



Non-Preemptive Scheduling with History-Dependent Execution Time

Björn Andersson, Sagar Chaki, Dionisio de Niz
Software Engineering Institute

Brian Dougherty, Jules White
Virginia Tech

Russell Kegley
Lockheed Martin Aeronautics



Example of Task Set in the Old Model

n=3

$T_1=50, D_1=10$

$T_2=150, D_2=15$

$T_3=500, D_3=500$

$C_1=4$

$C_2=7$

$C_3=5$



Example of Task Set in the New Model



Example of Task Set in the New Model

n=3



Example of Task Set in the New Model

$n=3$

$T_1=50, D_1=10$

$T_2=150, D_2=15$

$T_3=500, D_3=500$



Example of Task Set in the New Model

$n=3$

$T_1=50, D_1=10$

$T_2=150, D_2=15$

$T_3=500, D_3=500$

Assume non-preemptive scheduling



Example of Task Set in the New Model

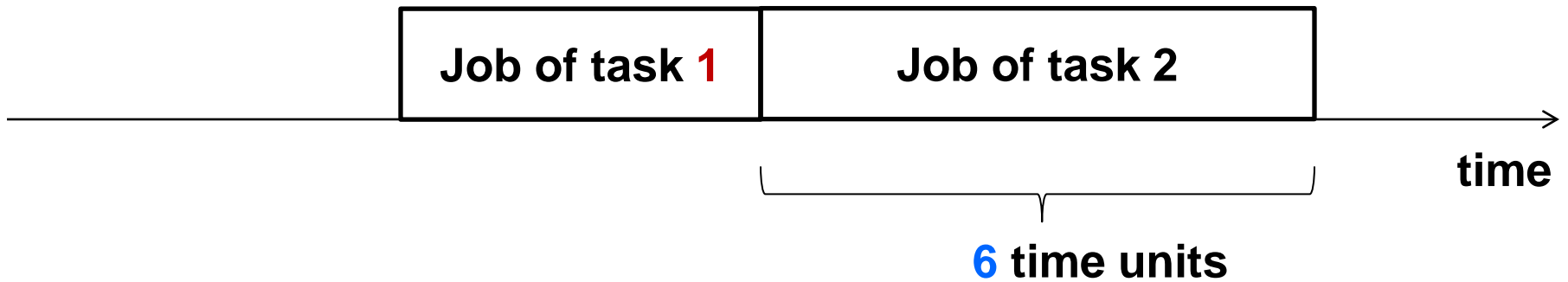
n=3

$T_1=50, D_1=10$

$T_2=150, D_2=15$

$T_3=500, D_3=500$

$ubc_2^2=6, lhubc_2^2=1, ihubc_2^{2,1}=1$



Execution time of a job when a pre-specified history is matched at run-time



Example of Task Set in the New Model

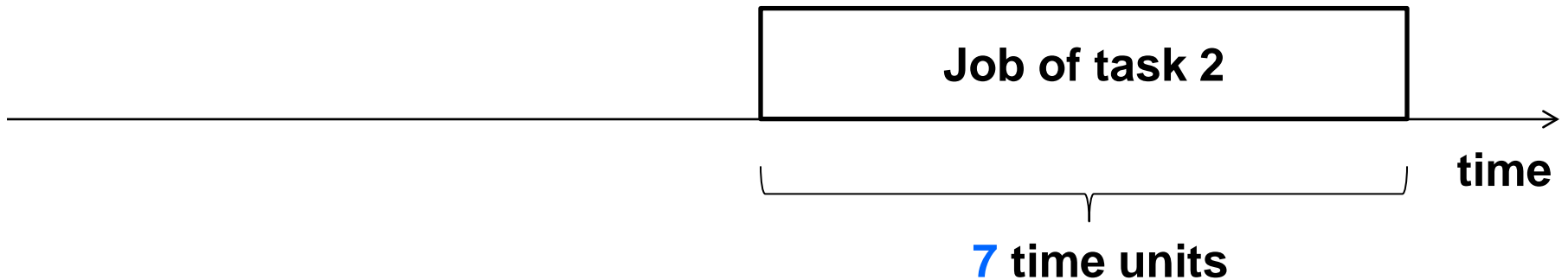
n=3

$T_1=50, D_1=10$

$T_2=150, D_2=15$

$T_3=500, D_3=500$

$ubc_2^1=7, lhubc_2^1=0$



Execution time of a job when the zero-length pre-specified history is matched at run-time



Example of Task Set in the New Model

$n=3$

$T_1=50, D_1=10$

$T_2=150, D_2=15$

$T_3=500, D_3=500$

$n_{hubc_2}=2$

$ubc_2^1=7, lhubc_2^1=0$

$ubc_2^2=6, lhubc_2^2=1, ihubc_2^{2,1}=1$

Pre-specified histories of task τ_2 and their associated upper bounds on execution times



Example of Task Set in the New Model

n=3

$T_1=50, D_1=10$

$T_2=150, D_2=15$

$T_3=500, D_3=500$

$n_{hubc_1}=2$

$ubc_1^1=4, lhubc_1^1=0$

$ubc_1^2=3, lhubc_1^2=1, ihubc_1^{2,1}=2$

$n_{hubc_2}=2$

$ubc_2^1=7, lhubc_2^1=0$

$ubc_2^2=6, lhubc_2^2=1, ihubc_2^{2,1}=1$

$n_{hubc_3}=1$

$ubc_3^1=5, lhubc_3^1=0$

Upper bounds on execution times of a job as a function of jobs that executed before it



Example of Task Set in the New Model

n=3

$$T_1=50, D_1=10$$

$$\text{nhlbc}_1=1$$

$$\text{lbc}_1^1=0, \text{lhbc}_1^1=0$$

$$T_2=150, D_2=15$$

$$\text{nhlbc}_2=1$$

$$\text{lbc}_2^1=0, \text{lhbc}_2^1=0$$

$$T_3=500, D_3=500$$

$$\text{nhlbc}_3=1$$

$$\text{lbc}_3^1=0, \text{lhbc}_3^1=0$$

Lower bound on execution times of a job



Example of Task Set in the New Model

n=3

$T_1=50, D_1=10$

$nhlbc_1=1$

$lbc_1^1=0, lhlbc_1^1=0$

$nhubc_1=2$

$ubc_1^1=4, lhubc_1^1=0$

$ubc_1^2=3, lhubc_1^2=1, ihubc_1^{2,1}=2$

$T_2=150, D_2=15$

$nhlbc_2=1$

$lbc_2^1=0, lhlbc_2^1=0$

$nhubc_2=2$

$ubc_2^1=7, lhubc_2^1=0$

$ubc_2^2=6, lhubc_2^2=1, ihubc_2^{2,1}=1$

$T_3=500, D_3=500$

$nhlbc_3=1$

$lbc_3^1=0, lhlbc_3^1=0$

$nhubc_3=1$

$ubc_3^1=5, lhubc_3^1=0$

Upper and lower bounds on execution times of a job as a function of jobs that executed before it



Example of Task Set in the New Model

n=3

$$T_1=50, D_1=10$$

$$nhlbc_1=1$$

$$lbc_1^1=0, lhlbc_1^1=0$$

$$nhubc_1=2$$

$$ubc_1^1=4, lhubc_1^1=0$$

$$ubc_1^2=3, lhubc_1^2=1, ihubc_1^{2,1}=2$$

$$T_2=150, D_2=15$$

$$nhlbc_2=1$$

$$lbc_2^1=0, lhlbc_2^1=0$$

$$nhubc_2=2$$

$$ubc_2^1=7, lhubc_2^1=0$$

$$ubc_2^2=6, lhubc_2^2=1, ihubc_2^{2,1}=1$$

$$T_3=500, D_3=500$$

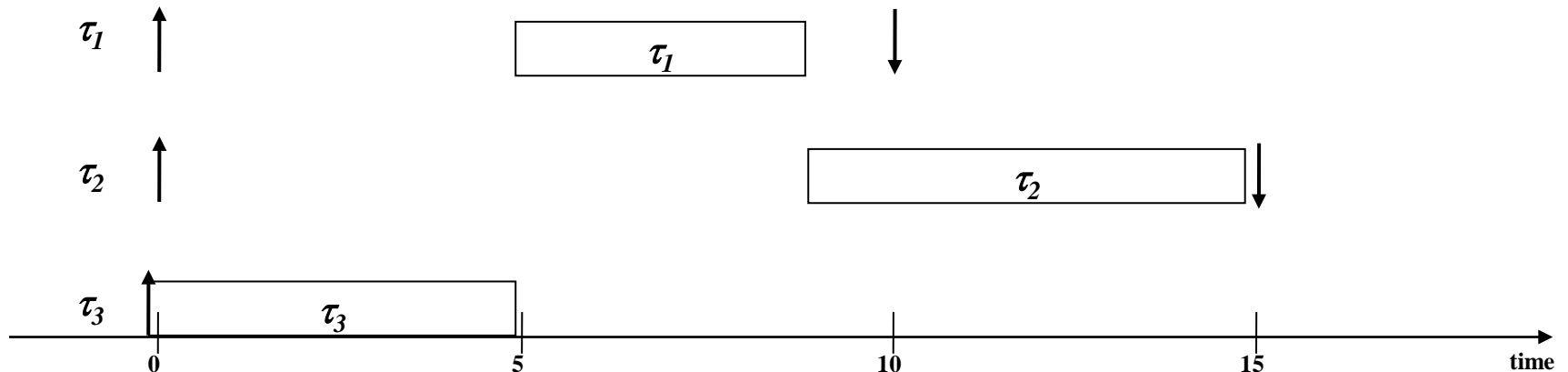
$$nhlbc_3=1$$

$$lbc_3^1=0, lhlbc_3^1=0$$

$$nhubc_3=1$$

$$ubc_3^1=5, lhubc_3^1=0$$

Schedule these tasks with non-preemptive fixed priority scheduling



This is the worst-case arrival for τ_2 . With new model: deadline of τ_2 is met.



Example of Task Set in the Old Model

n=3

$T_1=50, D_1=10$

$T_2=150, D_2=15$

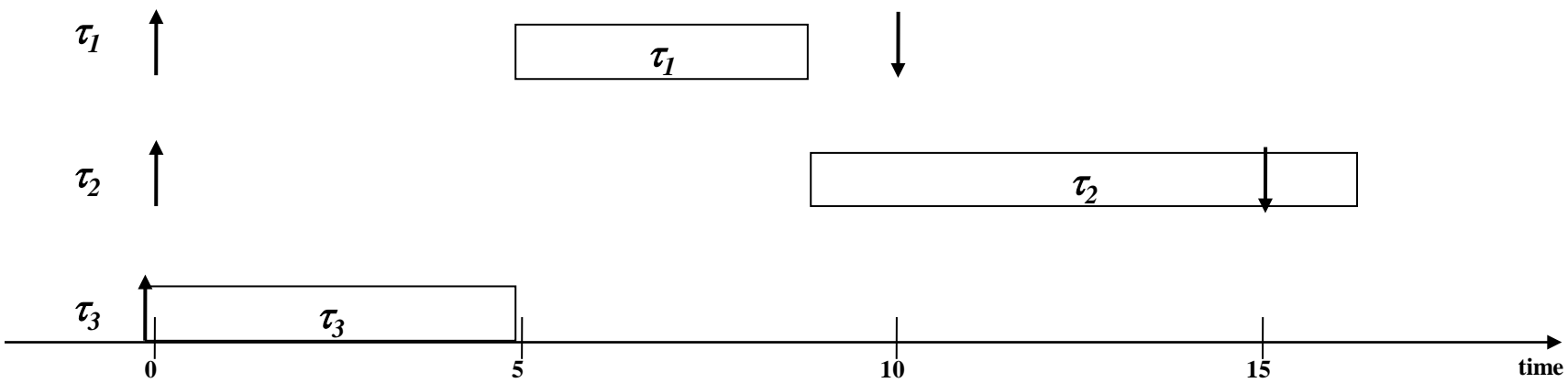
$T_3=500, D_3=500$

$C_1=4$

$C_2=7$

$C_3=5$

Schedule these tasks with non-preemptive fixed priority scheduling



This is the worst-case arrival for τ_2 . With old model: deadline of τ_2 is missed.

Example of Task Set in the New Model

n=3

$$T_1=50, D_1=10$$

$$nhlbc_1=1$$

$$lbc_1^1=0, lhlbc_1^1=0$$

$$nhubc_1=2$$

$$ubc_1^1=4, lhubc_1^1=0$$

$$ubc_1^2=3, lhubc_1^2=1, ihubc_1^{2,1}=2$$

$$T_2=150, D_2=15$$

$$nhlbc_2=1$$

$$lbc_2^1=0, lhlbc_2^1=0$$

$$nhubc_2=2$$

$$ubc_2^1=7, lhubc_2^1=0$$

$$ubc_2^2=6, lhubc_2^2=1, ihubc_2^{2,1}=1$$

$$T_3=500, D_3=500$$

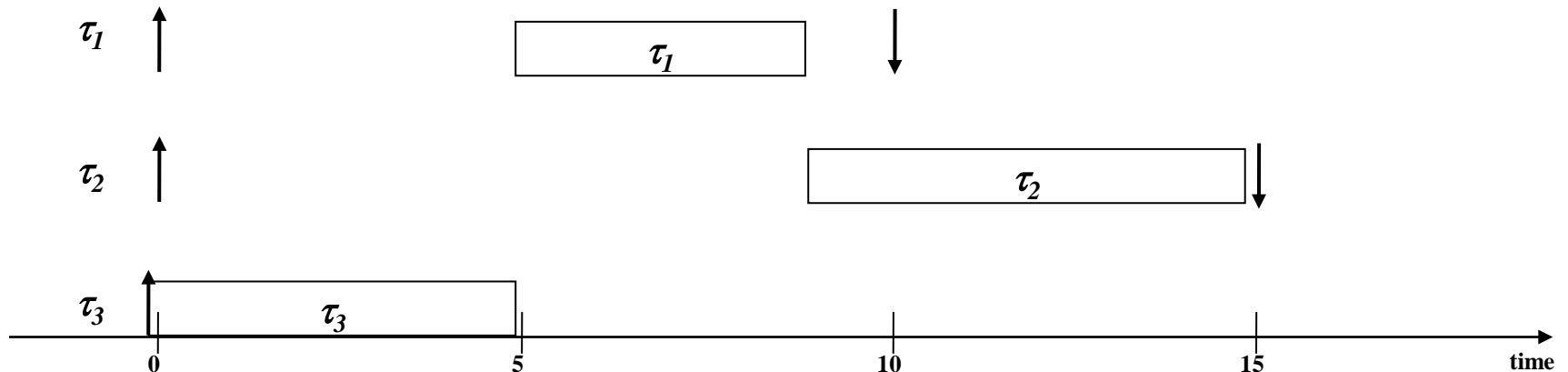
$$nhlbc_3=1$$

$$lbc_3^1=0, lhlbc_3^1=0$$

$$nhubc_3=1$$

$$ubc_3^1=5, lhubc_3^1=0$$

Schedule these tasks with non-preemptive fixed priority scheduling



This is the worst-case arrival for τ_2 . With new model: deadline of τ_2 is met.



Example of Task Set in the New Model

n=3

$T_1=50, D_1=10$

$n_{hlbc_1}=1$

$lbc_1^1=0, lhlbc_1^1=0$

$n_{hubc_1}=2$

$ubc_1^1=4, lhubc_1^1=0$

$ubc_1^2=3, lhubc_1^2=1, ihubc_1^{2,1}=2$

$T_2=150, D_2=15$

$n_{hlbc_2}=1$

$lbc_2^1=0, lhlbc_2^1=0$

$n_{hubc_2}=2$

$ubc_2^1=7, lhubc_2^1=0$

$ubc_2^2=6, lhubc_2^2=1, ihubc_2^{2,1}=1$

$T_3=500, D_3=500$

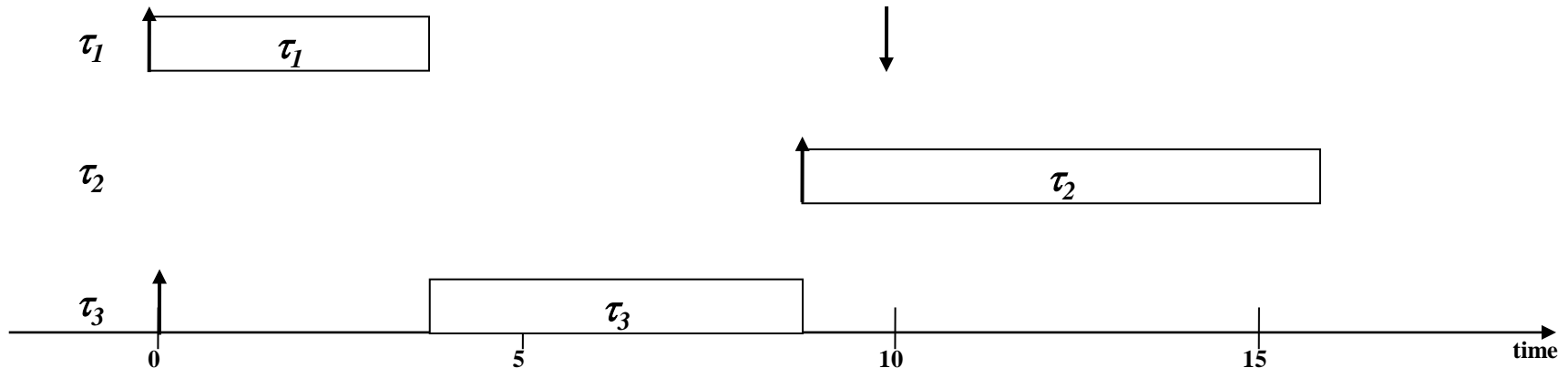
$n_{hlbc_3}=1$

$lbc_3^1=0, lhlbc_3^1=0$

$n_{hubc_3}=1$

$ubc_3^1=5, lhubc_3^1=0$

Schedule these tasks with non-preemptive fixed priority scheduling



For another arrival for τ_2 . Job of τ_2 has longer execution time but deadline of τ_2 is met.



Problem discussed in this presentation

How to perform schedulability analysis of tasks described with the new model under fixed-priority non-preemptive scheduling?

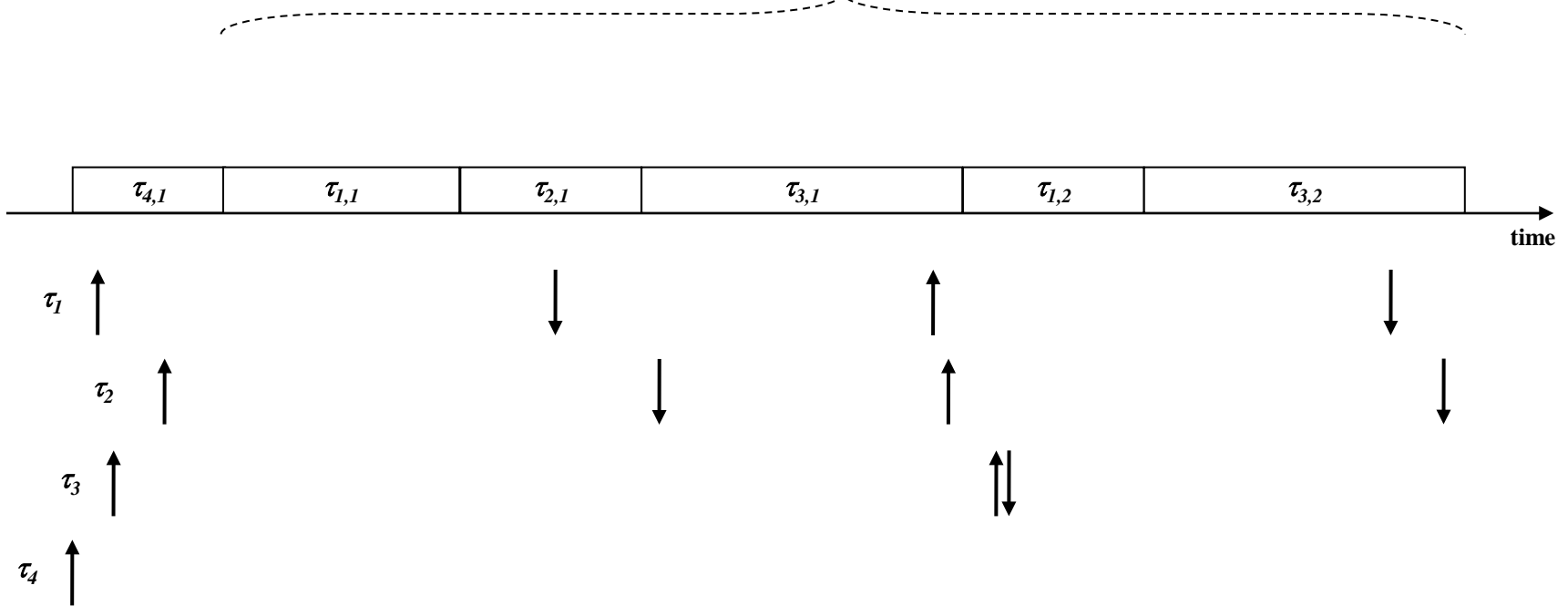


Ideas that do not work

- L&L critical instant



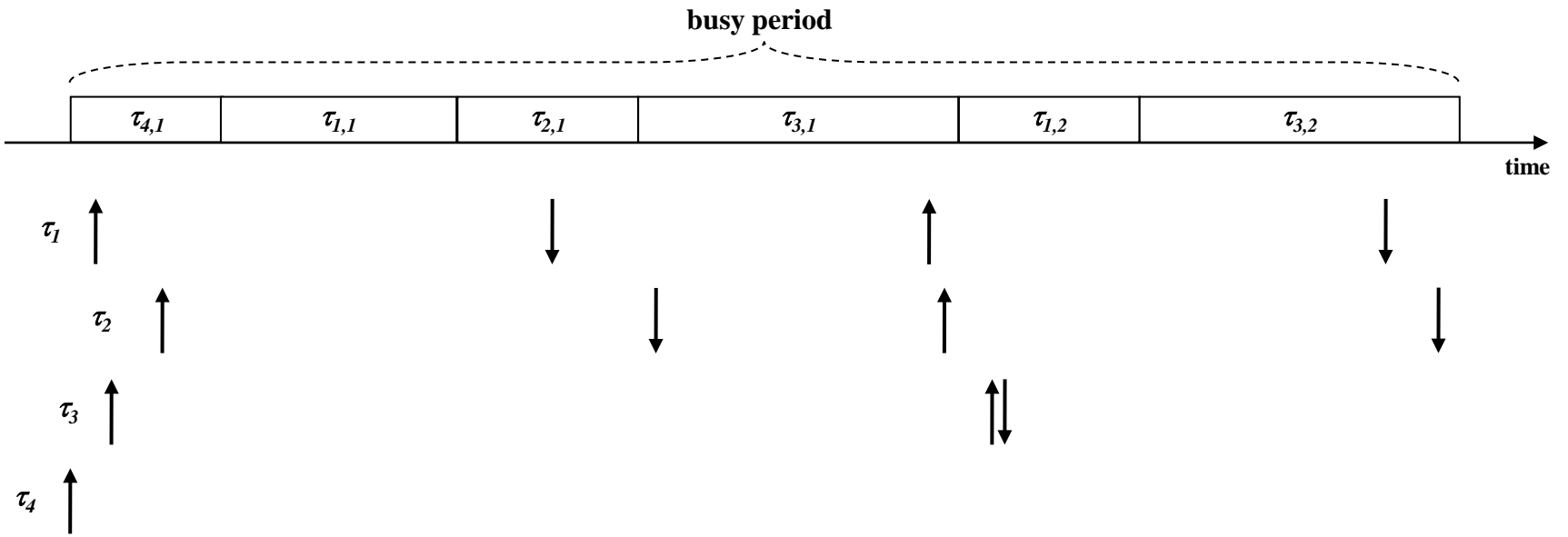
level-3 busy period

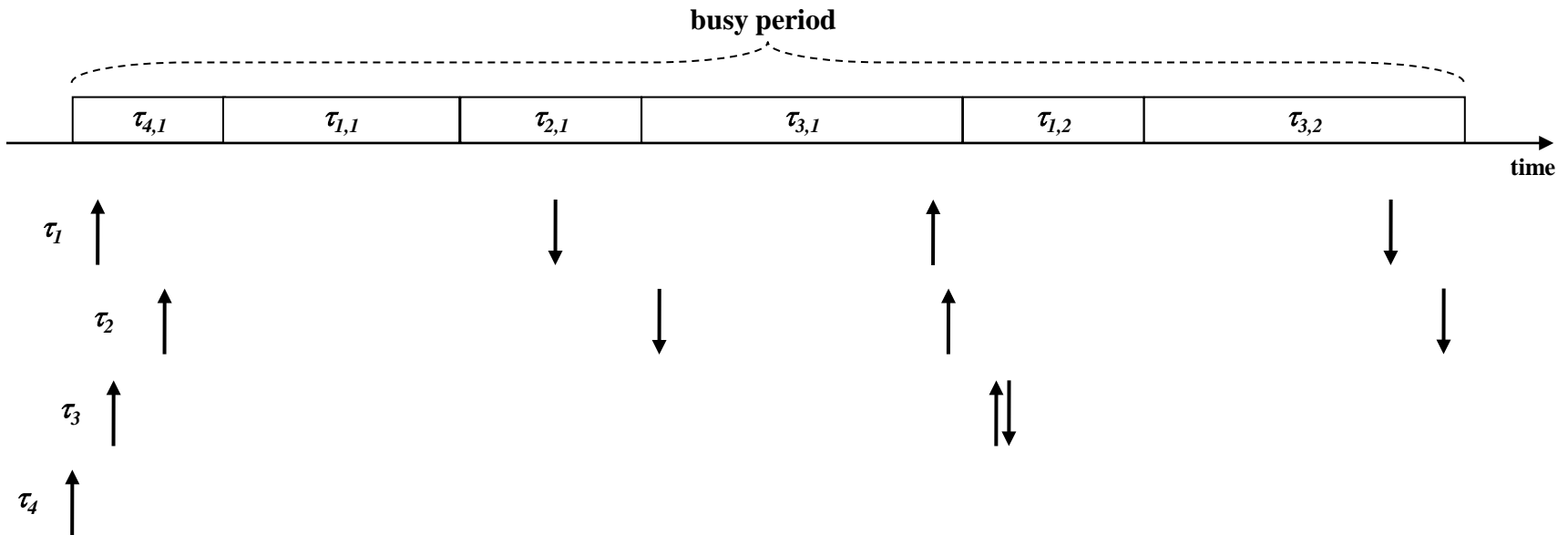


Ideas that do not work

- L&L critical instant
- Level- i busy period

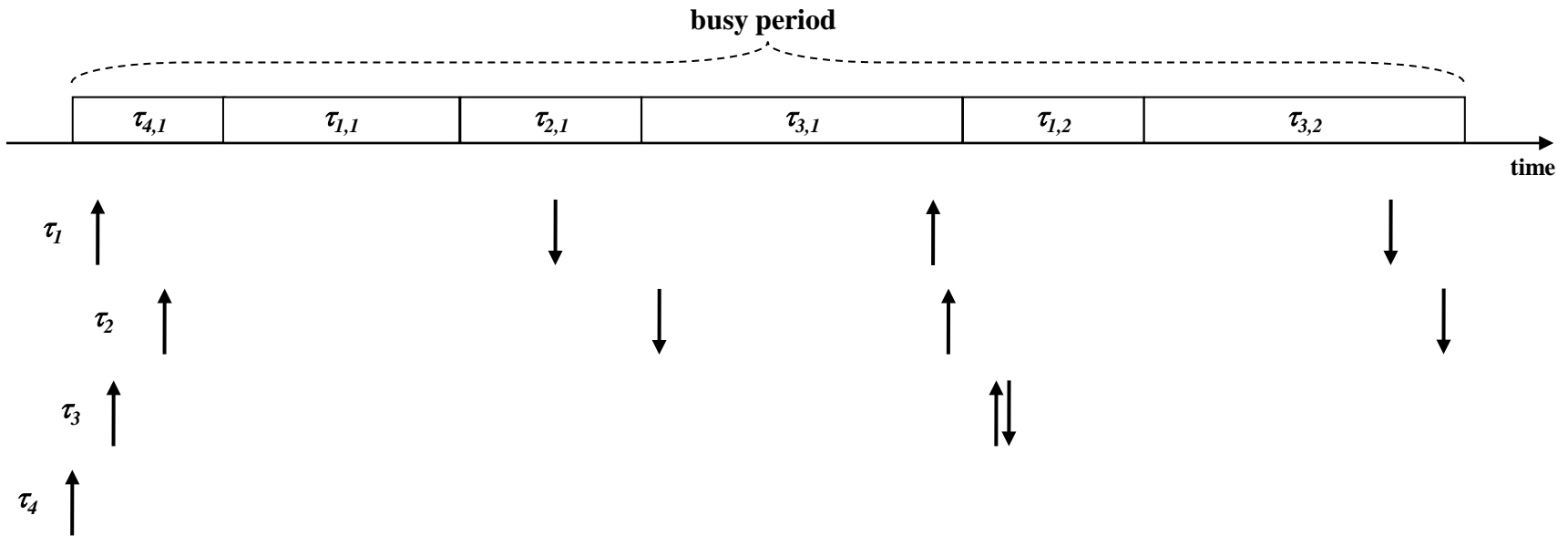






1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $R_{i,q} :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L
5. $R_i := \max_{q=1..Q} R_{i,q}$

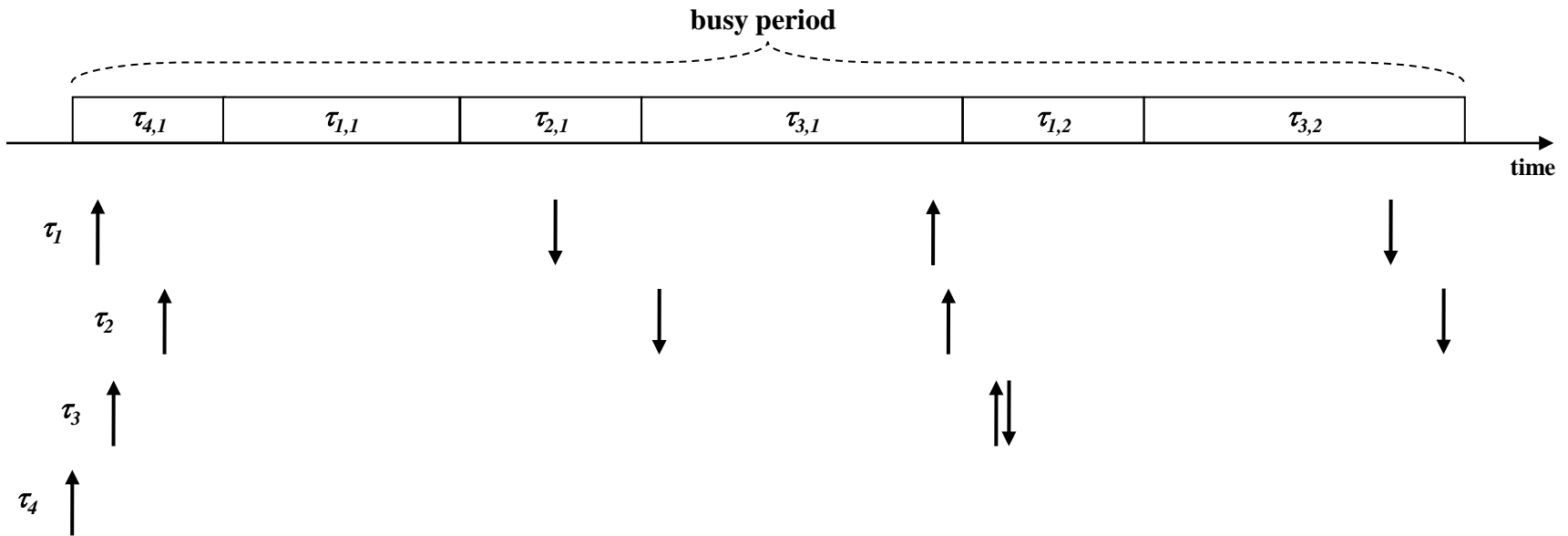




An idea that almost works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $R_{i,q} :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L
5. $R_i := \max_{q=1..Q} R_{i,q}$

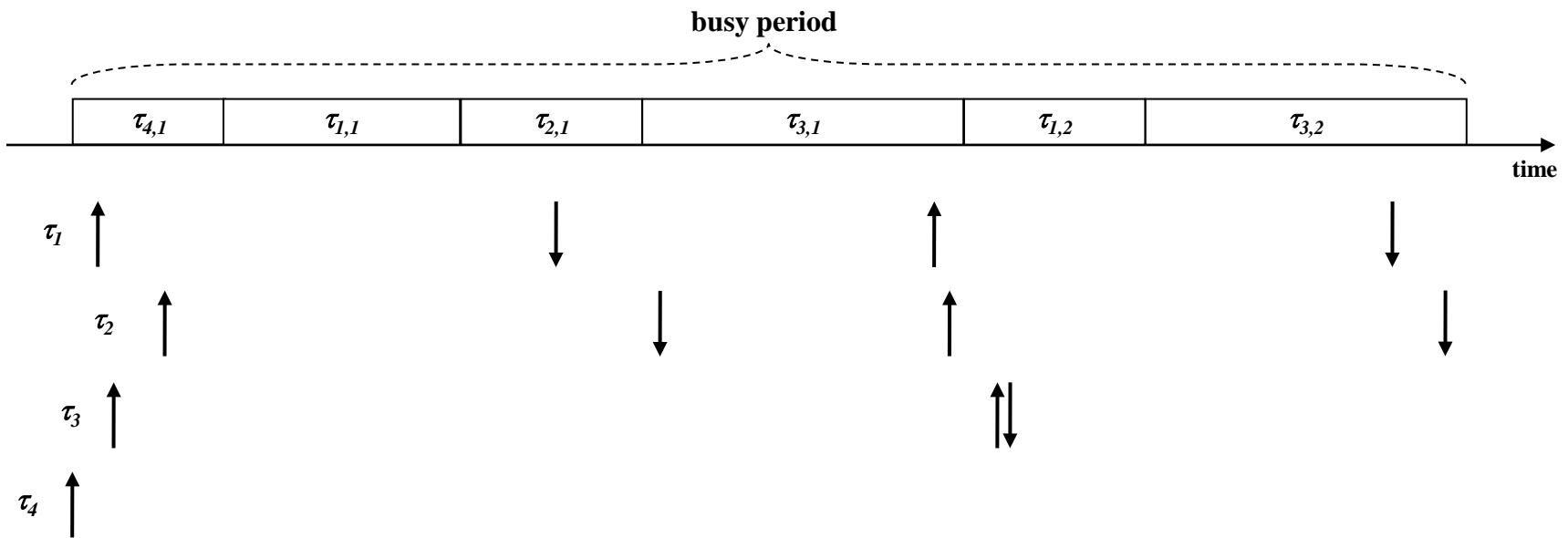




An idea that works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, valid_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$



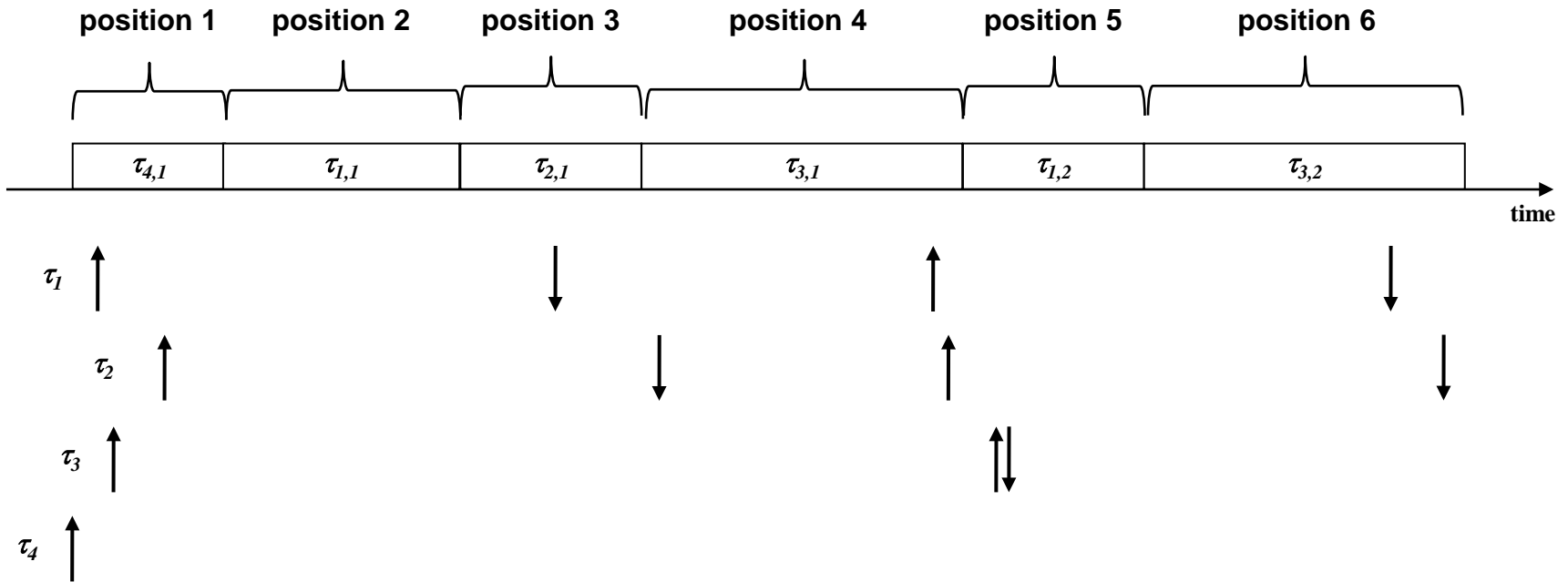


An idea that works for computing the response time of task

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, \text{valid}_{i,q} \rangle :=$ compute the maximum response time of — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period
5. $R_i = \max_{q=1..Q \text{ and } \text{valid}_{i,q}} R_{i,q}$

How to perform step 1 and step 4?

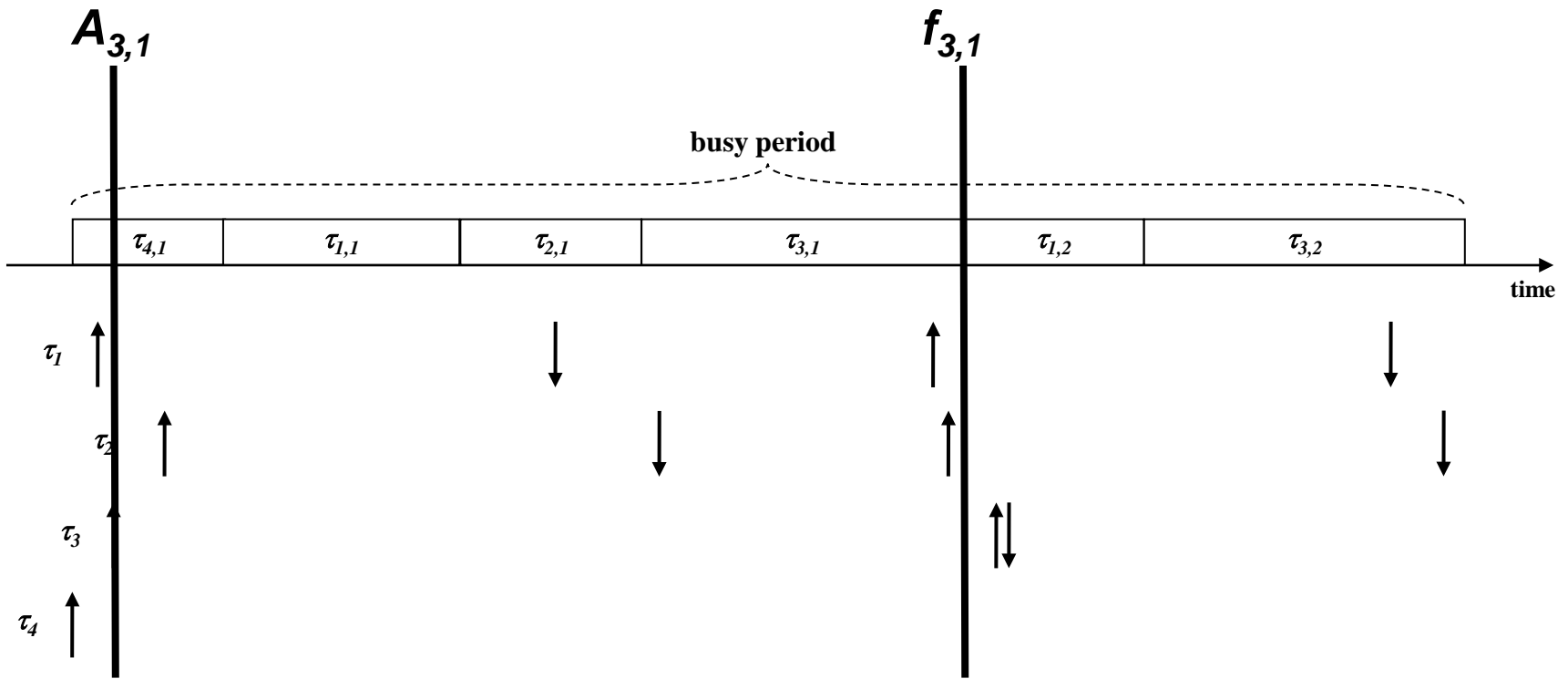




An idea that works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, \text{valid}_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } \text{valid}_{i,q}} R_{i,q}$

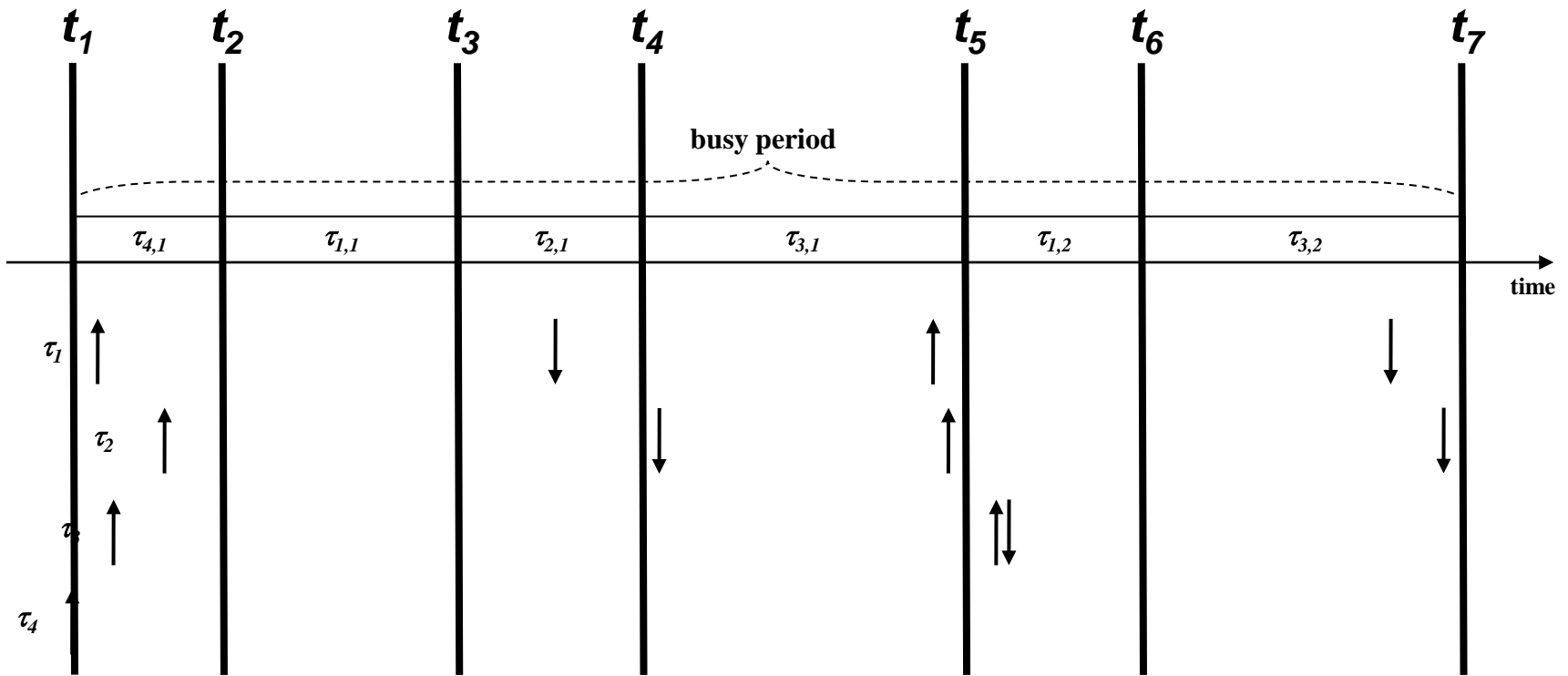




An idea that works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, valid_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$

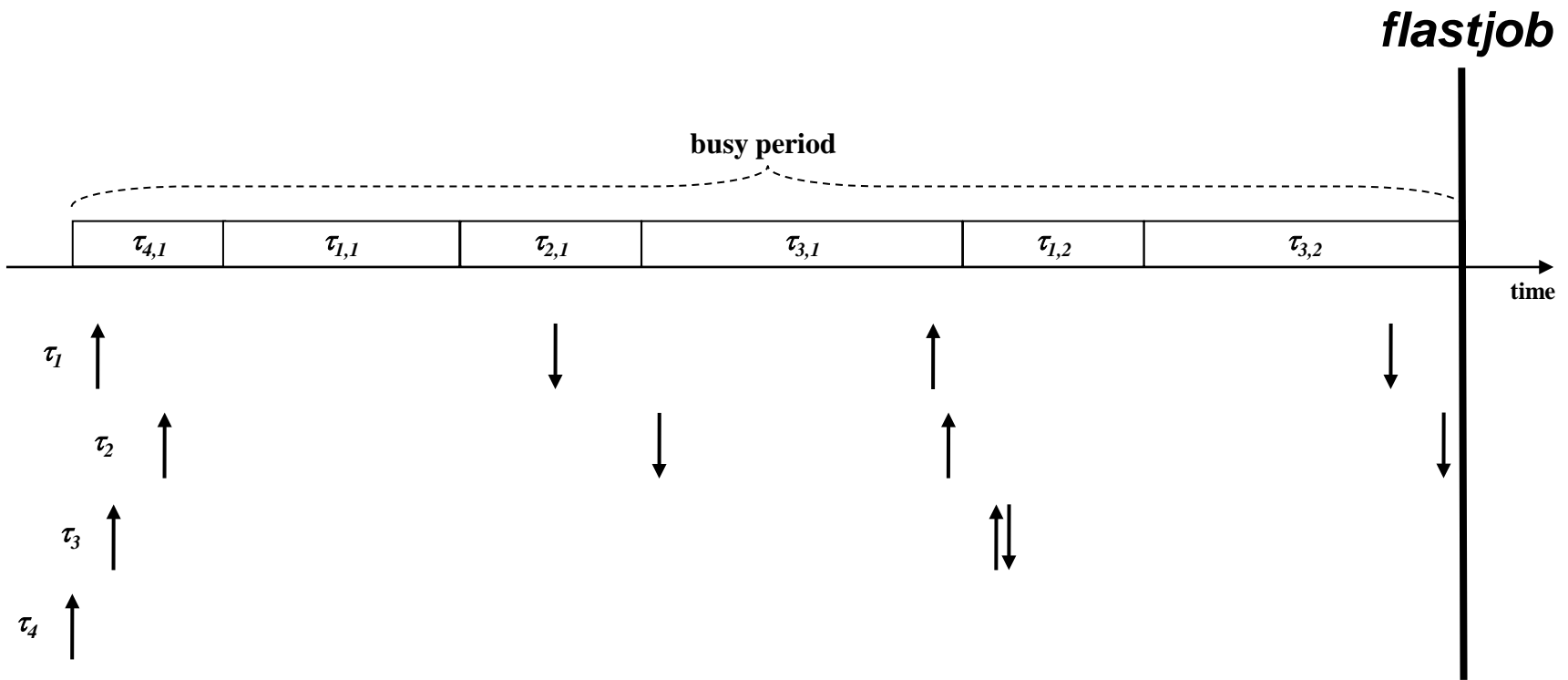




An idea that works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, \text{valid}_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } \text{valid}_{i,q}} R_{i,q}$

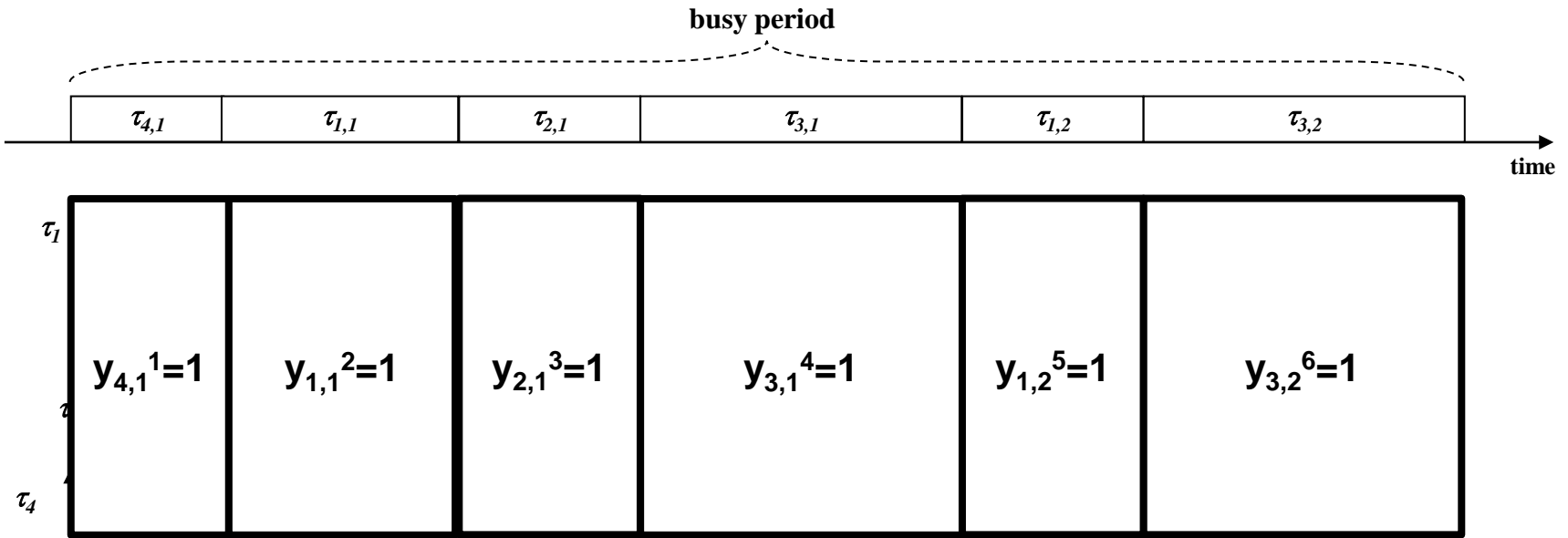




An idea that works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, valid_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$

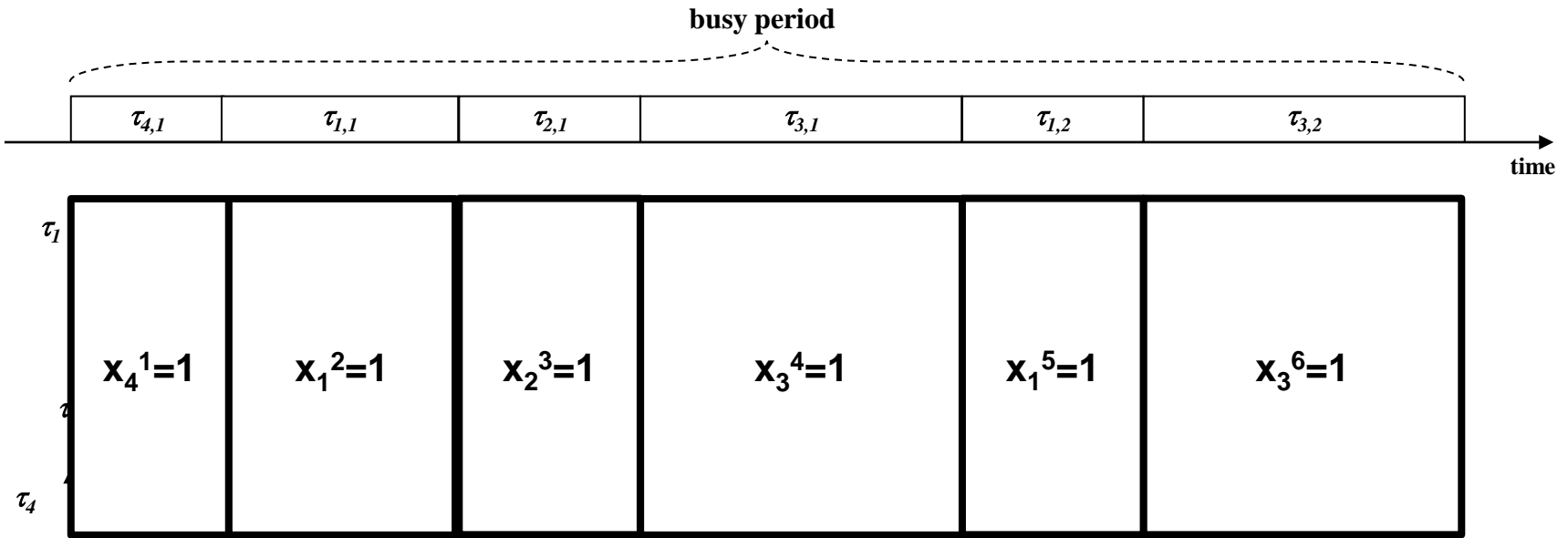




An idea that works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, \text{valid}_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } \text{valid}_{i,q}} R_{i,q}$





An idea that works for computing the response time of task τ_i

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, valid_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$
 — the q :th job of task τ_i — in every busy period of length
 at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$



Perform step 1: Compute the maximum duration of a busy period

Represent a schedule.

x_j^p = 1 iff a job of task τ_j executes in position p in busy period

$y_{j,k}^p$ = 1 iff job $\tau_{j,k}$ executes in position p in busy period

t_k = time of k :th context switch in busy period

Other variables

$A_{j,k}$ = arrival time of $\tau_{j,k}$

$f_{j,k}$ = finishing time of $\tau_{j,k}$

$ftlastjob$ = time when the busy period ends

Maximize $ftlastjob - t_1$ subject to constraints [see paper]

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, valid_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q} \text{and } valid_{i,q} R_{i,q}$



Perform step 4: Compute the maximum response time of $\tau_{i,q}$ during every busy period of duration at most L

Represent a schedule.

x_j^p = 1 iff a job of task τ_j executes in position p in busy period

$y_{j,k}^p$ = 1 iff job $\tau_{j,k}$ executes in position p in busy period

t_k = time of k :th context switch in busy period

Other variables

$A_{j,k}$ = arrival time of $\tau_{j,k}$

$f_{j,k}$ = finishing time of $\tau_{j,k}$

$ftlastjob$ = time when the busy period ends

Maximize $f_{j,k} - A_{j,k}$ subject to constraints [see paper]

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. **for** $q := 1$ to Q_i **do**
4. $\langle R_{i,q}, valid_{i,q} \rangle :=$ compute the maximum response time of $\tau_{i,q}$ — the q :th job of task τ_i — in every busy period of length at most L and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q} \text{and } valid_{i,q} R_{i,q}$



Conclusion

It is possible to compute exact response times of tasks scheduled by non-preemptive fixed-priority scheduling where execution times depend on history.



Thanks for listening!



Copyright 2011 Carnegie Mellon University.

This material is based upon work supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

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.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:* 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 external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

*These restrictions do not apply to U.S. government entities.

