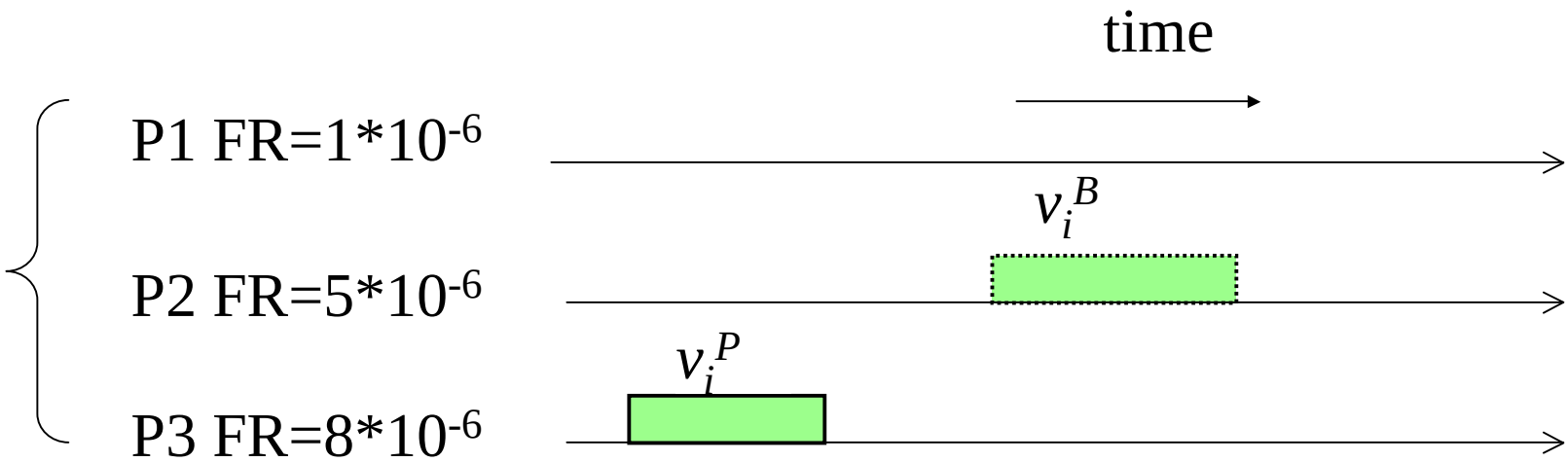
University of Nebraska-Lincoln

Overview

- Applied to embedded real-time heterogeneous systems
- Directed acyclic graph (DAG)
- Fault tolerance
- Reliability

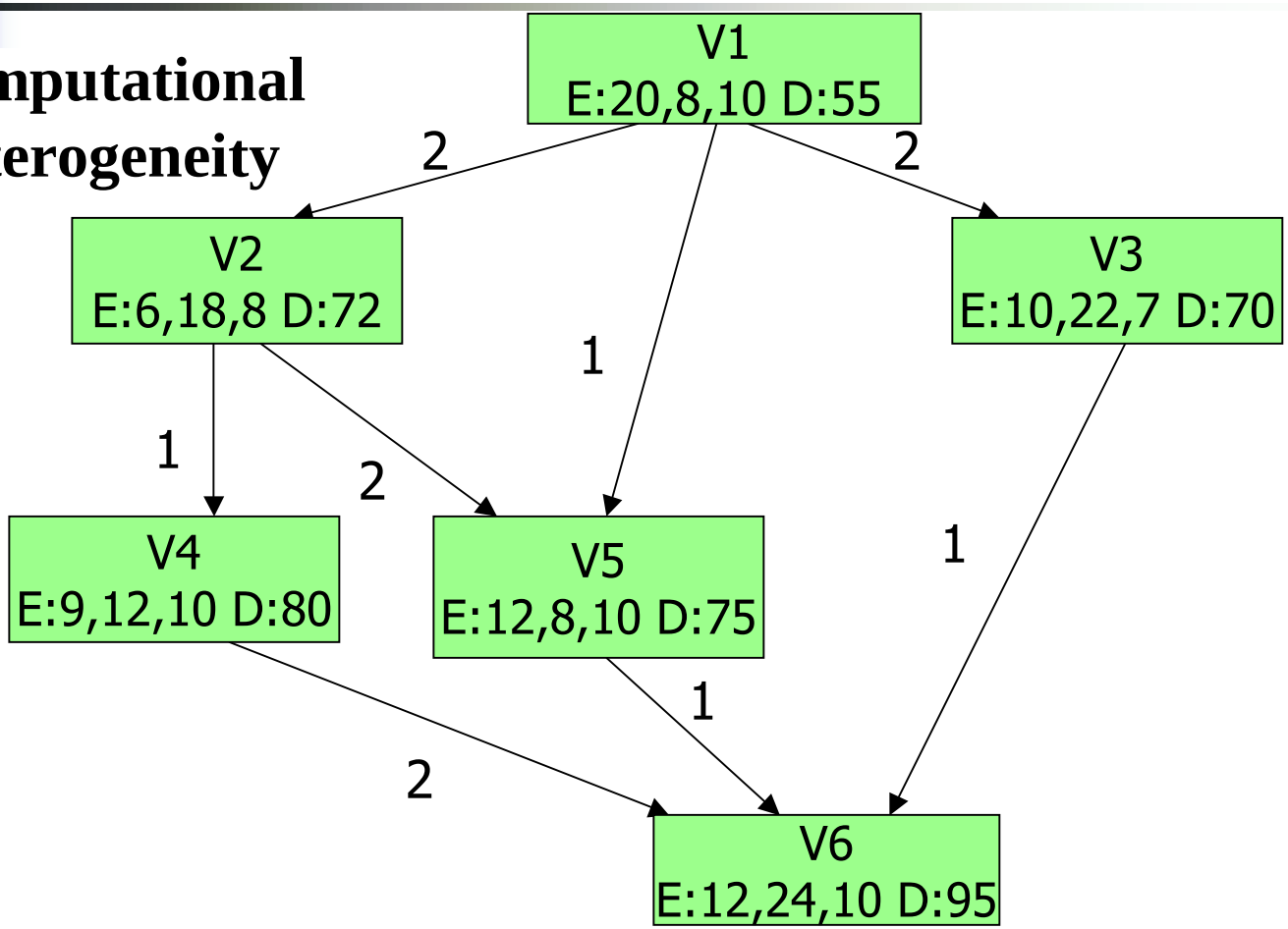
Fault Tolerance ≠ Reliability





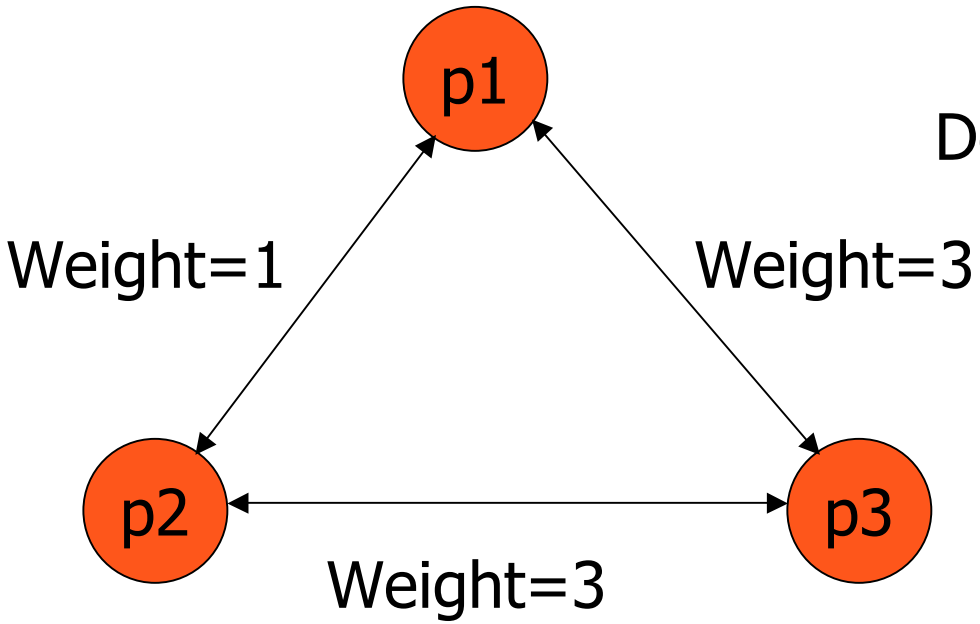
Model

computational heterogeneity

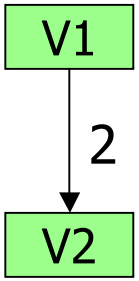


Model (cont.)

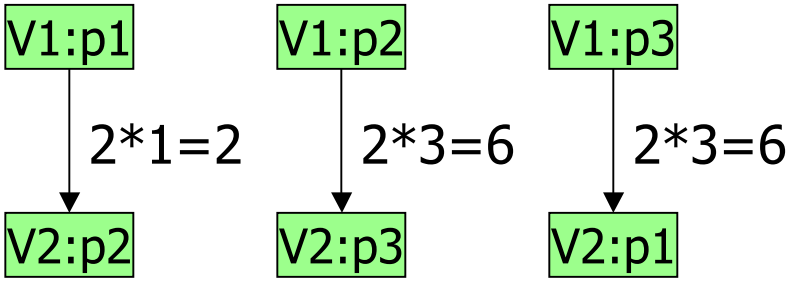
Communicational heterogeneity



Weight=1/Bandwidth

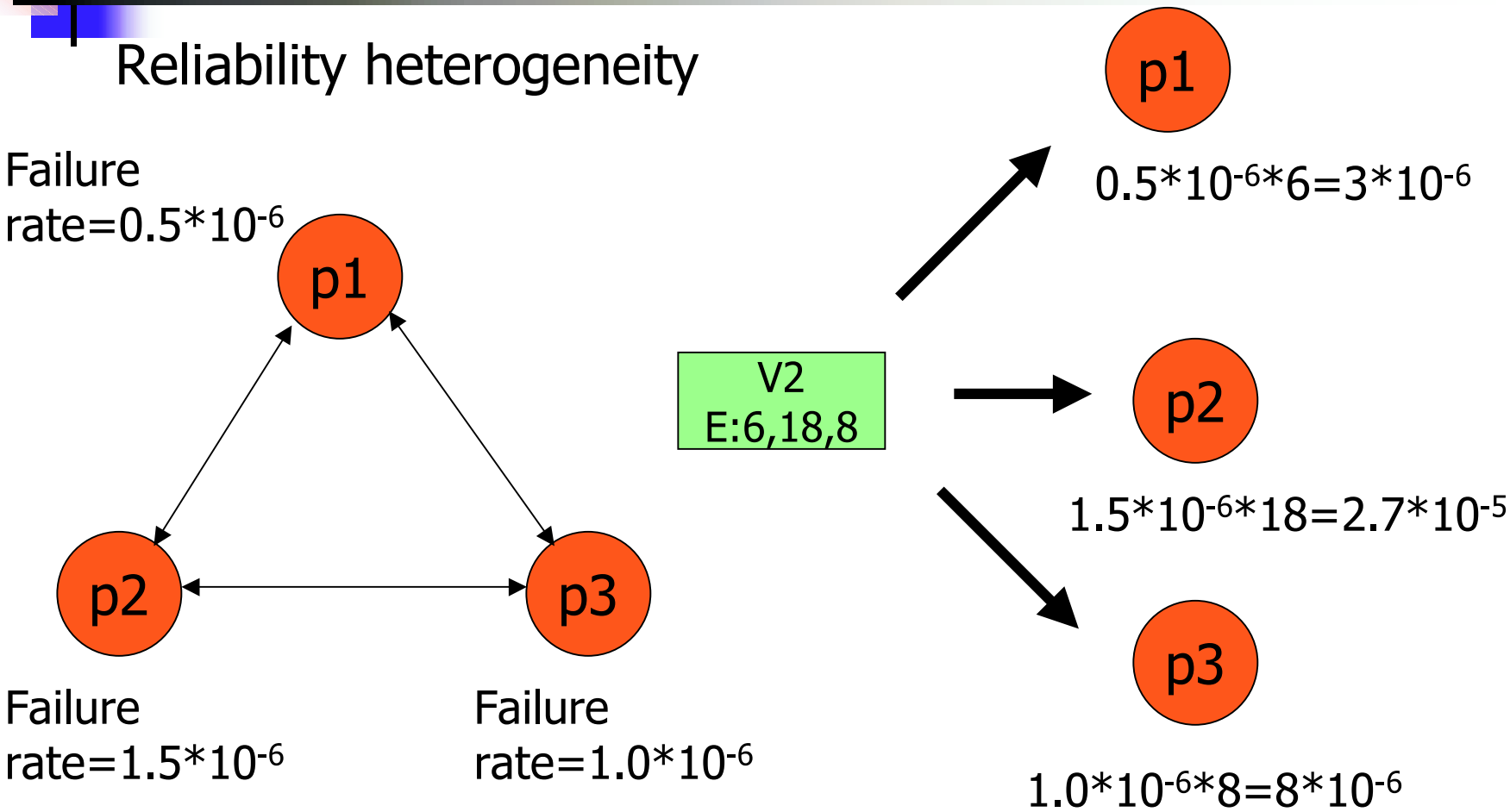


Determine communication time

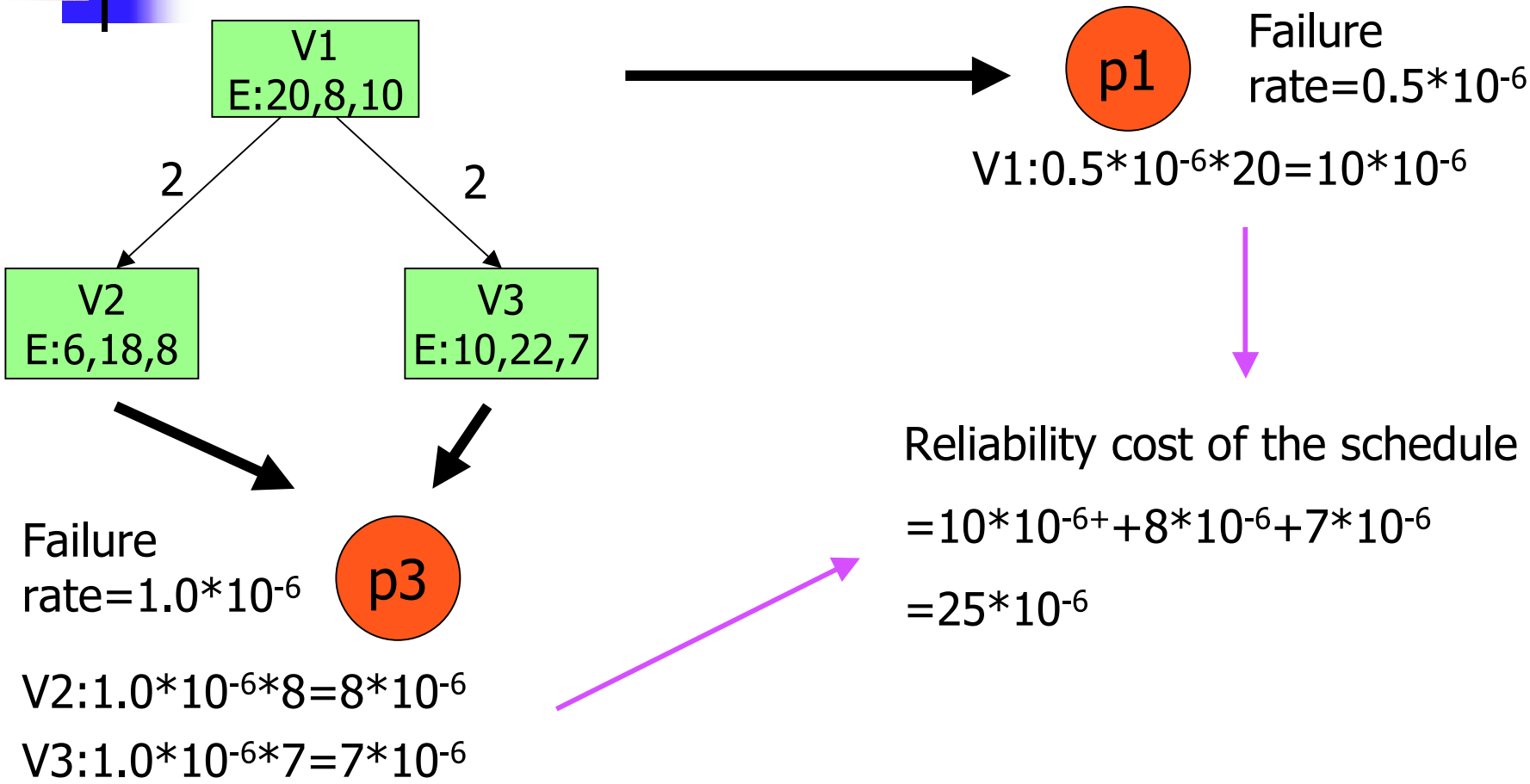


Reliability cost = Failure Rate*Execution Time

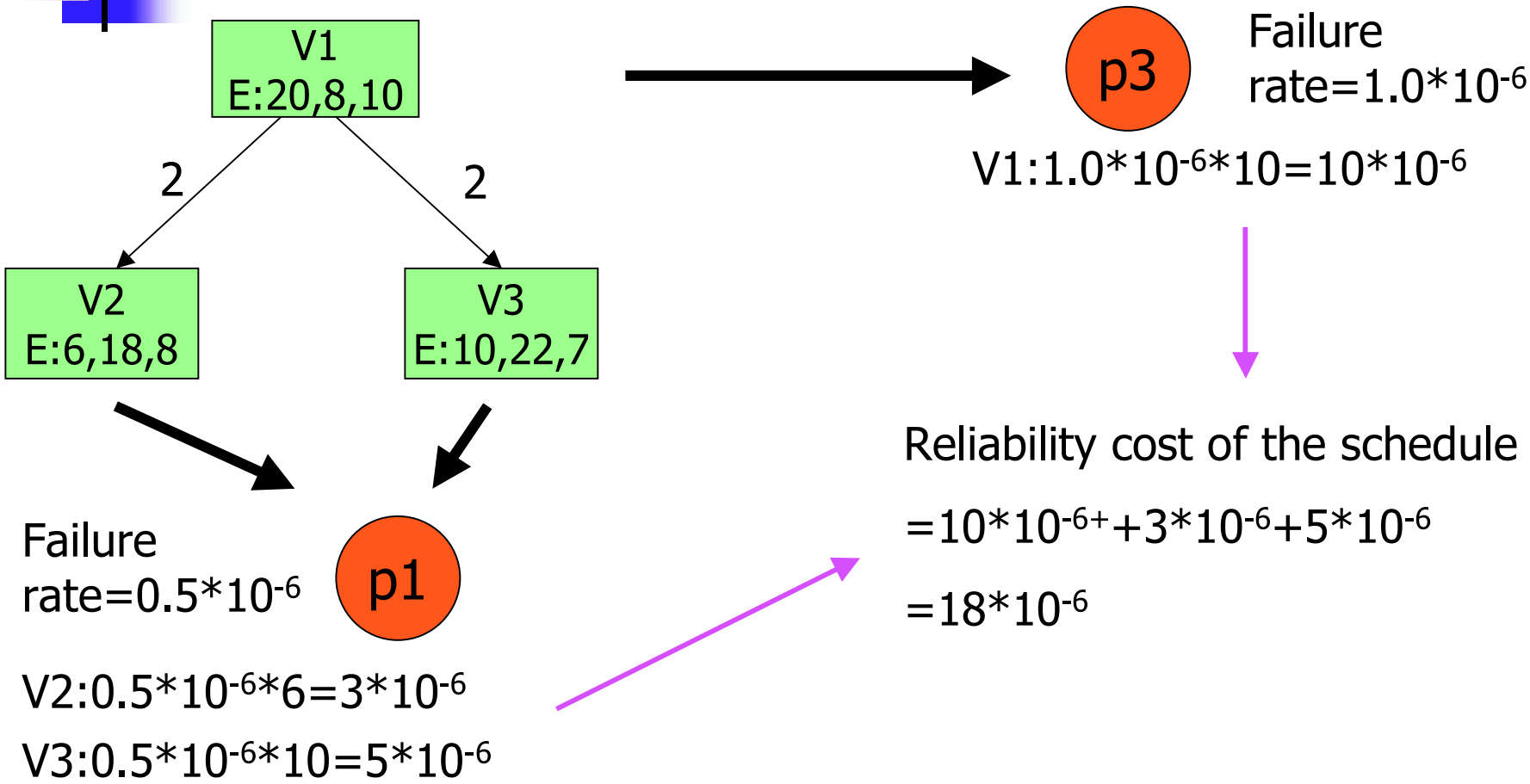
Reliability heterogeneity



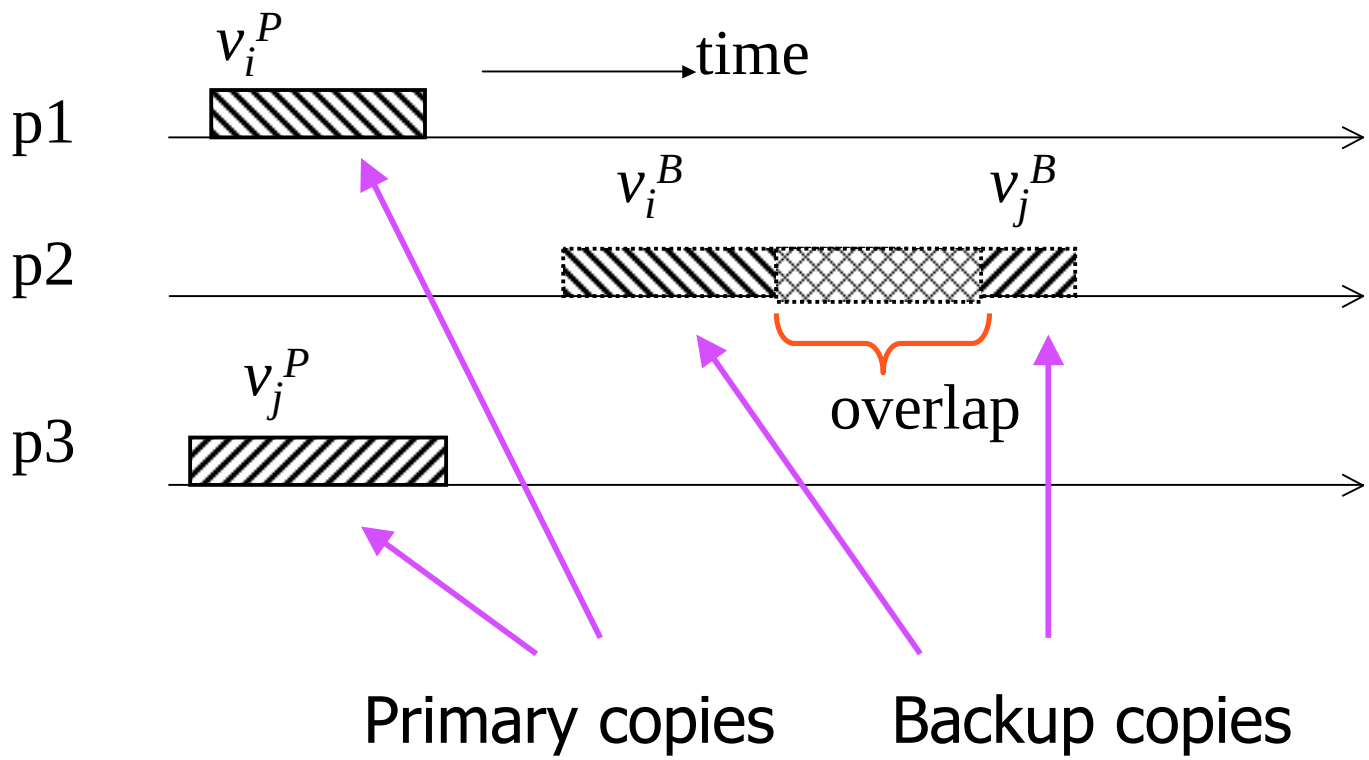
Reliability cost = Failure Rate*Execution Time (cont.)



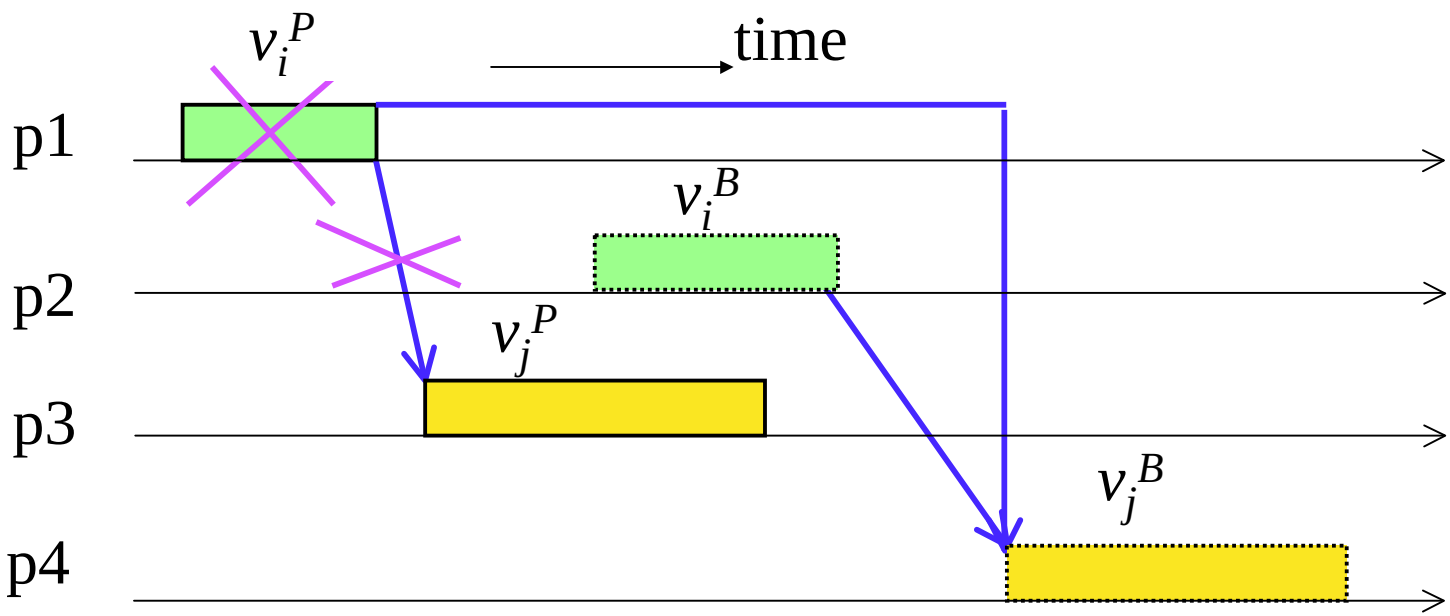
Reliability cost = Failure Rate*Execution Time (cont.)



Fault Tolerance

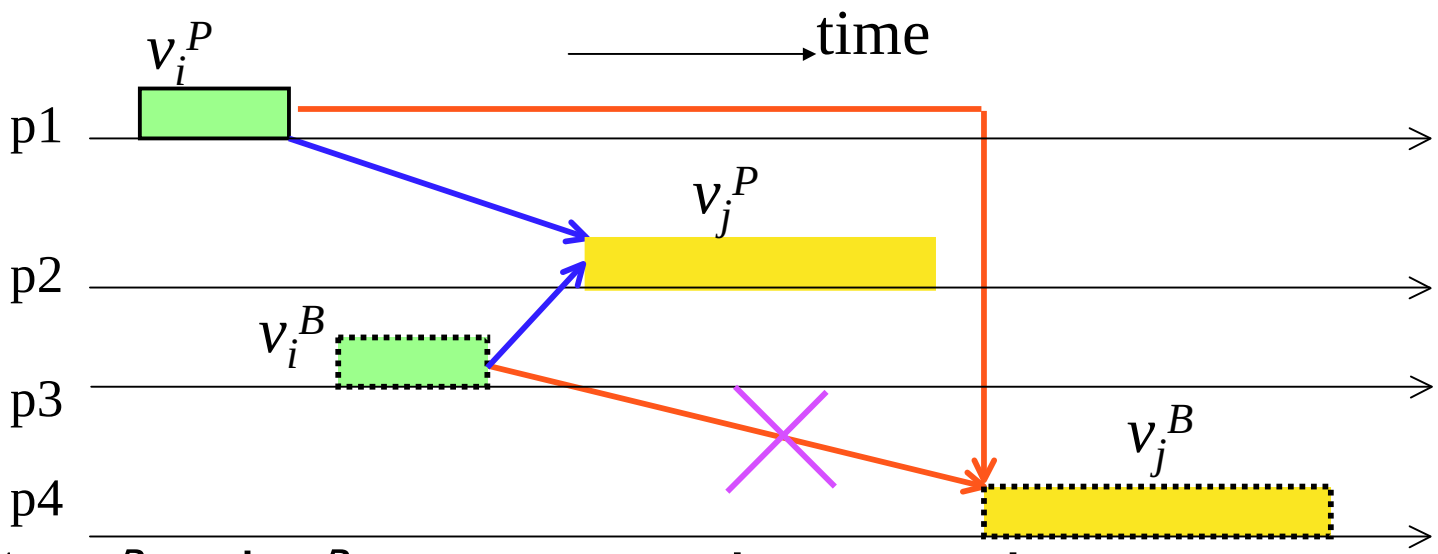


Strong Primary Copy



p_1 fails, v_i^B executes.
v_j^P cannot receive message from v_i^B, v_j^B must execute instead of v_j^P.

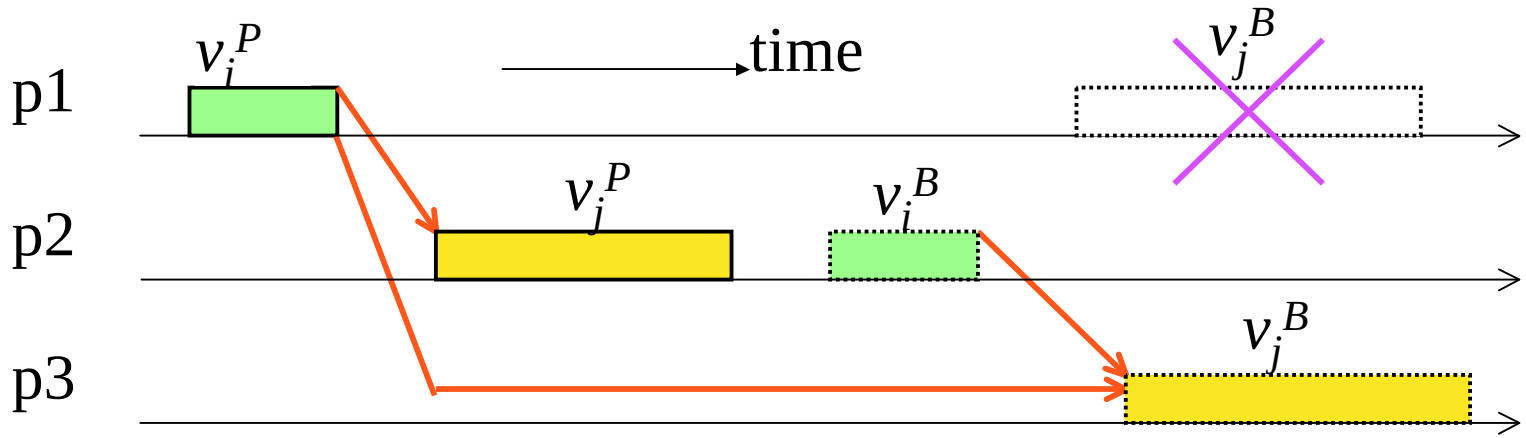
Principle 1



1. v_i^P and v_j^P are strong primary copies
2. v_i^P and v_j^P are scheduled on two different processors.

→ v_i^B is not execution-preceding v_j^B.

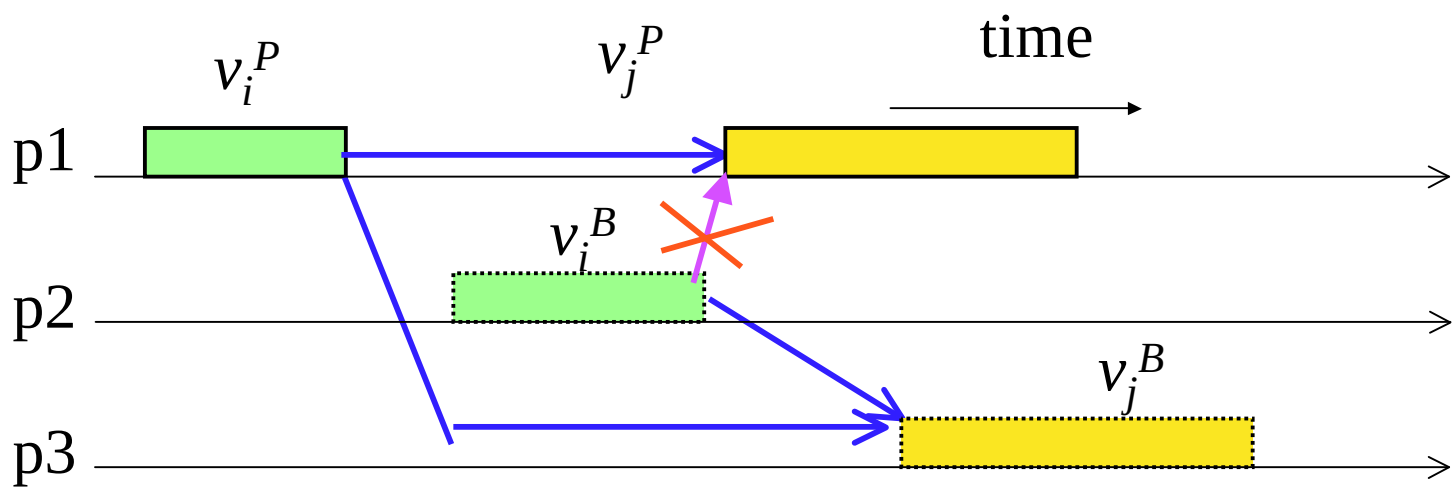
Principles 2



1. v_i^B,is not schedule-preceding v_j^P
2. v_i^P is a strong primary copy.

→ v_j^B can not be scheduled on the processor on which v_i^P is scheduled.

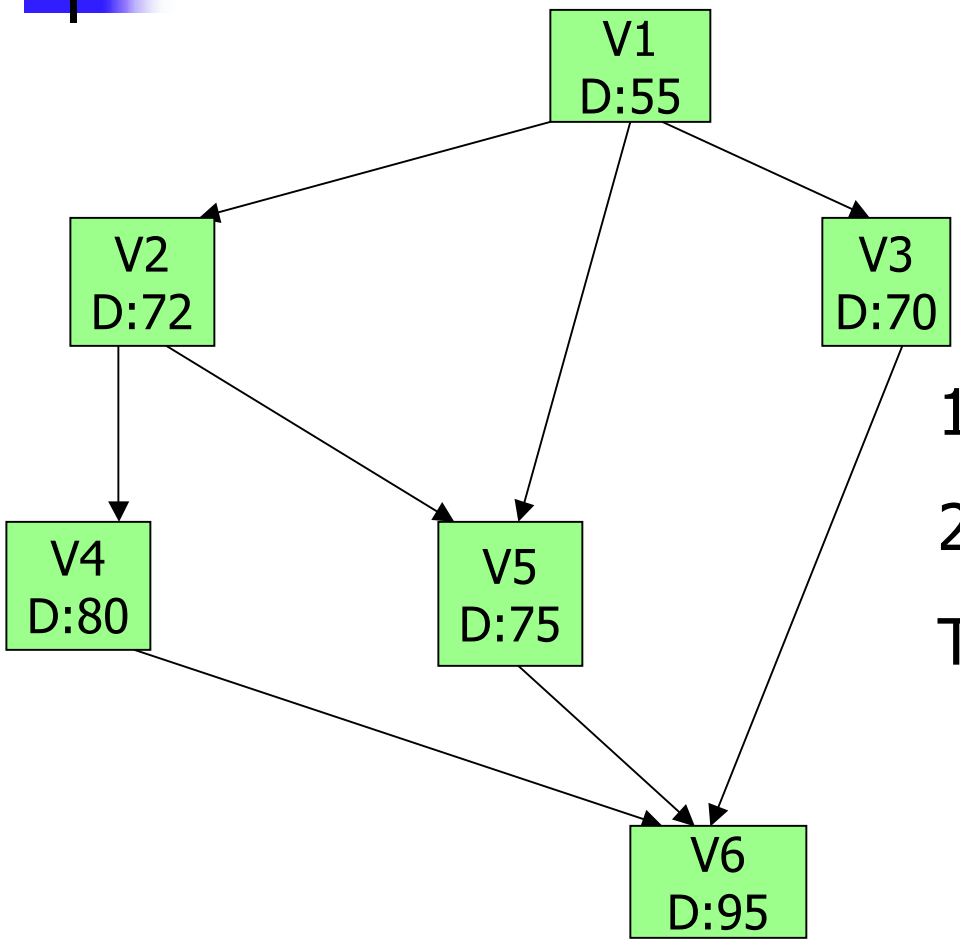
Principles 3



1. v_i is the predecessor of v_j,
2. v_i^P and v_j^P are scheduled on the same processor
3. v_i^P is the strong primary copy.

→ In this case, v_i^B is not execution-preceding v_j^P.

Step 1: Sort Task

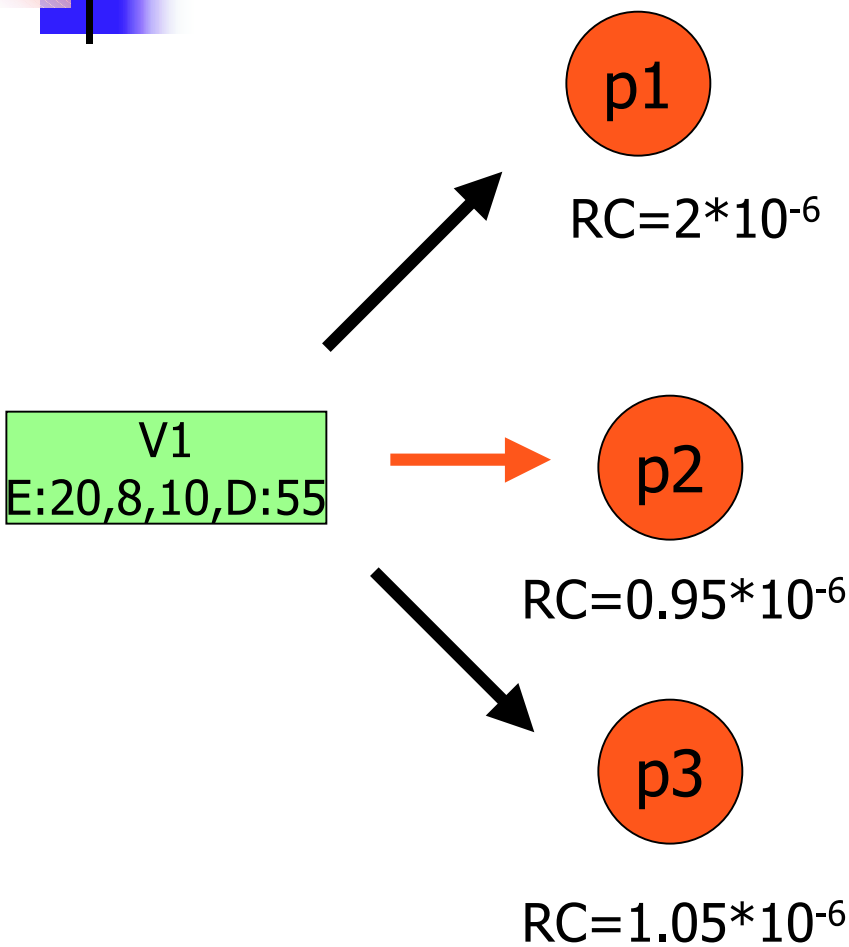


1. Deadlines
2. Precedence constrains

Task sequence =

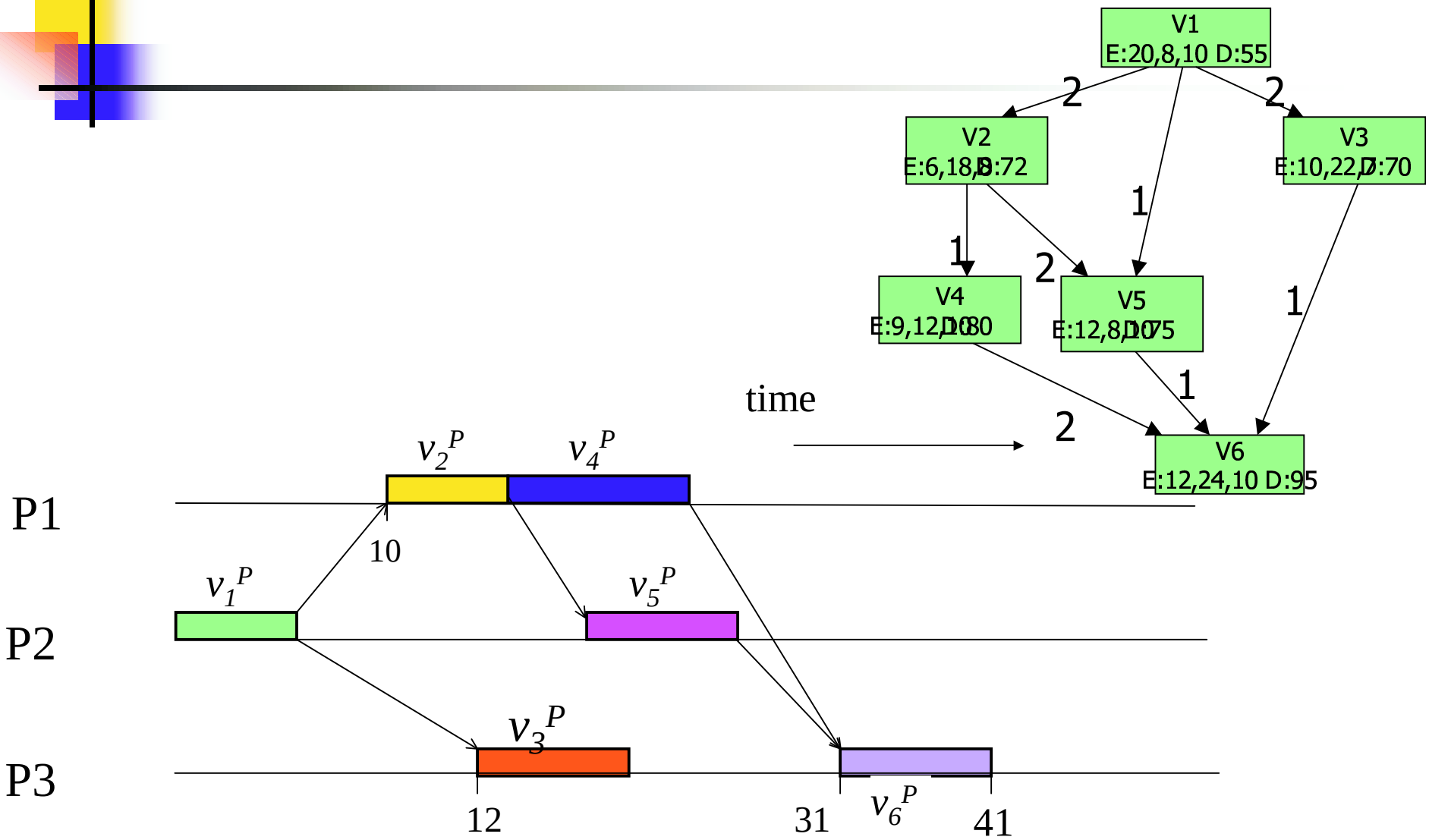
(V1, V3, V2, V5, V4, V6)

Step 2: Schedule Primary copies



1. Can be completed before deadline?
2. Reliability cost is minimized?

Step 2: Schedule Primary copies(cont.)



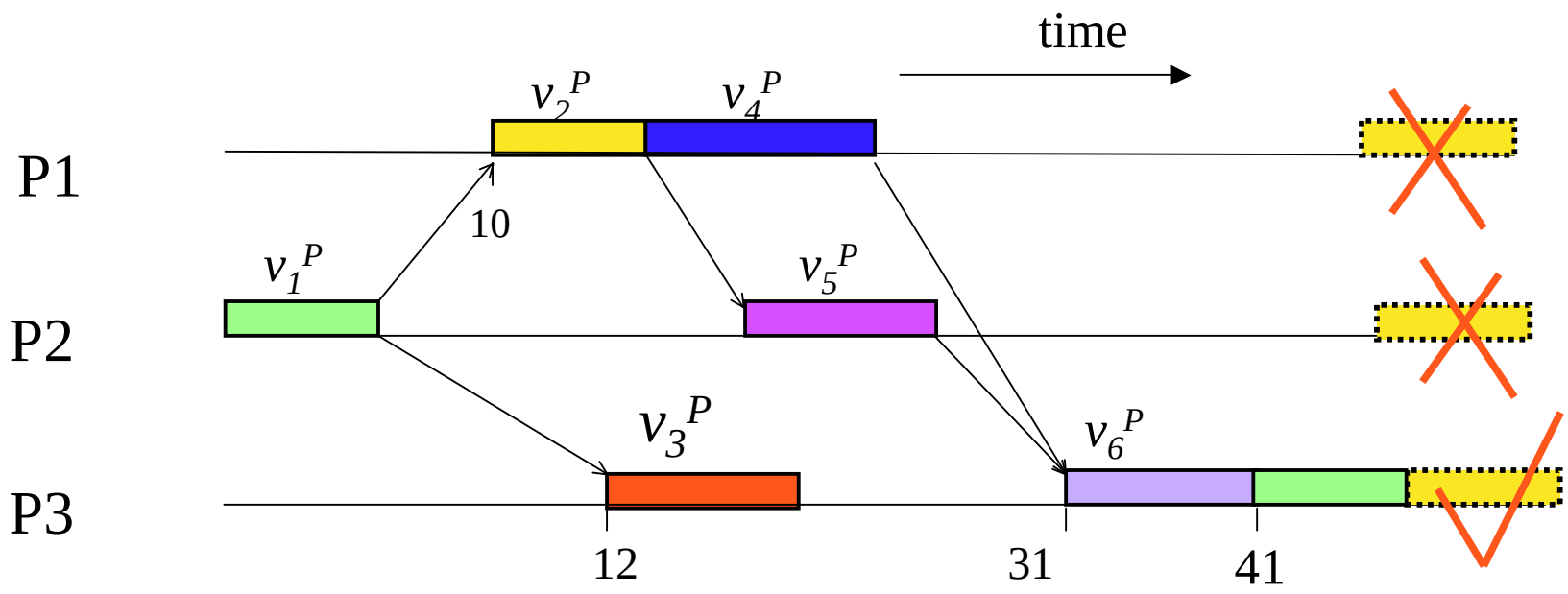
Step 3: Schedule Backup Copies

1. Determine candidate processors
2. For all candidate processors:

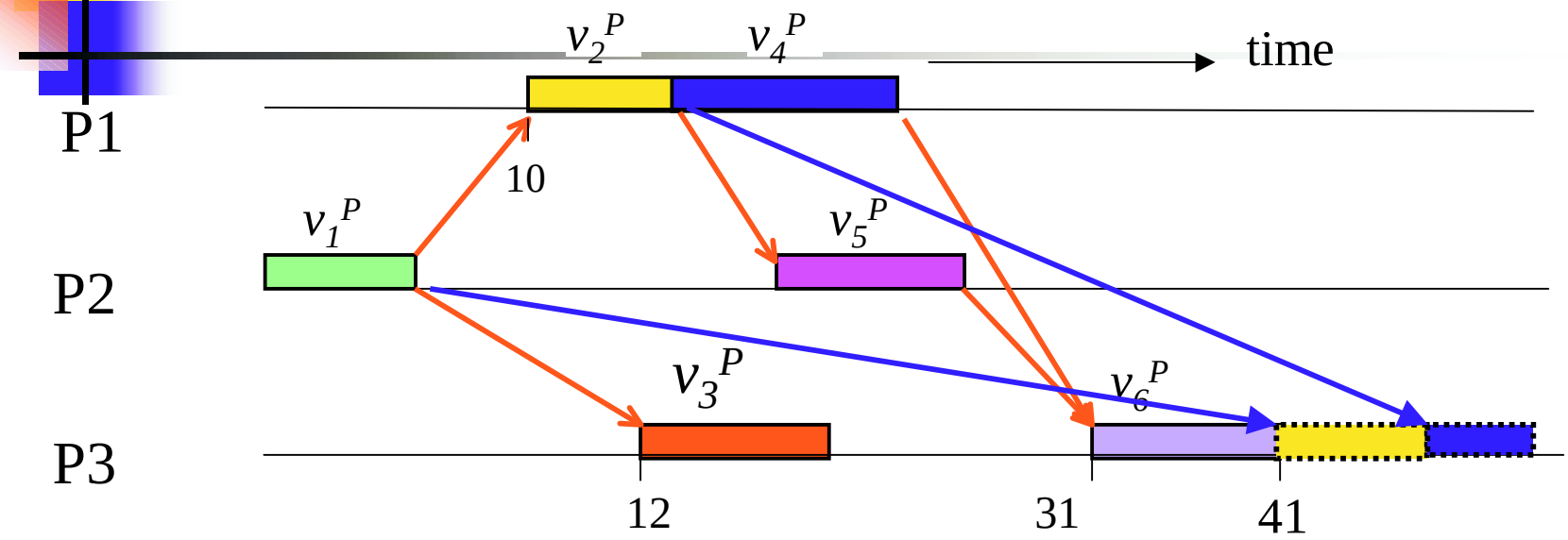
   2.1 Can be completed before deadline?

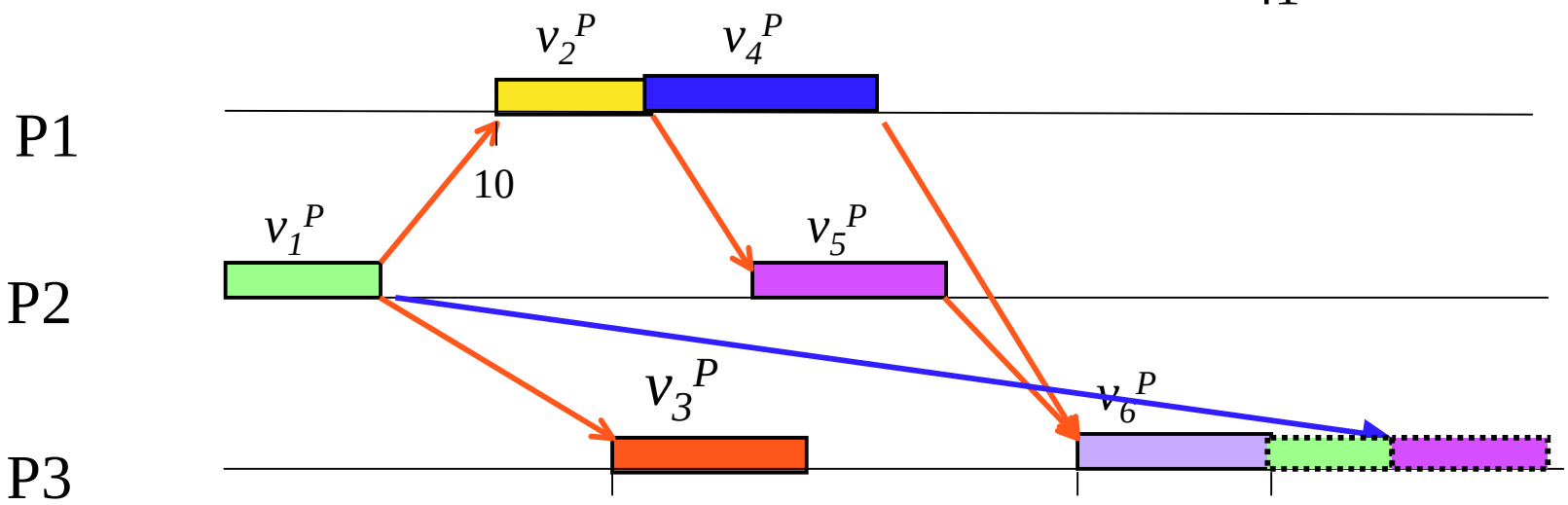
   2.2 Reliability cost is minimized?

Step 3: Schedule Backup Copies (cont.)



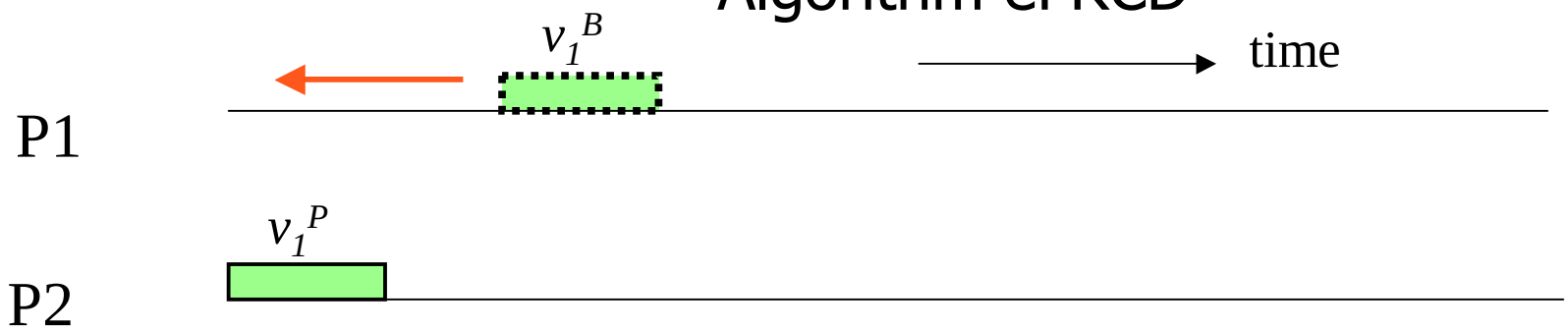
Step 3: Schedule Backup Copies (cont.)



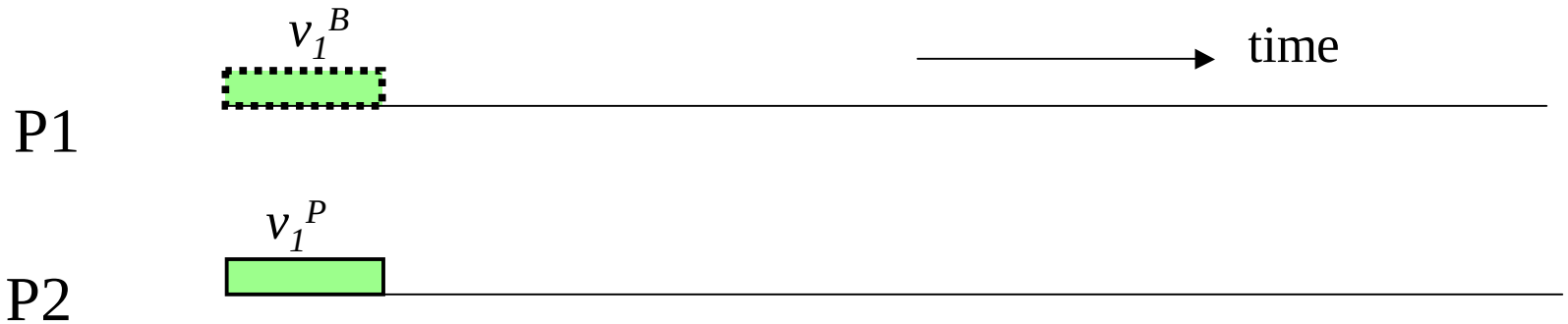


Related Work

Algorithm eFRCD



Algorithm FGLS: A. Girault, C. Lavarenne, M. Sighireanu and Y. Sorel, in ICDCS2001



Related Work (cont.)

Algorithm FGLS: A. Girault, C. Lavarenne, M. Sighireanu and Y. Sorel, “Fault-Tolerant Static Scheduling for Real-Time Distributed Embedded Systems,” *In Proc. of the 21st International Conference on Distributed Computing Systems*(ICDCS), Phoenix, USA, April 2001.

Algorithm OV: Y. Oh and S. H. Son, "Scheduling Real-Time Tasks for Dependability," *Journal of Operational* Research Society, vol. 48, no. 6, pp 629-639, June 1997.

Performance Evaluation (cont.)

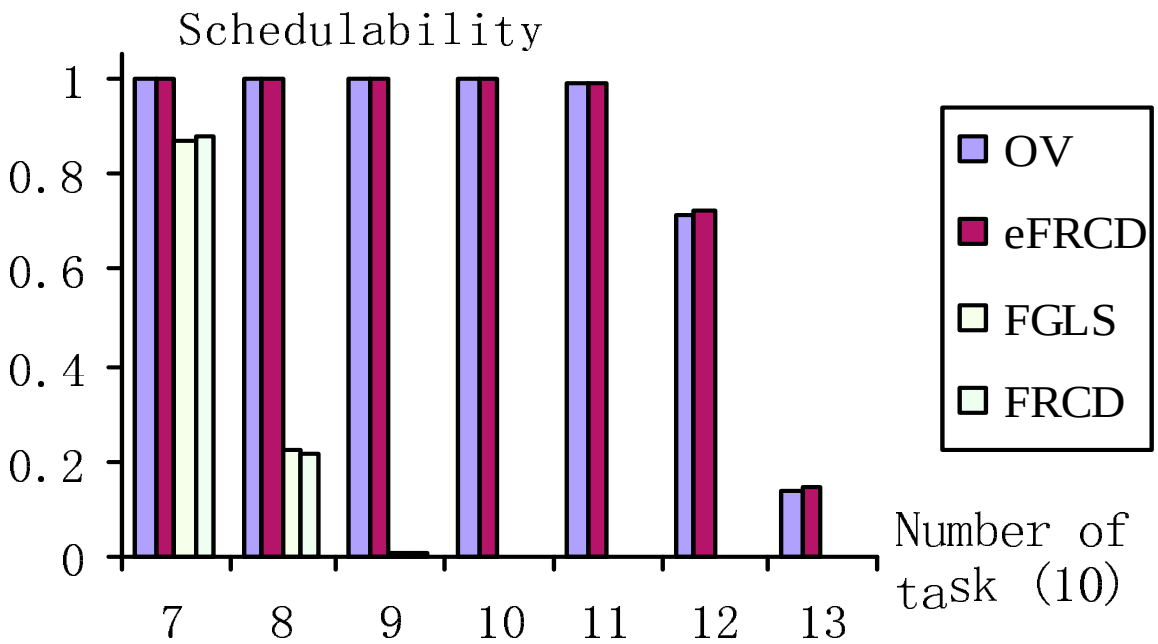
Schedulability =

Number of jobs with feasible schedules / Total number of tested jobs

$$Reliability = e^{-(Reliability\ Cost)}$$

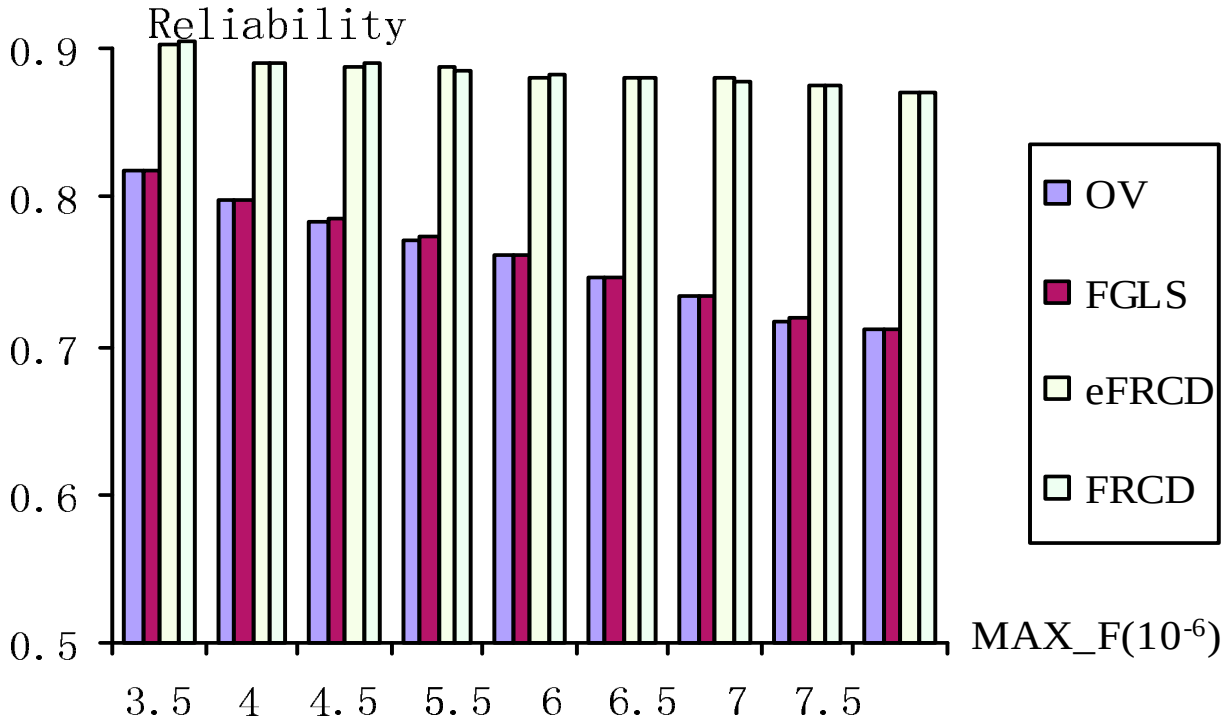
$$Performability = Reliability \times Schedulability$$

Performance Evaluation (cont.)
Schedulability



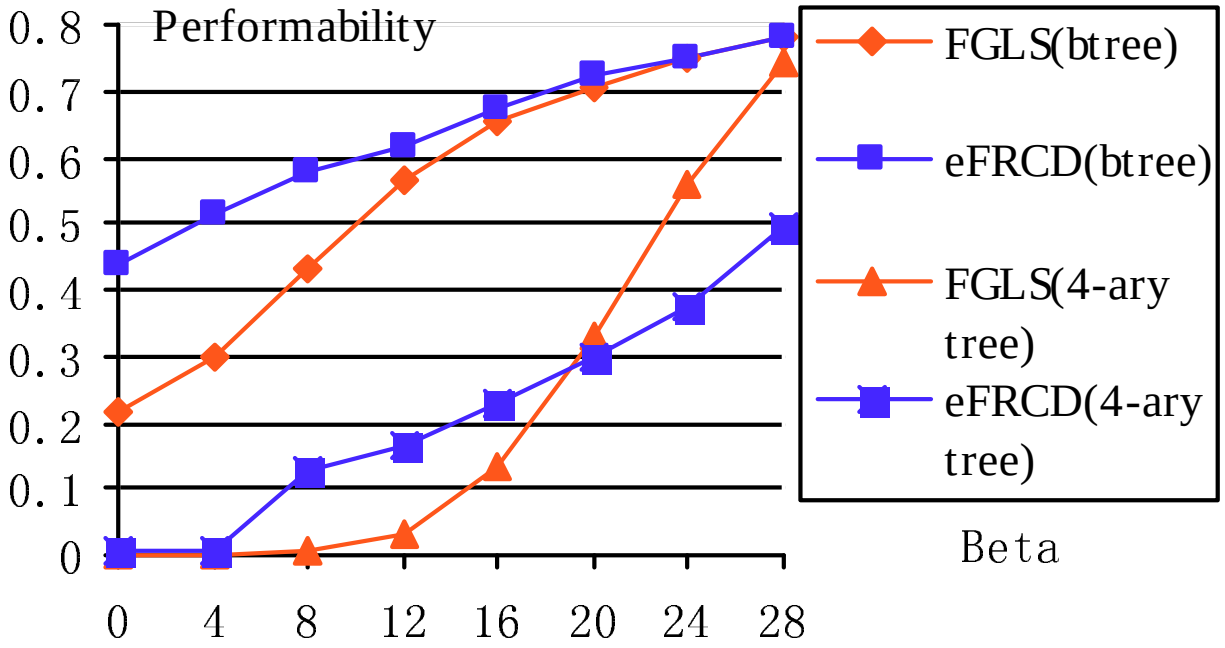
Schedulability as a function of N. Common deadline = 100, m = 16, MIN_F = 0.5*10^{-6}, MAX_F = 3.0*10^{-6}, EX = [1, 20]

Performance Evaluation (cont.)
Reliability



Reliability as function of MAX_F. N = 50, m = 20, MIN_F = $1*10^{-6}$, EX = [500, 1500]

Performance Evaluation (cont.)
Heterogeneity Level



PF of btree and 4-ary tree as a function of heterogeneity level. H=[1,100], N=150, m=20, alpha=20

Conclusions

- Static Scheduling
- Real-time
- Fault tolerance
- Tasks with Precedence Constraints
- Heterogeneous Systems

Thank You