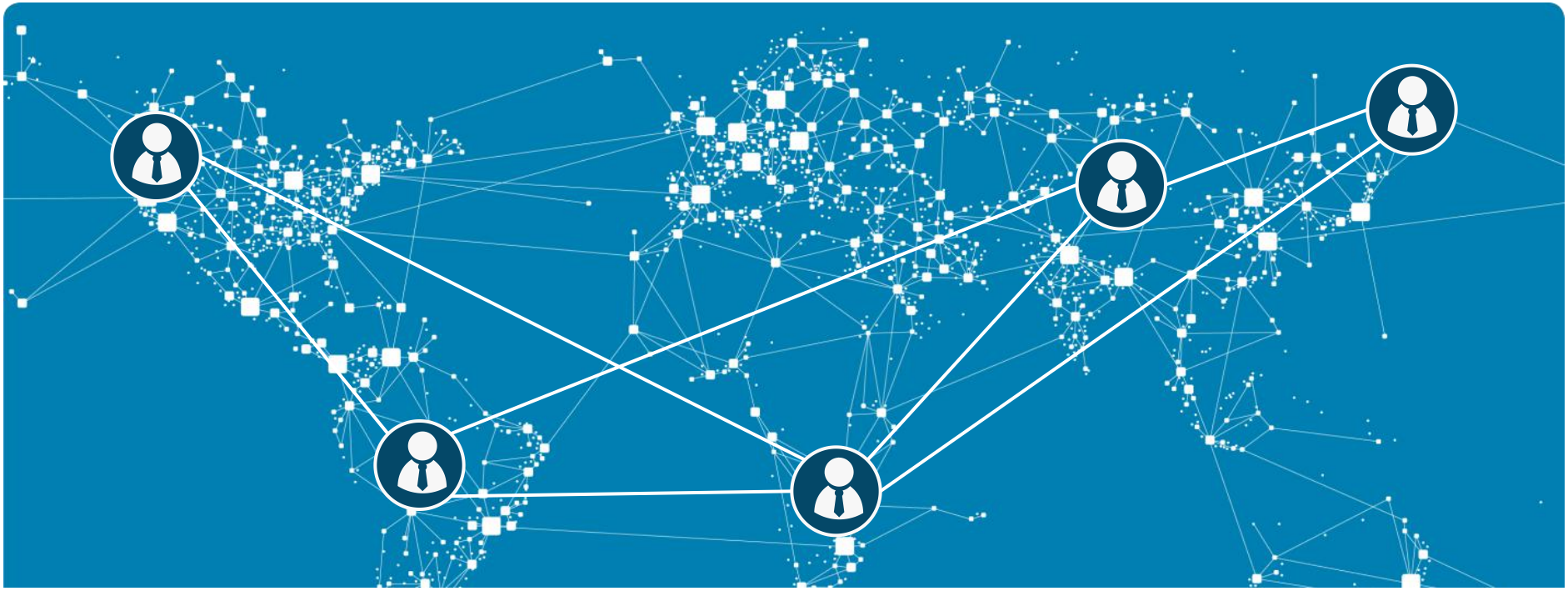


Response-Time Analysis for Task Chains in Communicating Threads
（论文阅读报告）

宋金林　导师：李仁发教授

论文基本信息

Response-Time Analysis for Task Chains in Communicating Threads

Johannes Schlatow and Rolf Ernst
Institute of Computer and Network Engineering, TU Braunschweig
{schlatow,ernst}@ida.ing.tu-bs.de

- 作者：Johannes Schlatow and Rolf Ernst
- 单位：Institute of Computer and Network Engineering, TU Braunschweig （计算机与网络工程研究所，德国布伦瑞克工业大学）
- 会议：RTAS2016（CCF B类）

目录

- 摘要
- 论文贡献
- 忙窗口分析方法
- 同步任务链响应时间分析
- 异步任务链响应时间分析
- 实验&结果
- 总结

摘要

在为软件组件建模进行时序分析时，我们通常会遇到具有优先关系的任务链。在实时系统中通常对其端到端的延时有苛刻的要求。本文扩展了忙窗口分析方法，可以对静态优先抢占系统中的任务链进行分析，并且通过仿真以及真实数据的测试，展示了比现有的分析方法更紧凑的响应时间界限。

论文贡献

- 提出一种基于忙窗口的完整任务链最坏响应时间分析方法，改善了最坏情况下端到端的时间延时界限。并且该方法具有良好的适用性。

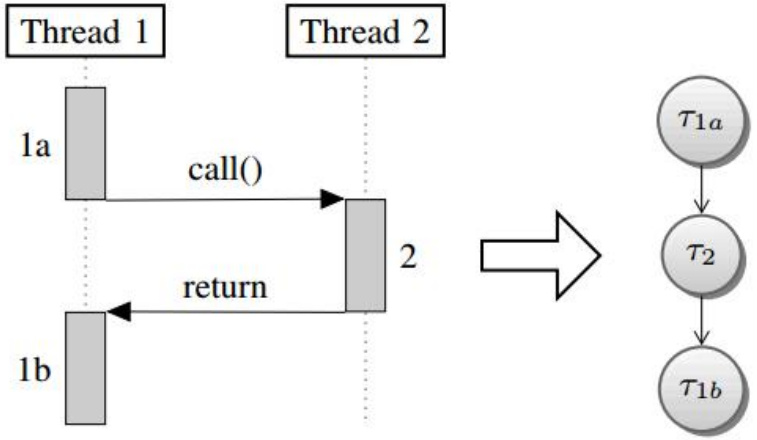


Figure 1. Communicating threads (implementation) naturally split up into a chain of tasks (timing model).

通信线程—>任务链模型

忙窗口分析方法

- 相关概念
- C_i^+ / C_i^- ：任务最坏/好执行时间；
- $\eta^+(\Delta t) / \eta^-(\Delta t)$ ：单位时间t内，任务触发的最大/小次数；
- $\delta^-(n) / \delta^+(n)$ ：n个事件序列，最后一个事件和第一个事件的最大/小距离；

忙窗口分析方法

- q-event busy-window：系统忙处理任务i的q个实例所需要的最大时间。

$$B_i(q) = q \cdot C_i^+ + \sum_{j \in \mathcal{I}_i} \eta_j^+(B_i(q)) \cdot C_j^+$$

- 假设前提：当前任务在上一个任务完成之前触发（除了第一个任务），所以q的最大值计算：

$$Q_i = \max\{n : \forall q \in \mathbb{N}^+, q \leq n : \delta_i^-(q) \leq B_i(q-1)\}$$

- 单个任务最坏响应时间：

$$R_i^+ = \max_{q \in [1, Q_i]} (B_i(q) - \delta_i^-(q))$$

- 整个任务链的最坏响应时间：简单的将所有相关任务的WCRT相加。

同步任务链响应时间分析

- Q：对单个任务进行忙窗口分析方法，任务链内的任务之间的干扰被多次计算——导致WCRT分析悲观
- 解决方法：从整条任务链的角度出发，进行忙窗口分析（q-event task-chain busy window）

基本前提：

- 静态优先级抢占
- 一条任务链i包含多个任务（$T_{i1}, T_{i2}, \ldots T_{in}$），任务链内的任务入度出度都为1
- 任意的优先级

同步任务链响应时间分析

- q-event task-chain busy window：忙处理q个完整任务链所需要的时间：

$$B_i^{sc}(q) = q\sum_k C_{ik}^{+} + \sum_{j\neq i}\left(\sum_{k\in\mathcal{I}_{ij}^{c}}(\eta_j^{+}(B_i^{sc}(q))C_{jk}^{+}) + \sum_{k\in\mathcal{S}_{ij}^{crit}} C_{jk}^{+}\right) \tag{10}$$

自身执行时间

高优先级抢占
（任意抢占）

高优先级抢占
（只抢占一次）

异步任务链响应时间分析

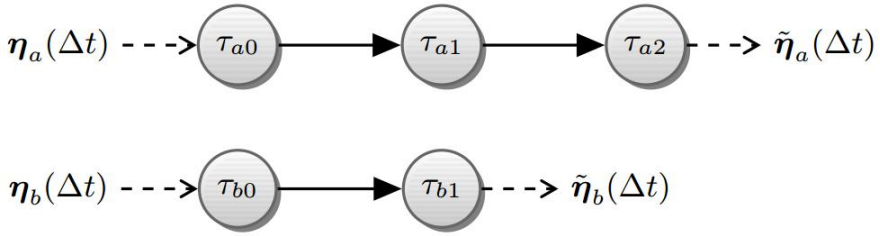


Figure 4. Two synchronous task chains a and b

$$B_i^{sc}(q) = q\sum_k C_{ik}^+ + \sum_{j\neq i}\left(\sum_{k\in\mathcal{I}_{ij}^c}(\eta_j^+(B_i^{sc}(q))C_{jk}^+) + \sum_{k\in\mathcal{S}_{ij}^{crit}} C_{jk}^+\right) \quad (10)$$

$$B_i^{ac}(q) = \sum_k \max(\eta_i^+(B_i^{ac}(q)), q)C_{ik}^+ + \sum_{j\neq i}\left(\sum_{k\in\mathcal{I}_{ij}^c}(\eta_j^+(B_i^{ac}(q))C_{jk}^+) + \sum_{k\in\mathcal{D}_{ij}^{ac}} C_{jk}^+\right) \quad (12)$$

$$D_{ba}^{ac} = \{2\}$$

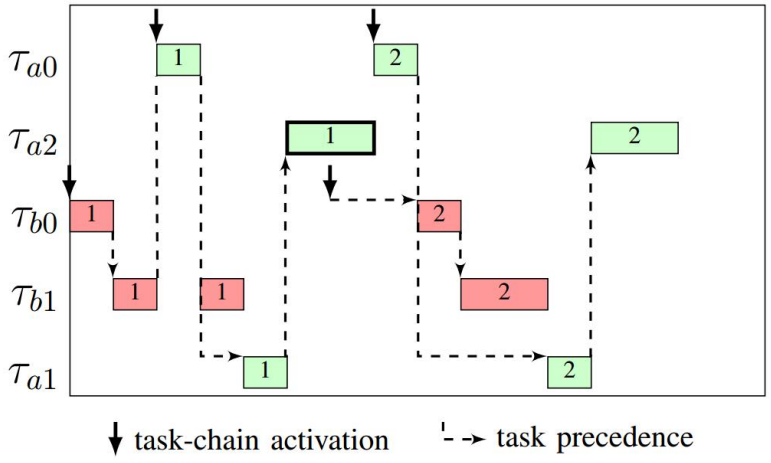


Figure 5. Gantt chart for a deferred synchronous activation of τ_{a2} (bold).

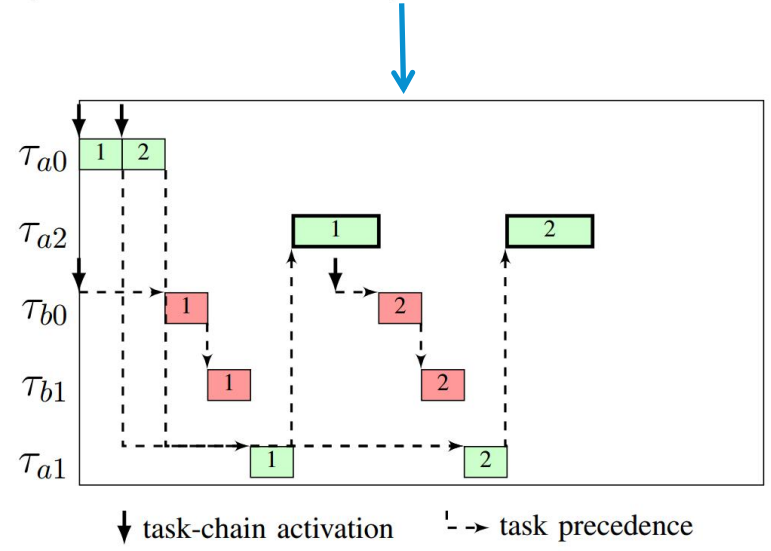
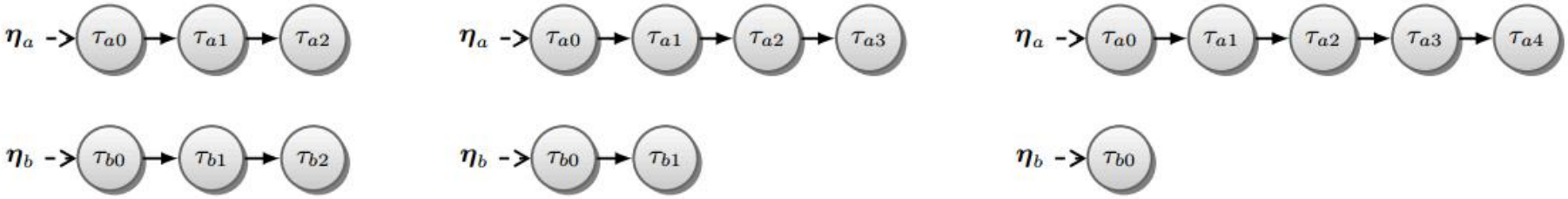


Figure 6. Gantt chart for an interleaved execution of task chain a and deferred asynchronous activations of τ_{a2} (bold).

实验-仿真数据



(a) 3:3 experiment (b) 4:2 experiment (c) 5:1 experiment

Figure 7. Task chains used for the different experiments.

6个任务有着固定的WCET/BCET互不相同的优先级，并对所有可能的优先级分配情况进行分析（6!种）

对比：

①传统分析方法

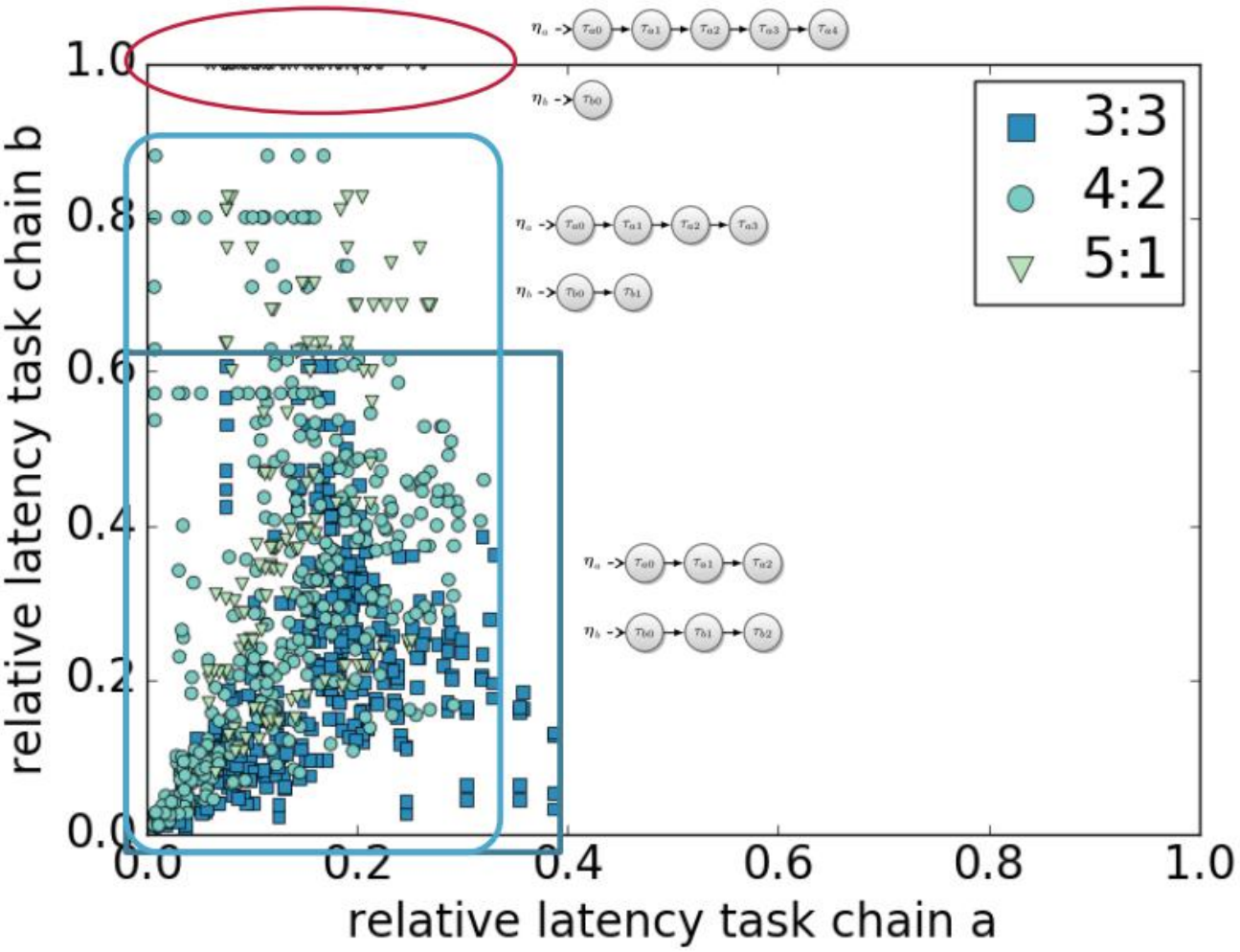
②任务链忙窗口方法

比较整条任务链的WCRT

Task(s)	τ_{a0}	τ_{a1}	τ_{a2}	τ_{b0}	τ_{b1}, τ_{a4}	τ_{b2}, τ_{a3}
WCET	10	2	4	3	9	5
BCET	1	2	2	1	4	3

Experiment	Period a	Jitter a	Period b	Jitter b
3:3	20	5	100	0
4:2	30	5	100	0
5:1	40	5	100	0

Synchronous



relative latency improvement:

改进方法

$$\frac{task\ chain\ WCRT}{\sum_i WCRT_i}$$

传统方法

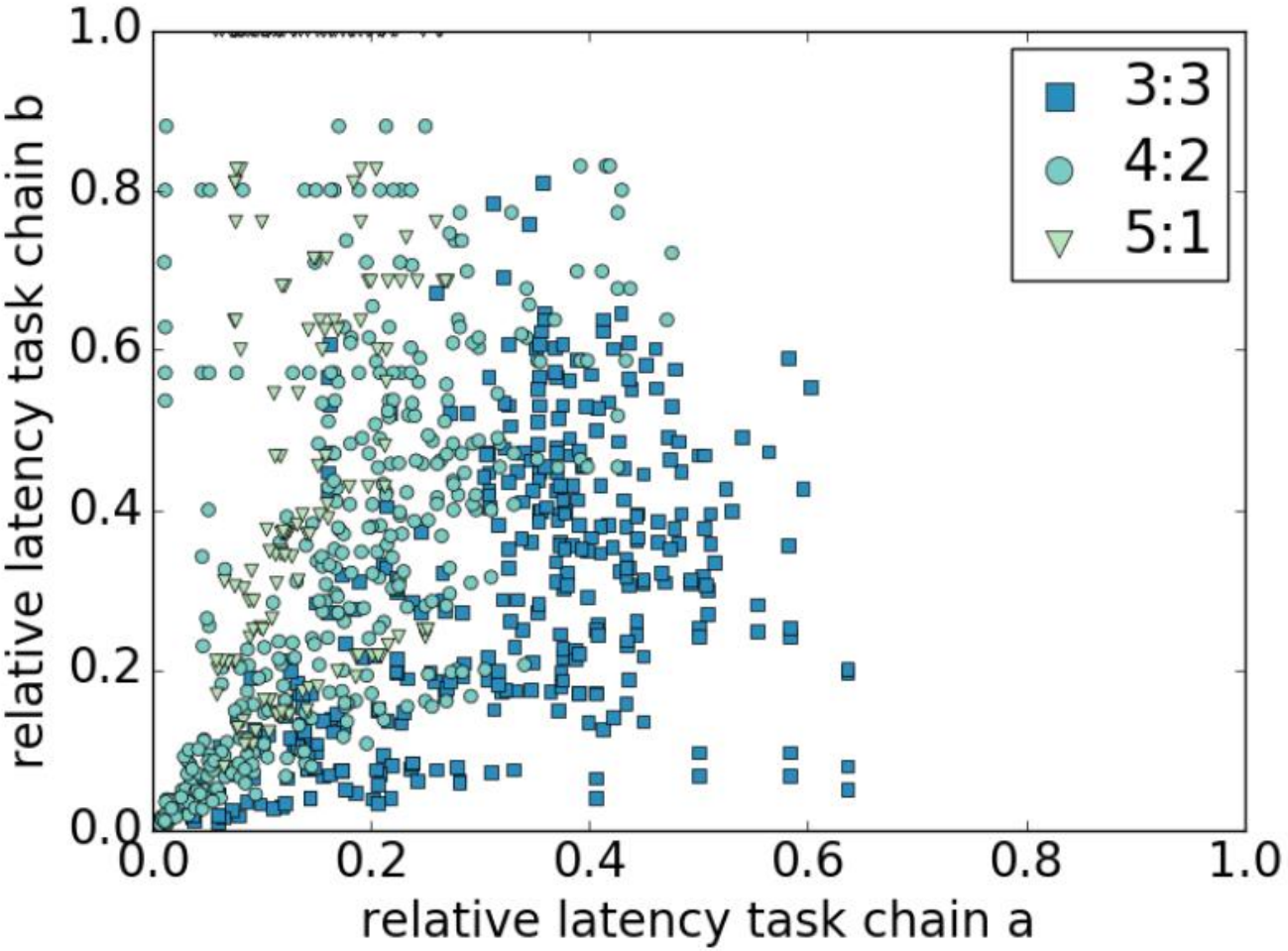
median improvement:

3:3) a: 0.18 | b: 0.19

4:2) a: 0.13 | b: 0.29

5:1) a: 0.13 | b: 0.6

Asynchronous



smaller improvement
due to self-interference

median improvement:
3:3) a: 0.35 | b: 0.29
4:2) a: 0.17 | b: 0.33
5:1) a: 0.13 | b: 0.6

工业应用实例测试

汽车ADAS系统中的两个功能链：
Parking assistant （停车辅助）和 lane detection （车道检测）

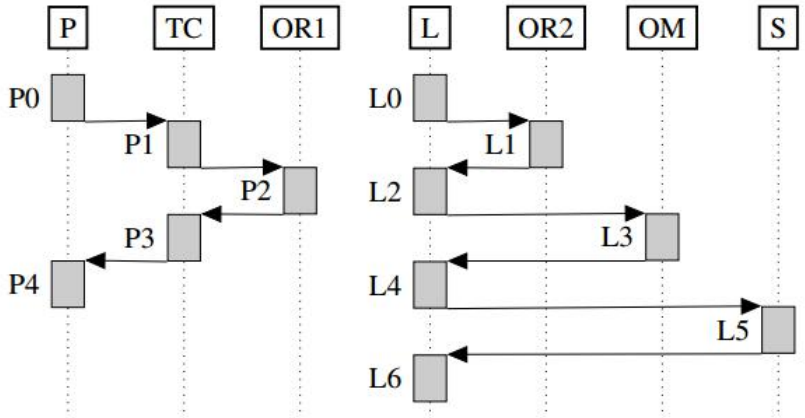
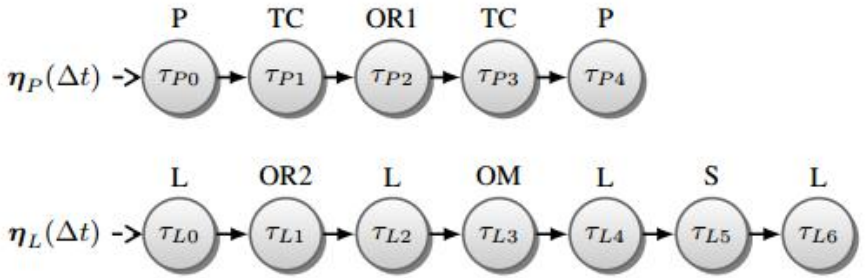


Figure 10. Thread communication for the park and lane assist use case.



目的：在给定的任务链最大延时约束下（150ms）寻找一个合适的优先级分配策略。（需要分析5040种优先级分配情况）

实验结果

- Conventional CPA (设置固定的迭代次数100次):
  - 分析5040种可能的优先级分配方法需要近8小时
  - 几乎都不能收敛，无法分析（5036/5040）
  - 延迟结果范围 4949 ~ 8613ms (P), 1017 ~ 2322ms (L)
  - 无法满足时间约束
- MAST:(offset-based analysis with precedence relations)
  - 分析需要34s
  - 只有11种情况满足时间约束需求
- Task-chain busy window:
  - 分析需要22s
  - 有2880种情况满足时间约束需求

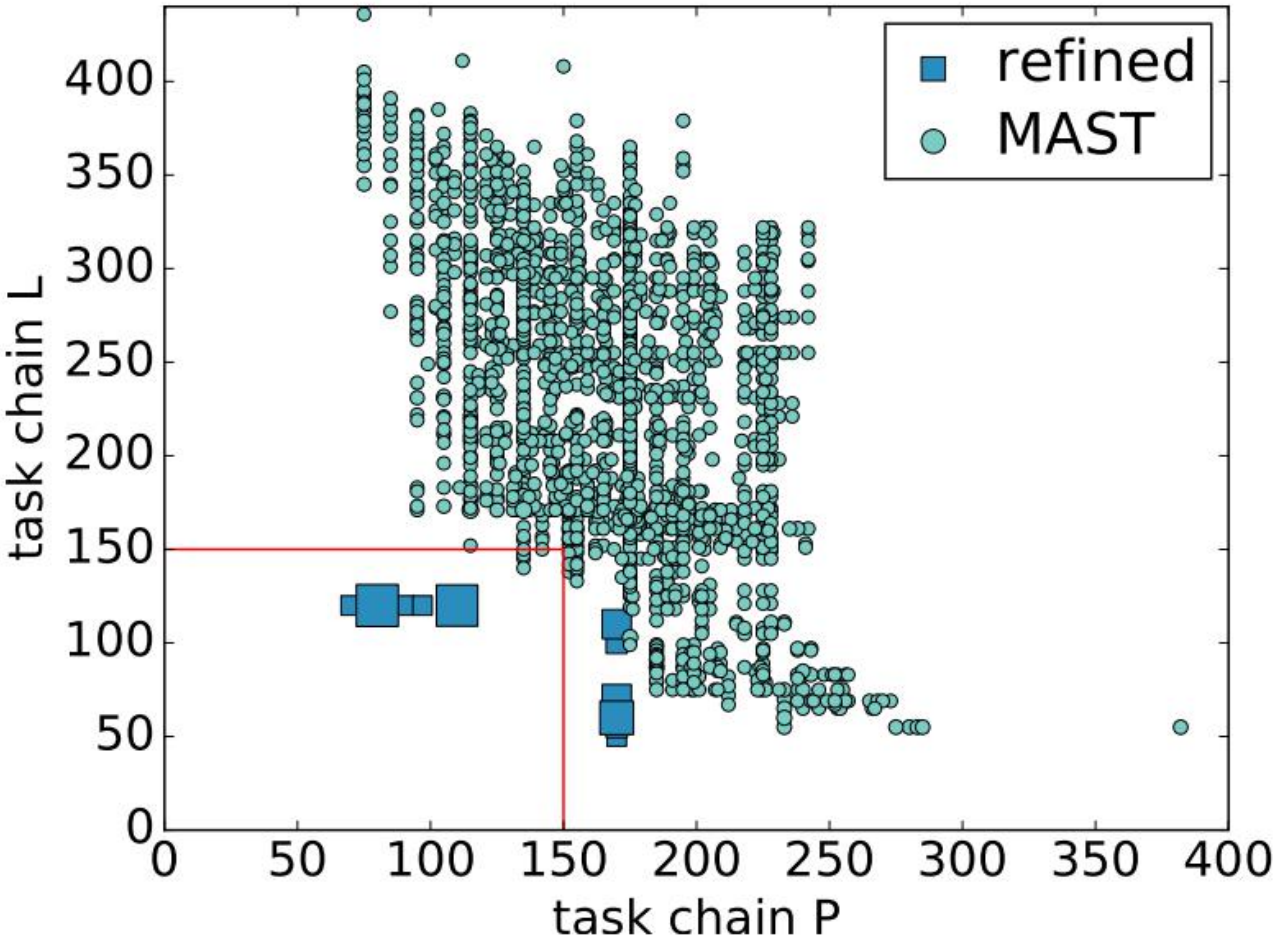


Figure 12. Resulting worst-case latency bounds [in ms] from the refined analysis and MAST. The lines indicate the acceptable solution space.

启示

- 任务链分析方法
- 当前CAN网络的消息调度分析是否也可以应用任务链的分析方法进行改进？？？

THANKS