面向具身智能的多层级嵌入式系统实时调度问题的挑战和模型、算法研究

邓庆绪 教授

CCF 嵌入式系统专委会副主任
辽宁省嵌入式系统重点实验室主任

研究团队：关楠、孙景昊、金曦、吕鸣松、冯智伟、唐月、刘松冉、王祥

www.neu-rtes.org
dengqx@mail.neu.edu.cn

面向具身智能的多层级嵌入式系统实时调度问题的挑战和模型、算法研究

目录

CONTENTS

研究背景

具身智能作为前沿研究领域，正成为全球科技竞争的新焦点。 2025年全国两会中政府工作报告首次将“具身智能”明确列为未来产业培育方向。强调通过技术突破与场景应用推动新质生产力发展，并部署“建立未来产业投入增长机制”以加速产业链成熟。这一政策导向为具身智能的发展提供了强有力的政策支持，凸显了具身智能在国家科技创新体系中的重要地位。

研究背景

- 操作系统是最基本也是最为重要的基础性系统软件，它处于用户应用软件和硬件资源之间，实现对硬件资源的优化管理、协调各应用程序之间的关系，简化用户软件开发的难度，为系统的可靠性和标准化提供保障。
- 回顾操作系统的发展历程（DOS→WINDOS → ANDROID/IOS ），应用驱动的发展过程清晰
- 在泛在AI趋势下，微软、华为等已经开始研发 AI+OS（AI for OS， OS for AI），方便AI应用的开发、部署和应用

研究背景

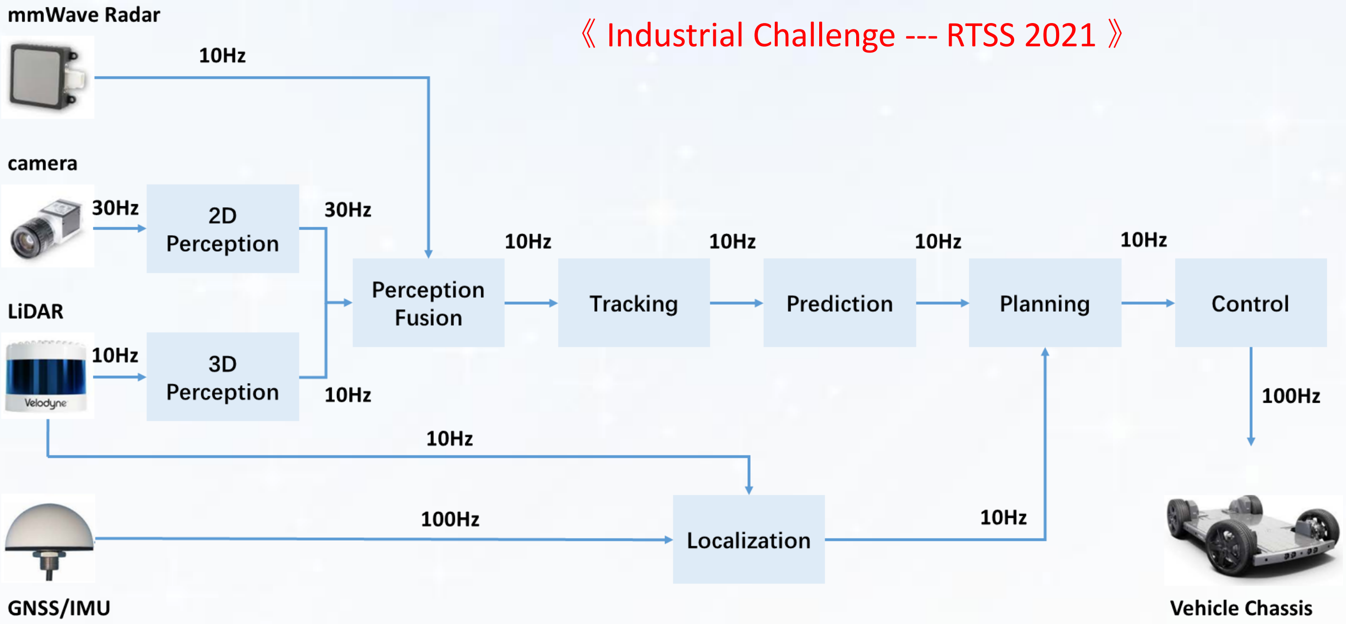
- 开源鸿蒙机器人操作系统 M-Robots OS 正式开源，由开放原子开源基金会孵化、深开鸿牵头发起，旨在以开源共建的方式打造基于开源鸿蒙的统一机器人操作系统M-Robots OS，推动机器人行业生态融合、能力复用、智能协同。
- 指出机器人行业面临两大关键挑战：一方面，传统通用操作系统难以满足实时性要求，而实时系统又缺乏开放性和生态支持；另一方面，不同厂商使用各自封闭的技术栈，造成系统割裂，难以实现高效协同。
- M-Robots OS是全国首个基于开源鸿蒙构建的分布式异构多机协同机器人操作系统，具备多机实时协同、多硬件形态兼容、AI 原生以及丰富API与开发工具链四大核心能力，为行业提供“底层统一、场景多元”的全栈式系统平台

- **多机实时协同：**依托分布式软总线、分布式实时通信与调度能力、高并发低时延（中断响应时延≤1 微秒，任务切换时延≤1 微秒）、多内核混合部署，支持多机高效协作。
- **多硬件形态兼容：**通过弹性部署、软硬件解耦、机器人应用开发中间件模块化及接口标准化机制，适配工业机械臂、AGV、人形机器人和无人机等不同形态、不同大小的设备。
- **AI原生：**内置 AI 框架与算法库，配备智能体开发框架，支持原生 AI 能力构建，推动机器人从“单体智能”迈向“群体智能”。
- **丰富的API与开发工具链：**支持仿真、调试、推理训练等多工具开发模式，提升研发效率。

这个机器人操作系统，强调分布式、内置AI框架与算法库、支持弹性部署，完善了ROS很多不具备的能力。面临的挑战：
（1）连ROS也没有成功应用于工业和自动驾驶等具身智能系统（原因是ROS没有通过ISO 26262功能安全认证）。
（2）进入工业界，要挑战老牌国外公司。像西门子等公司的技术壁垒，自封闭的成熟技术生态（自成熟的基础软件：操作系统、中间件—CODESYS、IDE、仿真环境），国产替代需要一定时间。

具身智能系统的重要特点

- Deployed on an embedded System (Limitation of resources)
- React to open environment adaptively
  - Use multiple sensors to capture events, and then
  - Controllers react to these events in real time
- Most of them are real time system (The correctness of the system depends not only on the logical result of computation, but also on the time at which the results are generated)



Processing graph of an autonomous driving system

面向具身智能的多层级嵌入式系统实时调度问题的挑战和模型、算法研究

目录

CONTENTS

主要挑战

随着具身智能系统向全时域泛在感知、泛在智能演进，传统单层嵌入式架构已经难以满足上述复杂的需求，硬件架构趋向于多种异构处理器集成（通用处理器、GPU/NPU/FPGA等），多操作系统协同调度以完成数据采集、硬件加速、AI算力支撑、实时控制等复杂任务。

面临的系统性瓶颈：在垂直资源协同维度，硬件加速器、操作系统与应用服务的割裂式调度机制难以满足AI推理（百毫秒级）、控制决策（毫秒级）、信号处理（微秒级）等多时间域任务的协同需求；在时序保障维度，深度耦合的AI推理-实时控制任务图因其复杂依赖关系导致端到端时延建模、分析失效，异构计算单元（CPU/GPU/NPU）与多级软件栈（RTOS/中间件/应用层）的跨层级资源竞争更引发时间行为失配，直接影响关键场景的可靠性，如自动驾驶毫秒级决策响应等。

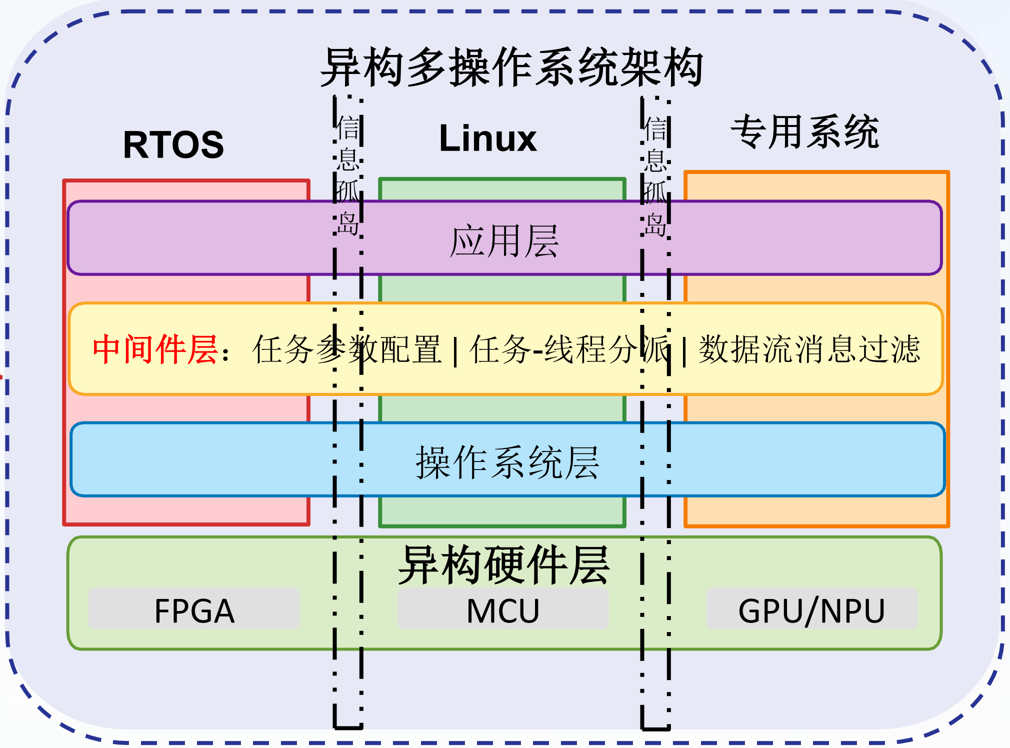
挑战一 多操作系统信息孤岛不易破除

核心问题

- 跨系统任务状态可见性缺失
- 资源使用数据异步性加剧
- 架构设计理念差异
- 调度机制冲突

技术根源

- 调度指令与资源数据语义鸿沟
- 安全机制对状态采集刚性约束
- 实时性要求对通信带宽压制



挑战二 多层级垂直维度资源动态整合难

- 各层调度机制相互独立
- 缺乏统一实时监控和闭环反馈
- 资源分配不均与优先级错配
- 系统整体实时性受损

挑战三 任务图时空耦合建模与调度优化难

- 多时间域任务深度融合
- 执行路径动态分叉、嵌套反馈
- 端到端时延累积效应非叠加性
- 异构硬件多模态时间行为差异

主要研究问题1：多层级嵌入式系统功能与时间行为建模

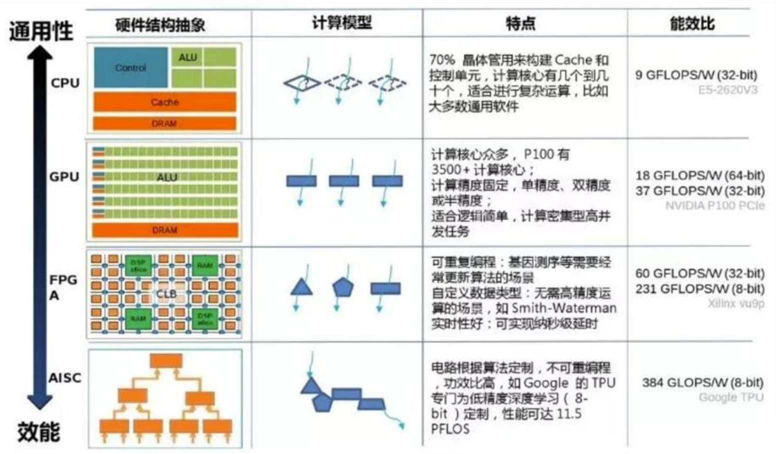
在资源受限与边缘环境动态不确定的双重约束下，传统静态建模方法因缺乏对系统运行时复杂交互效应的动态表征能力而面临显著局限性。其主要体现在两个方面：

（1）硬件资源的动态变化（如算力波动、内存碎片化）与软件任务的非线性行为（如并发竞争、事件触发逻辑）之间形成的时空耦合效应，导致系统功能模块间的资源竞争关系呈现出非稳态特征。

（2）任务执行路径的随机分叉、中断嵌套与资源抢占行为，使得系统时间维度涌现出延迟累积现象，传统静态模型难以刻画此类复杂时序特征。

这两类问题不仅造成系统时间可预测性衰减，更使得基于静态模型的优化决策存在显著偏差风险。因此，如何基于运行时监测数据驱动，构建融合功能抽象与时间行为表征的多层级嵌入式系统模型，是该类系统需要解决的第一个问题。

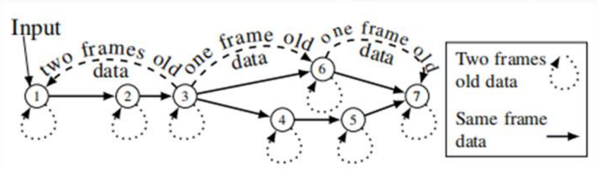
计算资源的异构导致计算模型无法统一、执行效率难以衡量



映射

?

AI任务包含大量关系复杂的子任务，且子任务可能由不同厂商独立开发，无法统一、难以协调



主要研究问题问题2：跨层级耦合干扰下实时性分析理论问题

实时性分析理论面临的核心挑战来源于跨层级交互机制的深度耦合特性，其耦合效应严重限制了系统实时性的精准分析。其复杂性在架构维度和时序维度呈现**双重特征**：

（1）在**纵向架构层面**，中间件服务与底层操作系统调度策略形成非线性交互依赖，难以通过传统线性叠加方法进行显式表达；

（2）在**横向时序层面**，硬件资源竞争、中断抢占等动态随机事件会引发干扰强度的时变性波动，并与系统层级间的耦合效应相互叠加，导致传统孤立层级分析方法会引入显著误差。

现有研究多基于单层抽象或平均负载假设，难以支撑高精度实时性理论分析。因此，如何突破层级边界约束，构建能够精准表征跨层级耦合干扰的实时性分析理论体系，是该类系统需解决的第二个问题。

主要研究问题3：异构资源约束下多操作系统实时协同调度问题

在多操作系统协同调度研究中，如何构建跨系统协同实时调度架构，成为解决异构平台间实时任务冲突与全局资源优化的核心挑战。

（1）一方面，多系统共享异构硬件时，原生调度策略的差异性会引发策略级冲突，导致跨系统任务执行链出现非确定性延迟；

（2）另一方面，异构计算单元在指令集、存储层次及接口协议等方面的本质差异，使得基于单系统优化的资源分配模型难以直接映射到多系统协同场景，加剧了任务同步与数据通信的时序失序风险。

现有方法多局限于单操作系统内的调度优化，或依赖粗粒度资源虚拟化实现跨系统协同，难以同时满足低延迟、高可靠性及资源高效利用的需求。因此，如何设计跨系统协同调度算法，深度融合时间同步、通信优化、状态感知与决策机制以及多层级协同机制，在异构资源约束下实现多操作系统的实时性保障与资源效率最大化，是该类系统需解决的第三个问题。

主要研究内容

- 基于监测驱动的异构嵌入式系统建模及架构研究
  - （1）嵌入式边缘端多层级系统架构研究
  - （2）面向多层级嵌入式系统的轻量化监测方法研究
  - （3）基于监测的嵌入式系统软/硬件建模研究
- 面向多层级嵌入式系统架构的实时性分析理论研究
  - (1) 跨层级干扰耦合量化模型构建方法
  - (2) 混合关键性任务可调度性分析方法
  - (3) 异构嵌入式系统端到端延迟分析方法
- 多操作系统实时保障与协同调度研究
  - （1）面向混合关键性任务的最优实时调度算法研究
  - （2）多操作系统时间同步与通信机制研究
  - （3）多操作系统间的协同调度框架研究
- 面向中间件的隔离、任务分配及协同优化研究
  - （1）硬件透明映射的轻量化资源隔离中间件构建
  - （2）保障端到端时延最优的任务分配方法研究
  - （3）多层级协同闭环优化调度框架研究

面向具身智能的多层级嵌入式系统实时调度问题的挑战和模型、算法研究

目录

CONTENTS

研究团队——实时嵌入式系统团队

研究方向：实时调度算法、最差执行时间分析、实时操作系统、功耗相关调度算法等，团队负责人邓庆绪教授

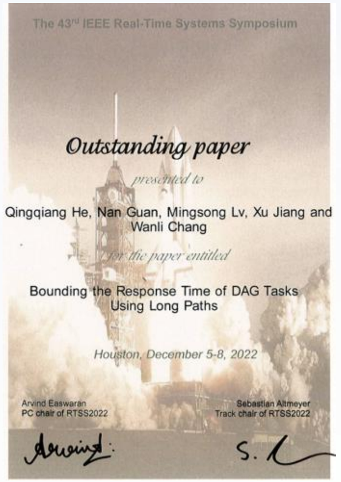
团队现成员包括：吕鸣松、唐月、王祥、何青强、刘松冉、冯智伟……

早期成员：关楠、吕鸣松、孙景昊、姜徐、孔繁鑫、陈刚、张天宇……

聚是一团火，散是满天星

已毕业并保持紧密合作：关楠、孔繁鑫、金曦、孙景昊、张天宇、孔繁鑫……

该研究团队是目前国际上实时嵌入式系统研究领域非常活跃的一个研究队伍，团队做出了一批国际领先的重要理论成果，培养的博士曾获得CCF优秀博士论文，团队所发表的论文多次获得实时系统排名第一的国际顶级会议RTSS最佳论文奖、最佳论文提名，以及RTAS、ECRTS、 DATE、EMSOFT等国际知名会议的最佳论文和最佳论文提名。



近三年围绕具身智能实时嵌入式系统建模、时间分析和调度部分研究成果

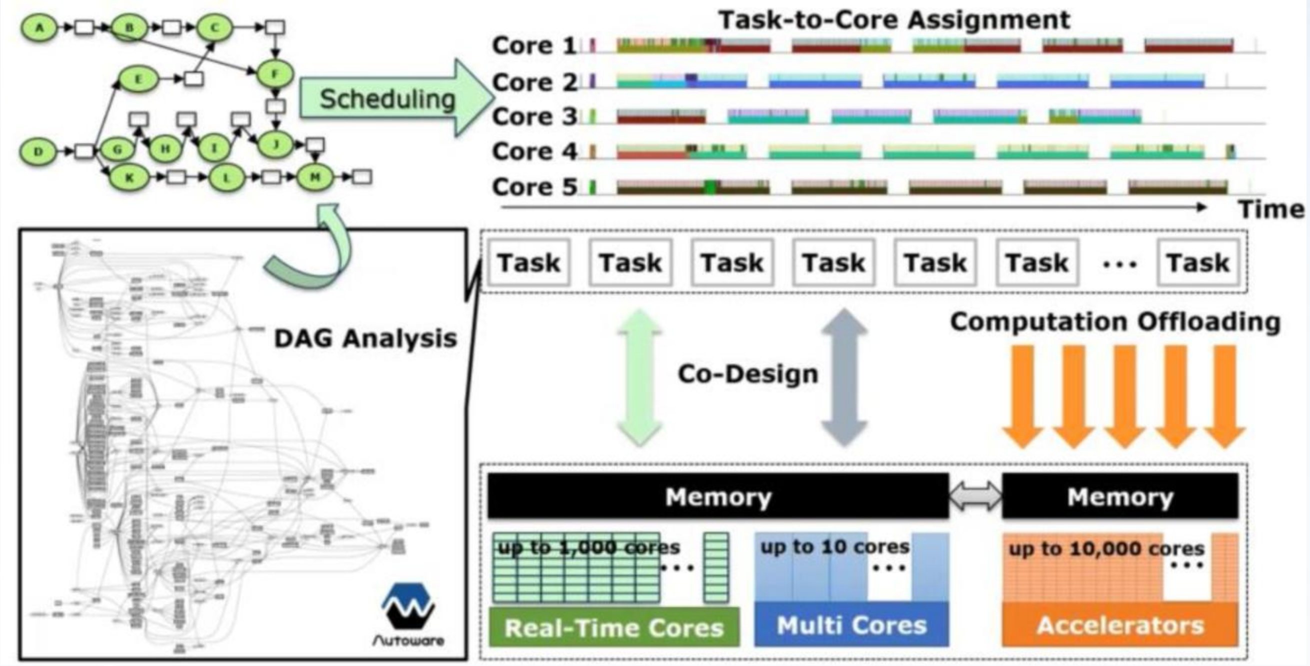
- Jointly Ensuring Timing Disparity and End-to-End Latency Constraints in Hybrid DAGs. RTAS 2025
- Priority Optimization for Autonomous Driving Systems to Meet End-to-End Latency Constraints. RTSS **2024**
- Real-Time Scheduling of Conditional DAG Tasks with Intra-Task Priority Assignment. **TCAD(2023)**
- Real-Time Scheduling of Autonomous Driving System with Guaranteed Timing Correctness. **RTAS (2023)**
- ROSGM: A Real-Time GPU Management Framework with Plug-In Policies for ROS 2. **RTAS (2023)**
- On the Degree of Parallelism in Real-Time Scheduling of DAG Tasks. **DATE(2023)**
- SEAM: An Optimal Message Synchronizer in ROS with Well-Bounded Time Disparity**. RTSS(2023)**
- Optimizing End-to-End Latency of Sporadic Cause-Effect Chains Using Priority Inheritance. **RTSS(2023)** （因果关系任务链）
- Reaction Time Analysis of Event-Triggered Processing Chains with Data Refreshing. **DAC(2023)** （因果关系任务链）
- Comparing Communication Paradigms in Cause-Effect Chains, **IEEE Transactions on Computers TC(2023)**）（因果关系任务链）
- Mixed-Criticality Scheduling of Energy-Harvesting Systems. **RTSS(2022)**
- Response Time Analysis for Energy-Harvesting Mixed-Criticality Systems . DATE(2022)
- Online Re-routing for Fault-Resilient Scheduling of Time-Sensitive Networking. EMSOFT-TCAD （2022）
- Towards Minimum WCRT Bound for DAG Tasks Under Prioritized List Scheduling Algorithms. TCAD（ 2022）
- Response time analysis of parallel tasks on accelerator-based heterogeneous platforms. J. Syst. Archit, 2022
- Computing exact WCRT for typed DAG tasks on heterogeneous multi-core processors. J. Syst. Archit. 2022 （响应时间分析）
- Bounding the Response Time of DAG Tasks Using Long Paths. RTSS(2022)

特别是针对ROS2的多个研究成果连续发表在本领域顶级国际会议RTSS上

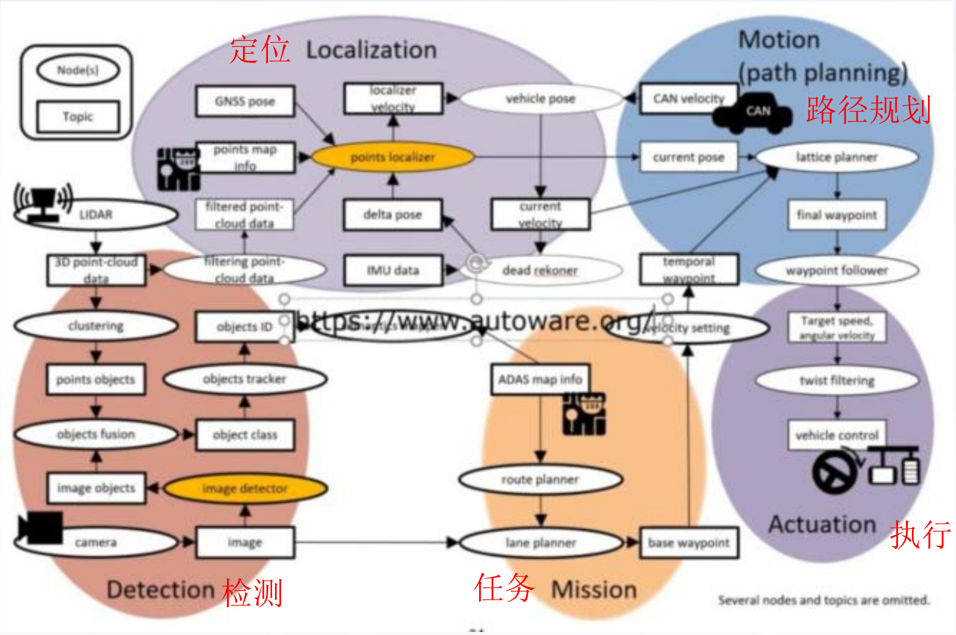
针对多层级嵌入式系统功能与时间行为建模研究

基于监测驱动的异构嵌入式系统建模及架构方面的主要难点在于当前工业互联网场景嵌入式系统任务图时空依赖高度复杂、异构资源难协调。

图式任务　　任务调度



任务分析　　实时核　　普通多核　　加速器(GPUs, FPGAs)

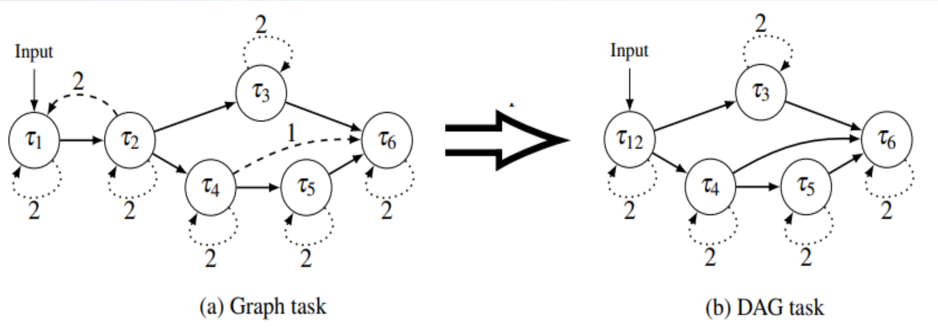


整合独立开发的组件

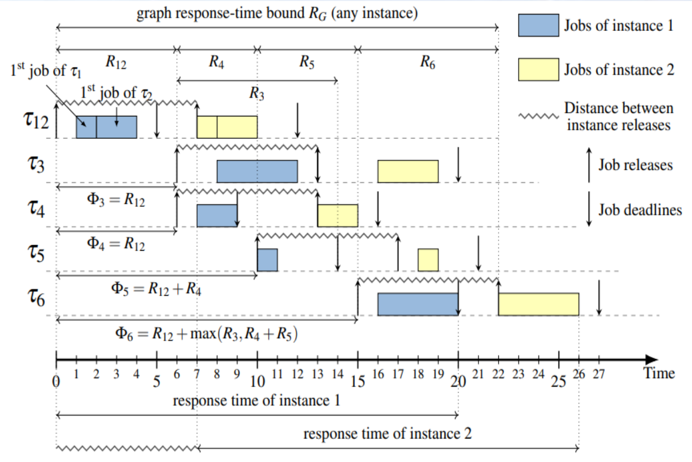
安全关键系统中, 基于图的AI应用存在非常复杂的数据依赖。现有响应时间分析技术在新型的异构硬件(异构多核+加速器)上处理图式任务存在限制, 整合这些单独开发的组件十分困难，如ARINC 653要求不同组件在时间上的完全隔离

针对多层级嵌入式系统功能与时间行为建模研究

采用的方法：



1.将图式任务转换为DAG任务



2.将DAG转换为普通任务

存在问题：

问题1：HAC（硬件加速器）访问通常是不可抢占的，可能会超出时间片边界。

问题2：当硬件资源因时间分区而部分可用时，现有的响应时间分析不适用。

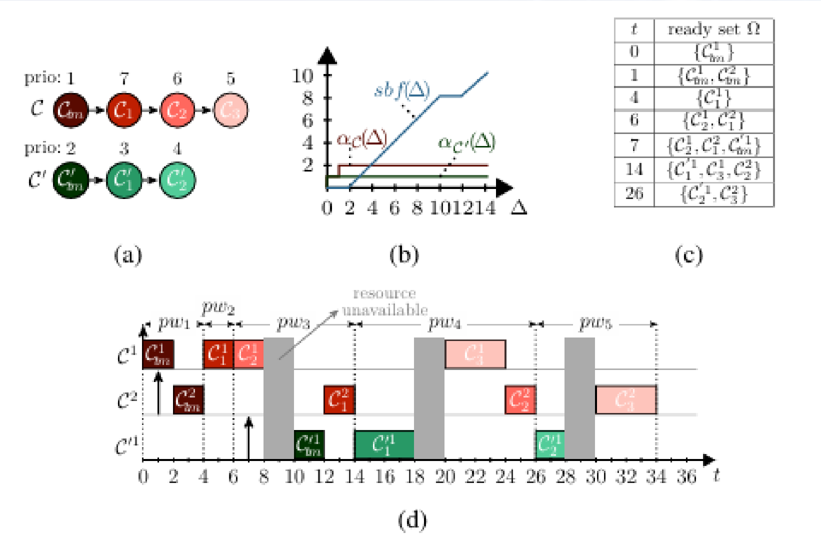
问题3：在AI用例中，某些HAC（特别是GPU）可能高度争用，导致阻塞界限（被分析转化为CPU执行时间）过大。

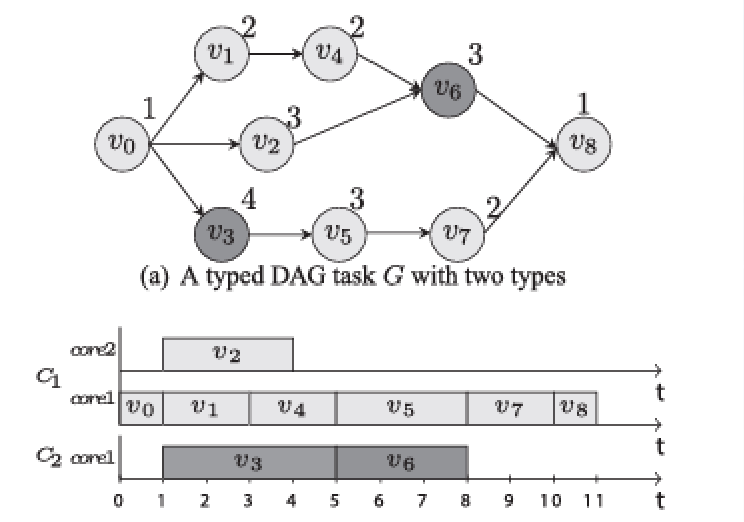
问题4：响应时间界限随着释放偏移的增加而增加，当依赖关系形成长路径时，会被时间分片进一步加剧，导致响应时间界限过大

把经典图理论引入实时分析领域，提出的基于长路径的方法，成果发表在顶级国际会议RTSS2020、RTSS2022并获得最佳论文奖。

针对多操作系统实时保障与协同调度的研究

- 团队研究了ROS2执行器上任务链的实时调度问题，并提出了新的响应时间分析技术，成果发表在：RTSS
- 提出基于虚拟处理器的实时任务调度方法，这对于解决多核平台上并行调度实时任务的问题具有重要意义，成果发表在：TPDS
- 提出统一阻塞分析框架，为分析使用自旋锁保护资源的并行实时任务提供了理论支持，这对于构建高可靠的多操作系统协同运行体系至关重要，成果发表在：TC
- 团队解决了ROS2中针对任务链延迟更新调度规则显著不同于现有实时调度理论问题带来的实时性无法保证问题，成果发表在：RTSS2020、TC、DATE2021
- 解决了ROS2中针对任务结构复杂、多条任务链交叉实时性无法保证问题，成果发表在：RTAS2023
- 针对ROS2中任务间通信模式对端到端实时性能有重要影响，我们给出了端到端通信的最优方案，成果发表在RTSS2023
- 针对ROS2中消息同步机制每次迭代追求局部最优，造成效率低、成功率低问题，我们提出了SEAM机制，成果发表在：RTSS2023并获得最佳论文奖

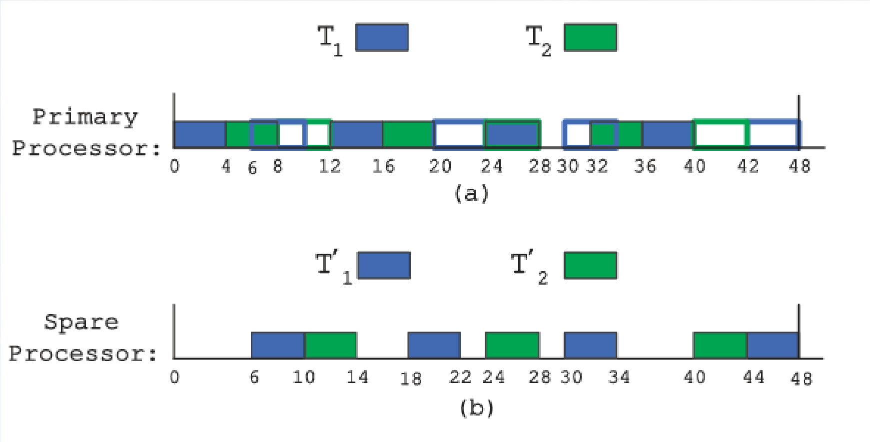




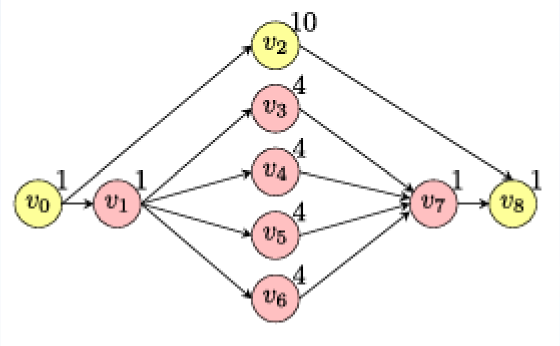
(a) A typed DAG task G with two types

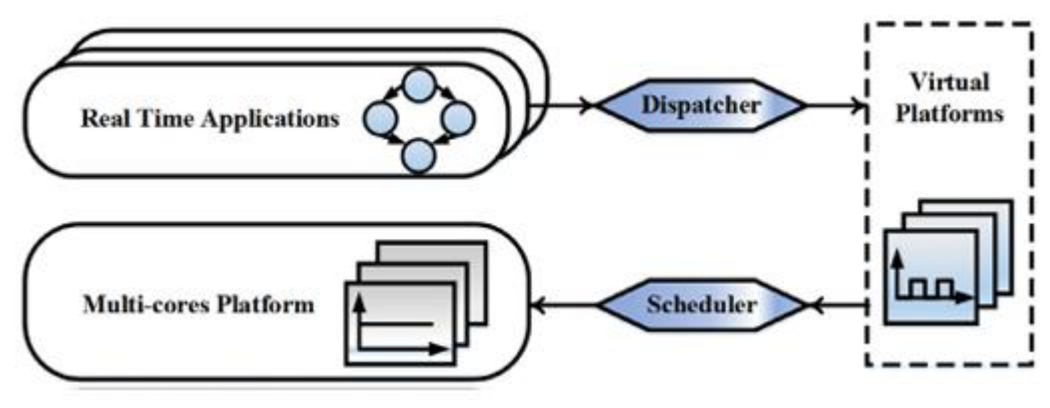
(b) An execution sequence under work-conserving algorithm

针对多层级嵌入式系统架构中实现有效的资源隔离、任务分配和协同调度的研究



- 团队提出了新的调度策略和任务划分方法，提高了能效并保障了系统故障容忍能力，成果发表在：TECS
- 提出了基于优先级列表调度（PLS）的新方法，适用于多核处理器上的并行任务调度，成果发表在：TCAD
- 提出用于在异构多核平台上调度带有类型约束的DAG任务的联邦调度方法，提出了新的核心分配方法，成果发表在：JSA
- 提出了层次化调度框架，结合全局调度和联邦调度的优势，这对于多层级协同闭环优化调度框架的研究具有重要参考价值，成果发表在：JSA

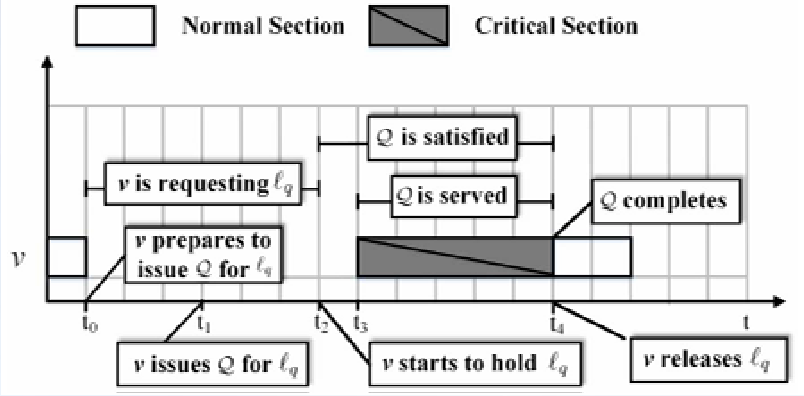




面向实时系统可调度性和端到端时延分析方面的研究

基于监测驱动的异构嵌入式系统建模及架构方面的主要难点在于当前具身智能场景嵌入式系统任务图时空依赖高度复杂、异构资源难协调。

- 团队较早开展针对多核处理器可调度性分析和调度策略进行研究，成果发表在：计算机学报
- 提出新的时间需求界限分析方法，以分析系统可调度性，成果发表在：RTSS
- 研究了不同通信范式对因果效应链的影响，并分析端到端时延，成果发表在：TC
- 提出了基于任务的关键路径进行任务可调度性分析，成果发表在：TCAD
- 提出基于 CAF 的线性时间可调度性测试方法，灵活评估系统可调度性，成果发表在：TCAD
- 考虑多重因素的影响并结合最新的单DAG任务分析方法对多个 DAG 任务的可调度性进行研究，成果发表在：计算机研究与发展
- 分析了任务的可调度性和任务链的端到端延迟约束，成果发表在：RTAS
- 研究了解决实时系统中支持动态缓存分配的抢占式全局 EDF 调度算法的可调度性分析问题，成果发表在：JSA
- 基于全局固定优先级（G-FP）调度策略提出轻任务的可调度性分析方法，成果发表在：JSA



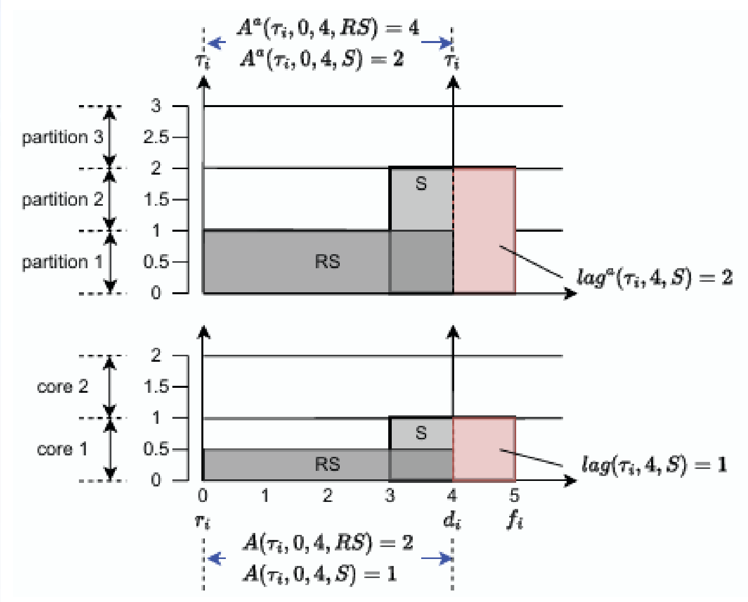


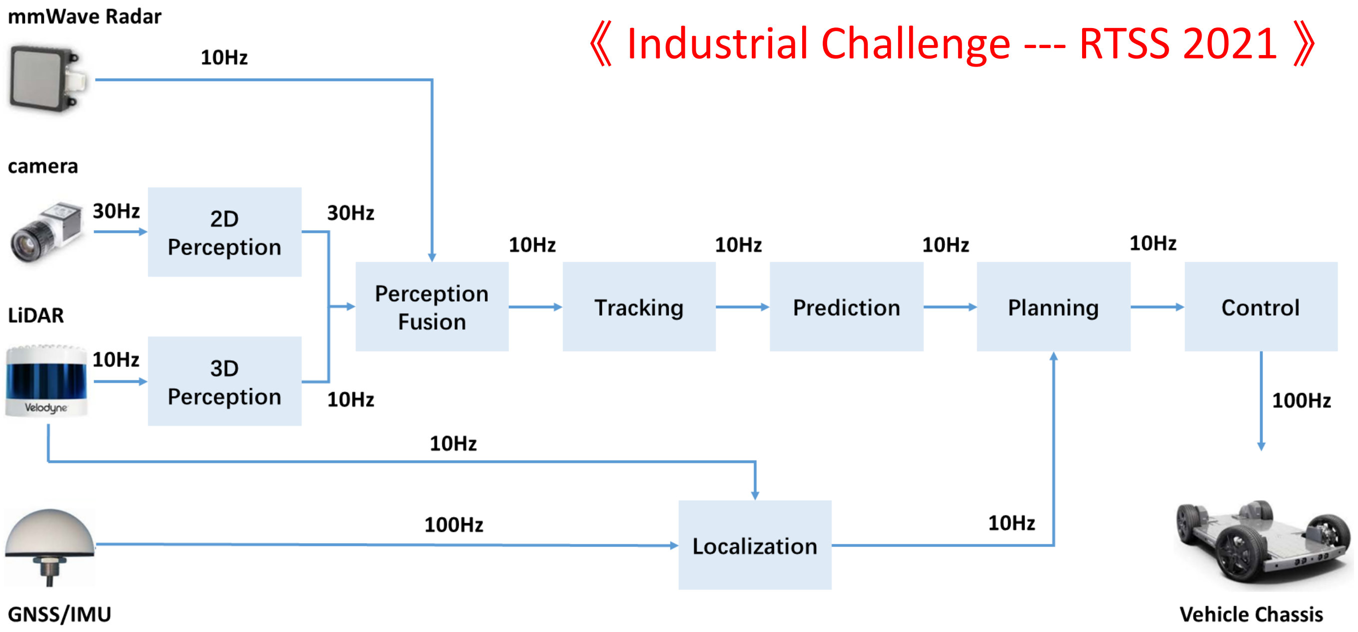
Fig. 3: An example of lag and cache lag

SEAM: An Optimal Message Synchronizer in ROS with Well-Bounded Time Disparity

Jinghao Sun[1], Tianyi Wang[1], Yang Li[1], Nan Guan [2], Zhishan Guo[3], Deng Qingxu[4], Tan Guozhen[1]

Common Features of Autonomous Machine

- A common feature of autonomous machine is
  - The ability to sense the surroundings and make decisions based on that data without human intervention



Processing graph of an autonomous driving system

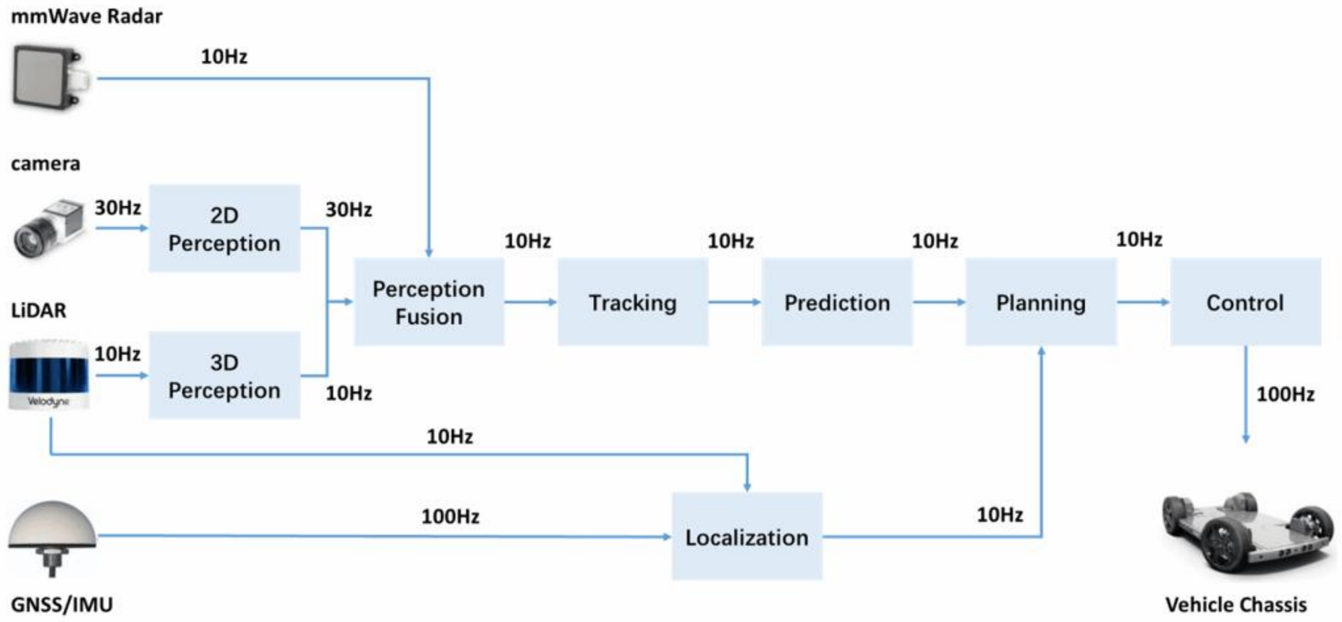
Timing Constraints in Autonomous Machines

- End-to-End Latency Guarantees

- Timing Consistency in Data Fusion

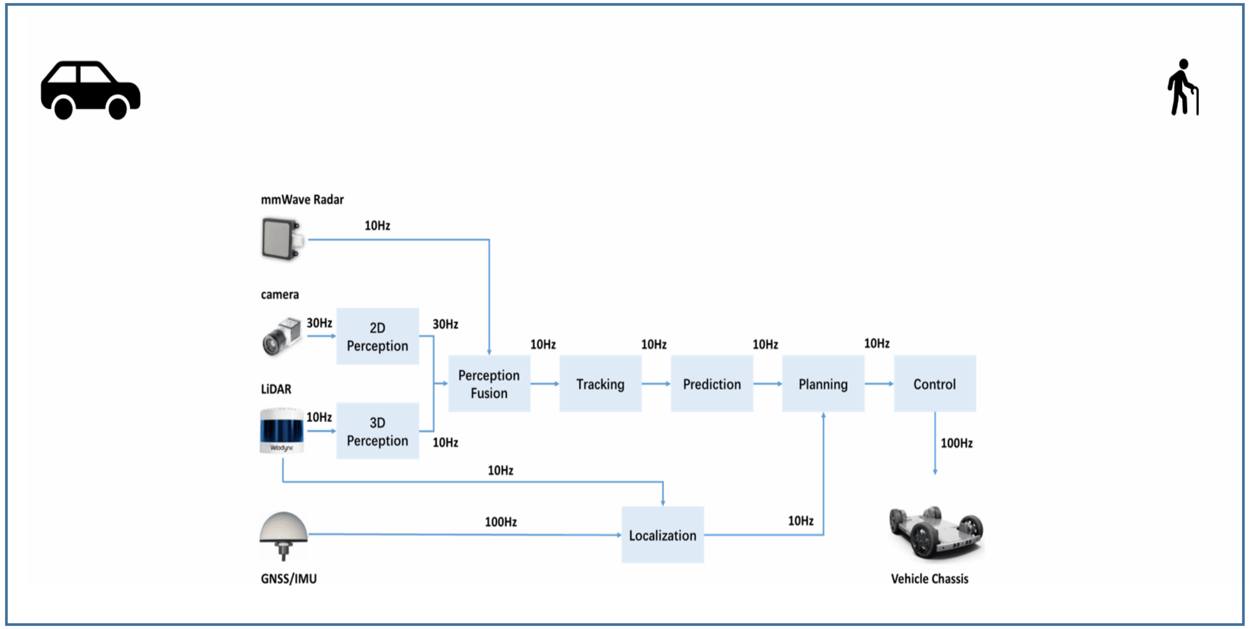
End-to-End Latency Guarantees

- Data propagation from sensors to controllers must satisfy the end-to-end latency constraint

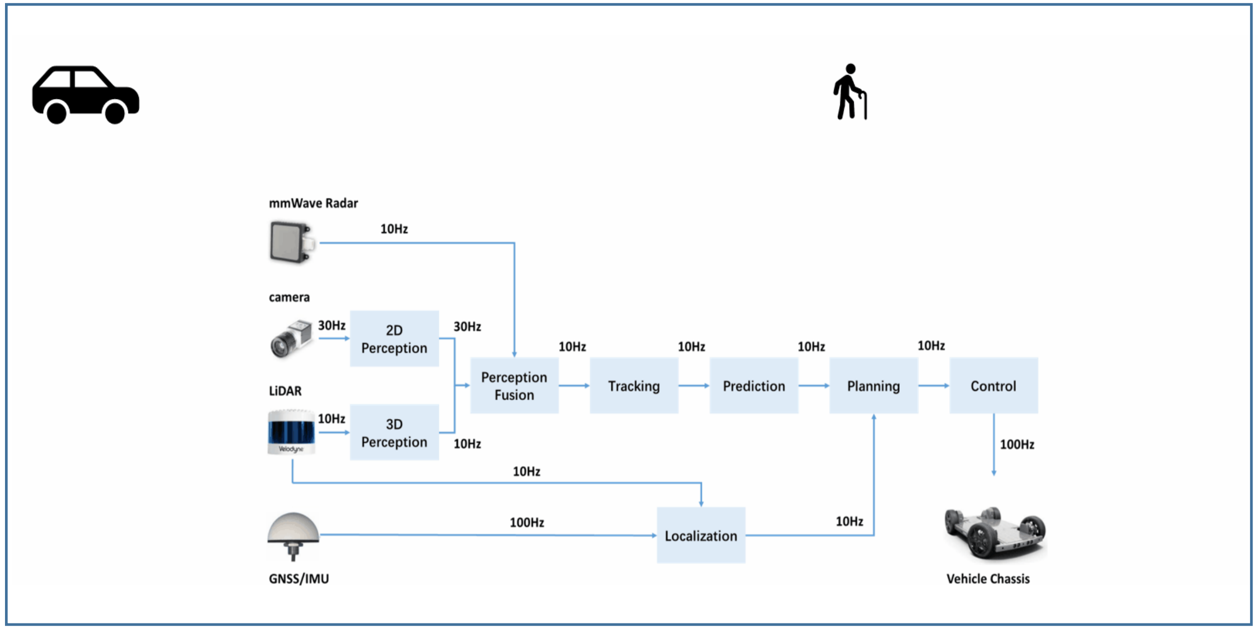


End-to-End Latency Guarantees

- Data propagation from sensors to controllers must satisfy the end-to-end latency constraint
- Large latency→ Slow response to environmental changes



Low latency, and quickly response to changes in the surroundings



Large latency, slow response, leading to an accident

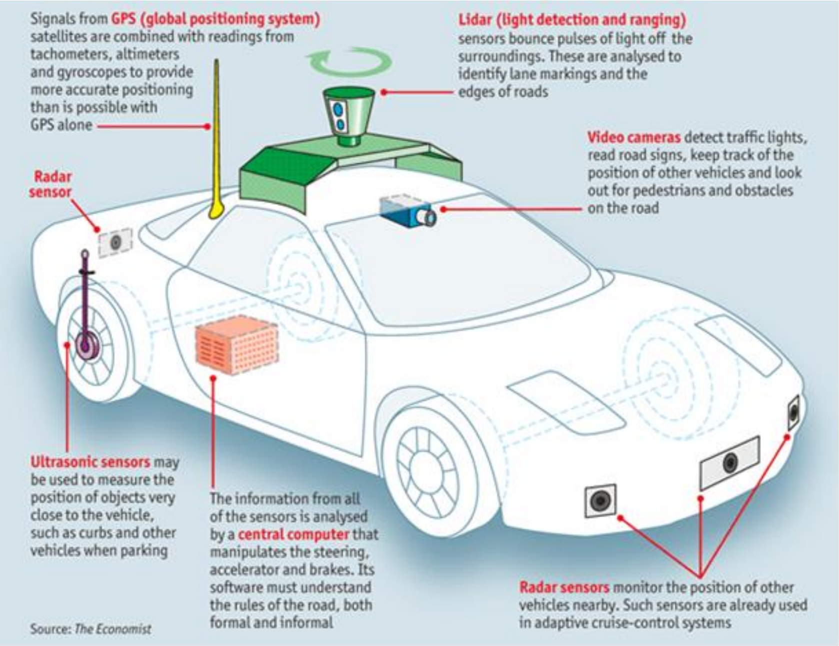
Timing Constraints in Autonomous Machines

- End-to-End Latency Guarantees

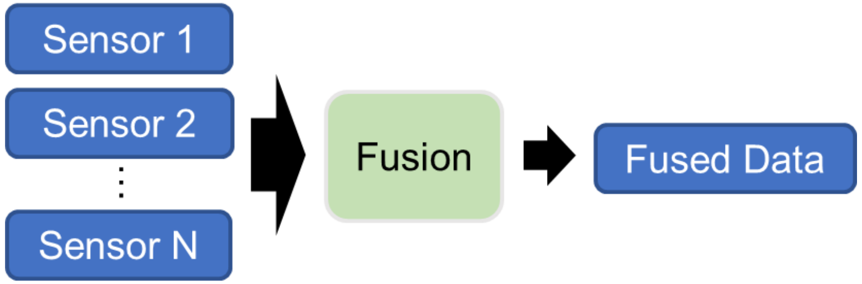
- Timing Consistency in Data Fusion

Timing Consistency in Data Fusion

- Sensor data from disparate sources can be safely fused only if their timestamps closely align with each other



Picture from The Economist, "How does a self-driving car work?"

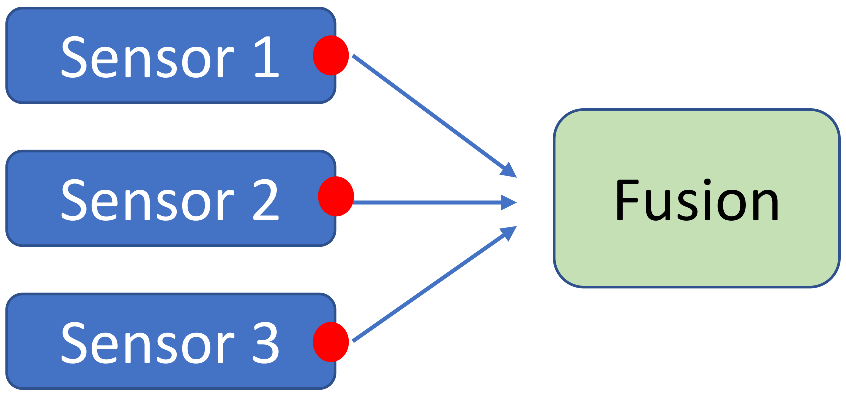


Timing Consistency in Data Fusion

- Sensor data from disparate sources can be safely fused only if their timestamps closely align with each other

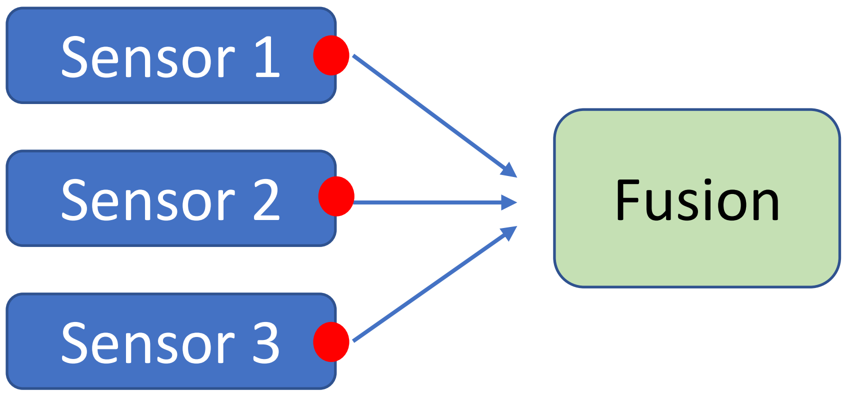
Ideal Situation

Data from different sensors are sampled at the same time



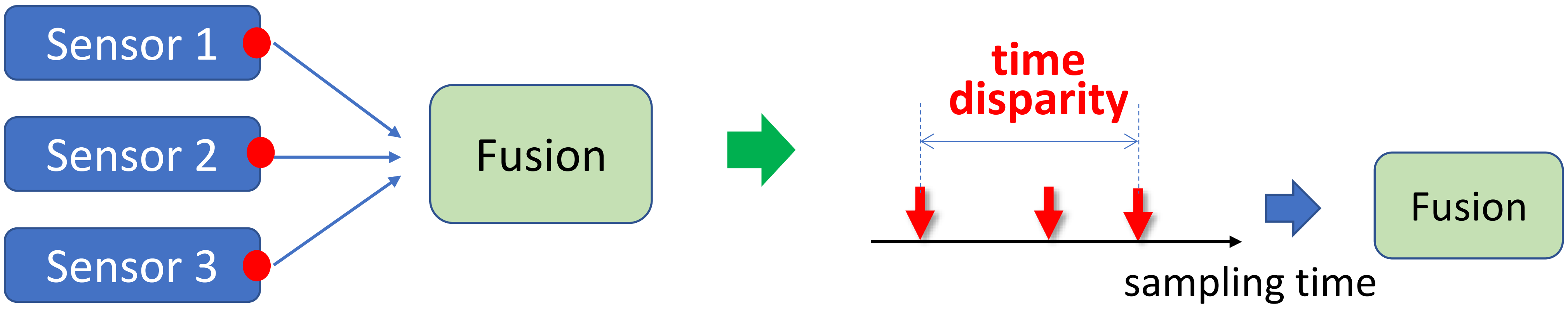
Real-World Situation

Fused data from different sensors have different timestamps



Timing Consistency in Data Fusion

- Sensor data from disparate sources can be safely fused only if their timestamps closely align with each other
  - Time disparity: difference of their sampling time



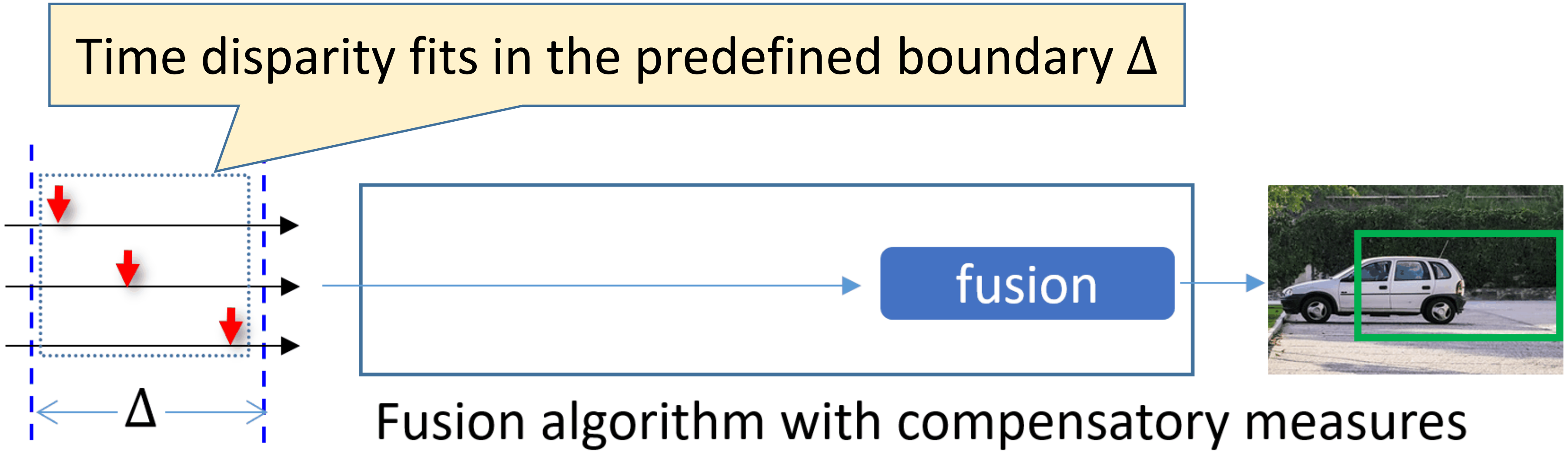
Innovation of This Work

- More deeply rooted in practical applications

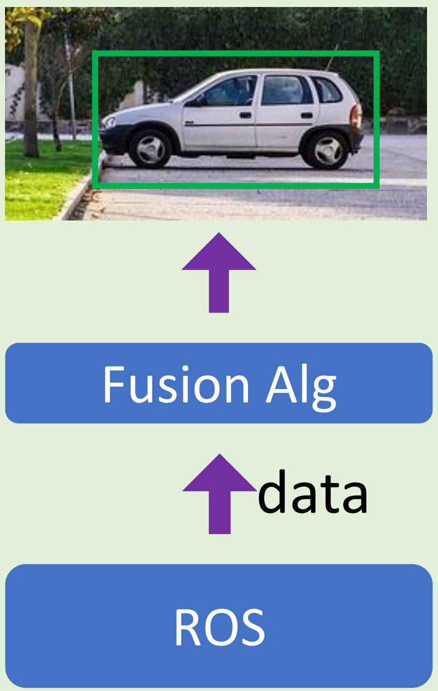
- Try to design more effective fusion mechanisms

Rooted in Practical Applications

- Fusion algorithms
  - Do not enforce the fused data to be sampled at the same time
  - Instead, fused data only needs to maintain a reasonable time disparity



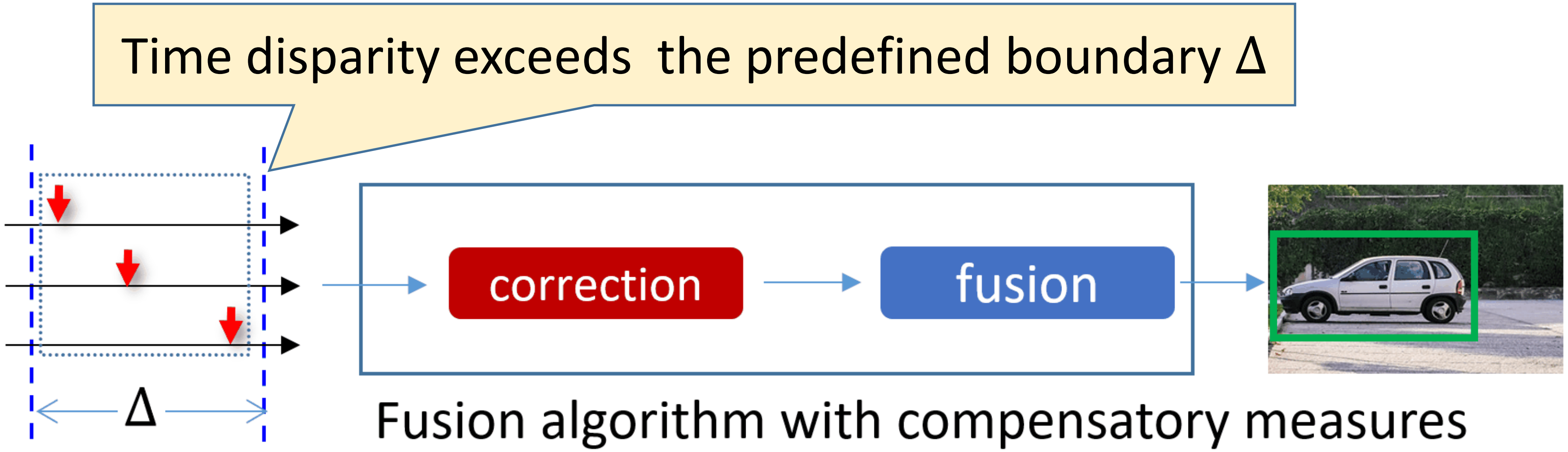
Fusion algorithm with compensatory measures



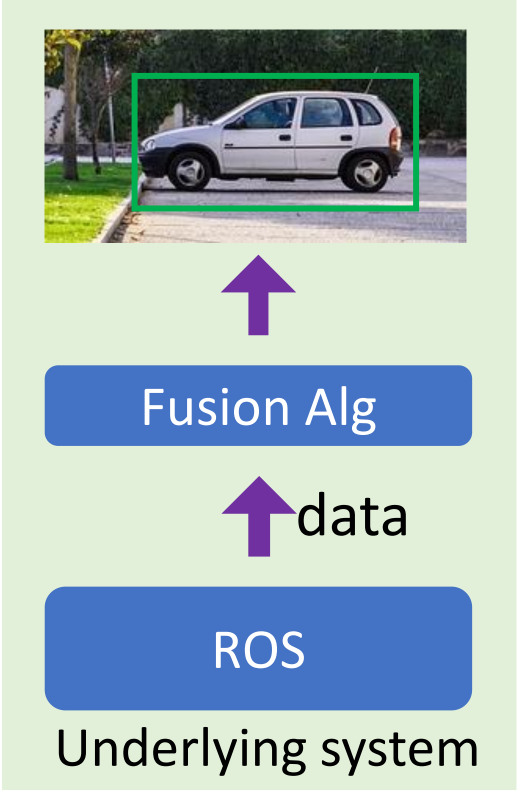
Underlying system

Rooted in Practical Applications

- Fusion algorithms
  - Do not enforce the fused data to be sampled at the same time
  - Instead, fused data only needs to maintain a reasonable time disparity

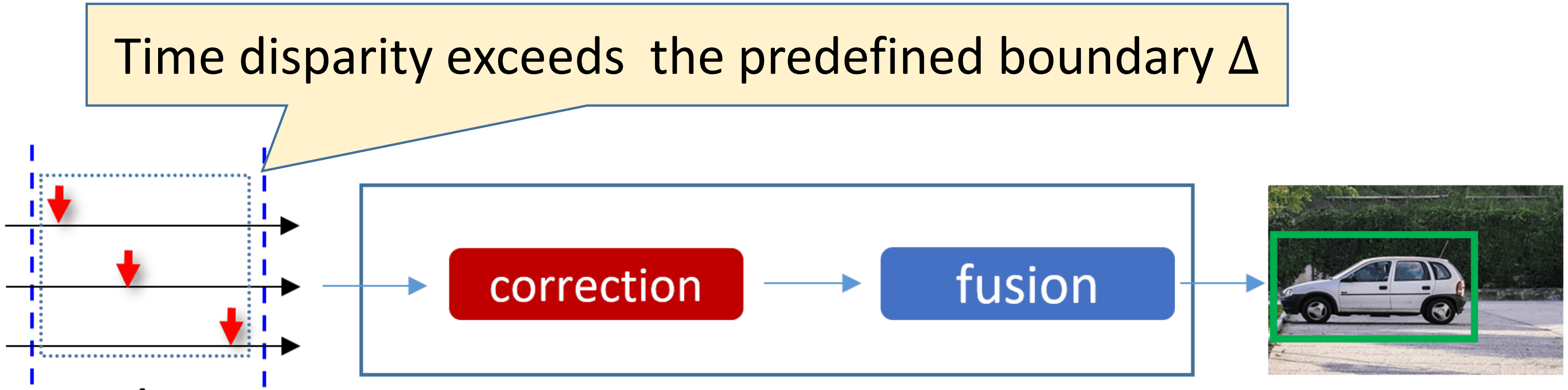


Fusion algorithm with compensatory measures

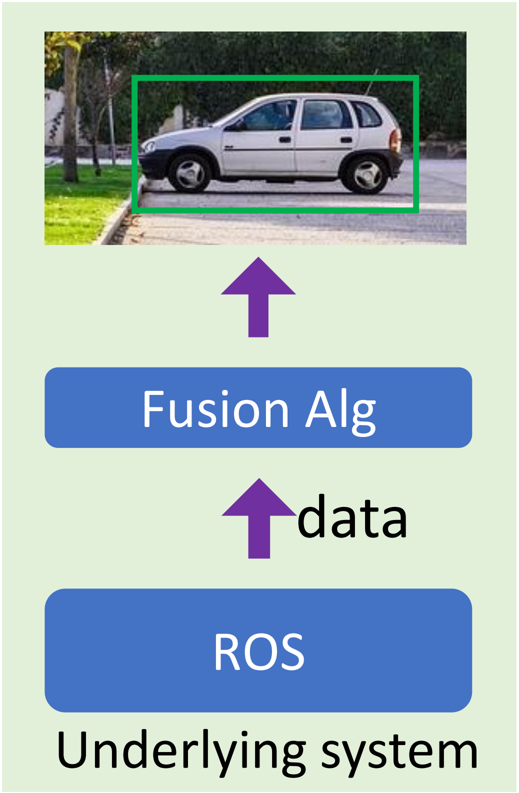


Rooted in Practical Applications

- Fusion algorithms
  - Do not enforce the fused data to be sampled at the same time
  - Instead, fused data only needs to maintain a reasonable time disparity



Fusion algorithm with compensatory measures



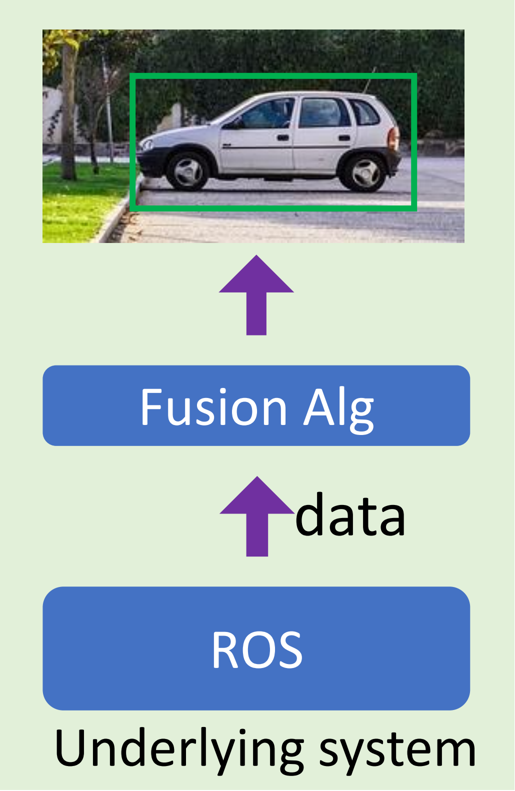
Basic Requirement of Fusion Algorithms

Data is fused only if the time disparity falls in an acceptable boundary.

Rooted in Practical Applications

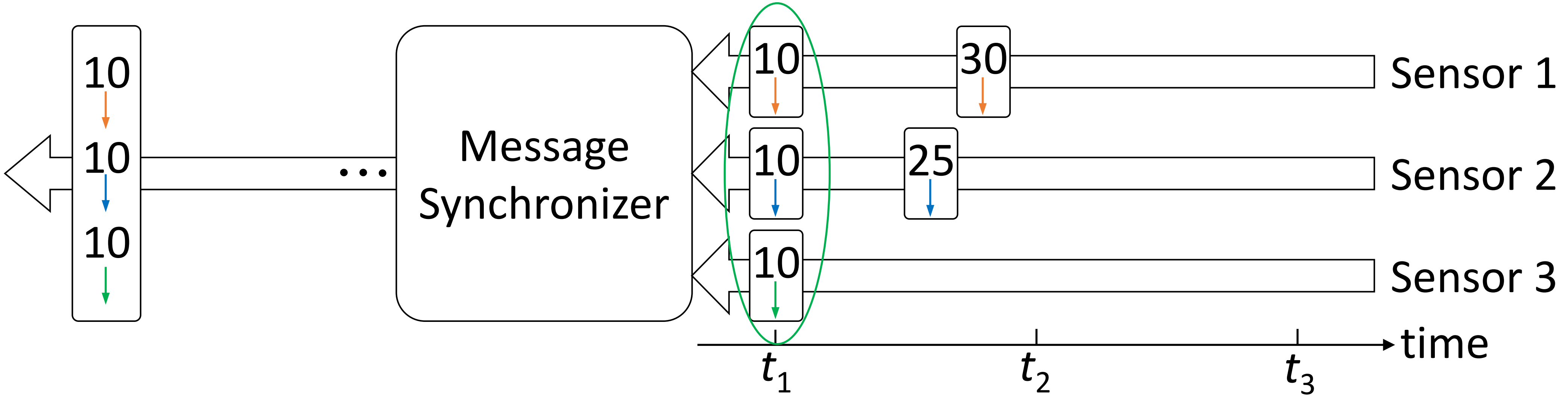
- Fusion algorithms
  - Do not enforce the fused data to be sampled at the same time
  - Instead, fused data only needs to maintain a reasonable time disparity
- Orient to basic requirements of fusion algorithms
  - Pursue small time disparity
  - but NOT the smallest time disparity

Basic Requirement of Fusion Algorithms
Data is fused only if the time disparity falls **in an acceptable boundary**.



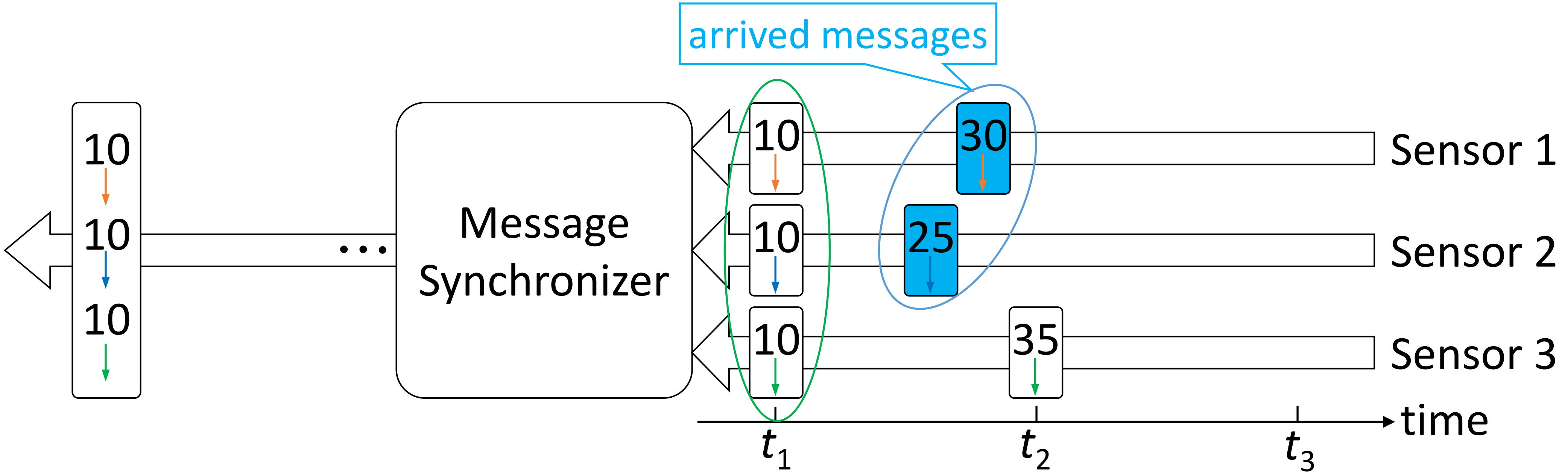
What happens when Pursuing the Smallest Disparity?

- Traditional methods in ROS
  - Called the ApproximateTime Algorithm
  - A sophisticated message synchronization policy
    - Predict the arrival time of messages
    - Wait for incoming messages to pursue the smallest time disparity



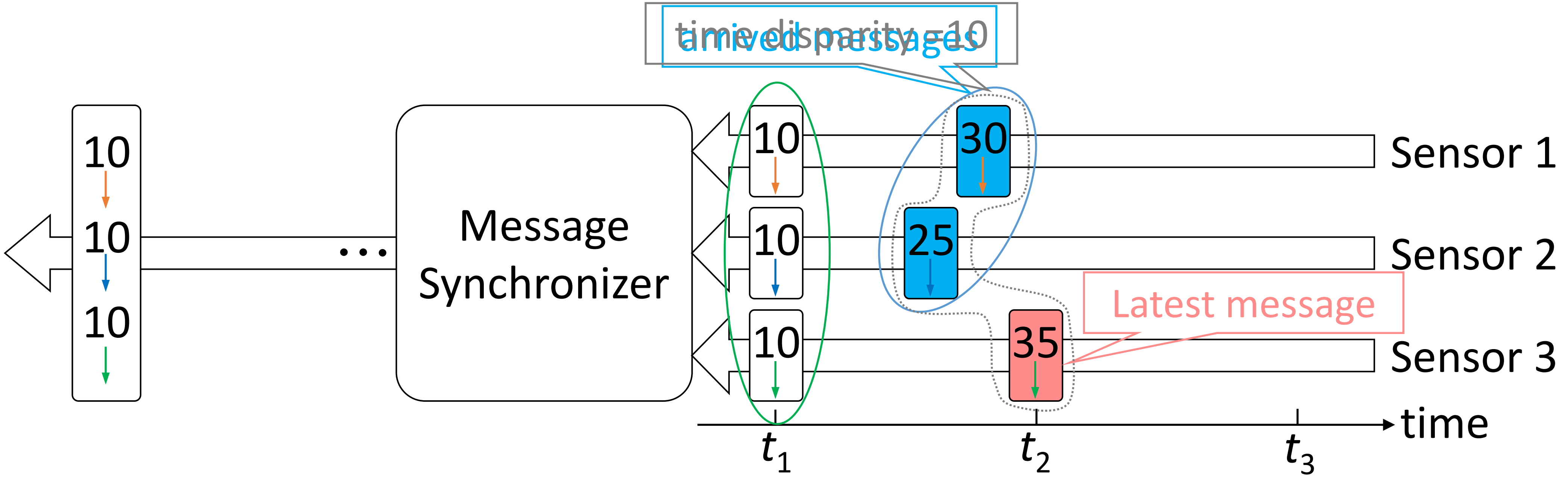
What happens when Pursuing the Smallest Disparity?

- Traditional methods in ROS
  - Called the ApproximateTime Algorithm
  - A sophisticated message synchronization policy
    - Predict the arrival time of messages
    - Wait for incoming messages to pursue the smallest time disparity



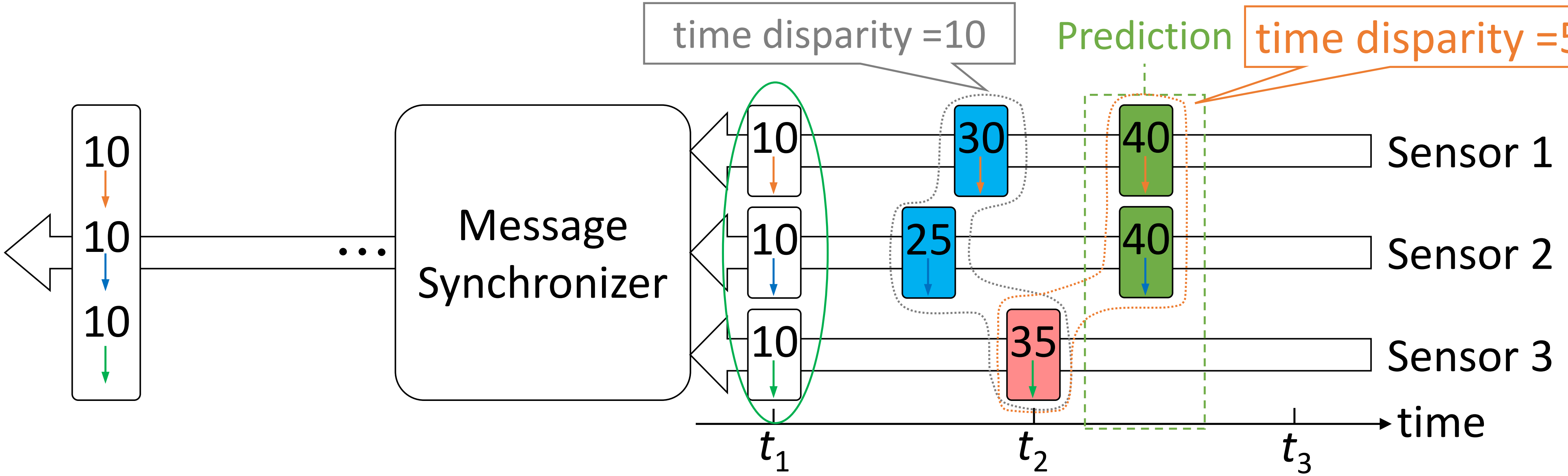
What happens when Pursuing the Smallest Disparity?

- Traditional methods in ROS
  - Called the ApproximateTime Algorithm
  - A sophisticated message synchronization policy
    - Predict the arrival time of messages
    - Wait for incoming messages to pursue the smallest time disparity



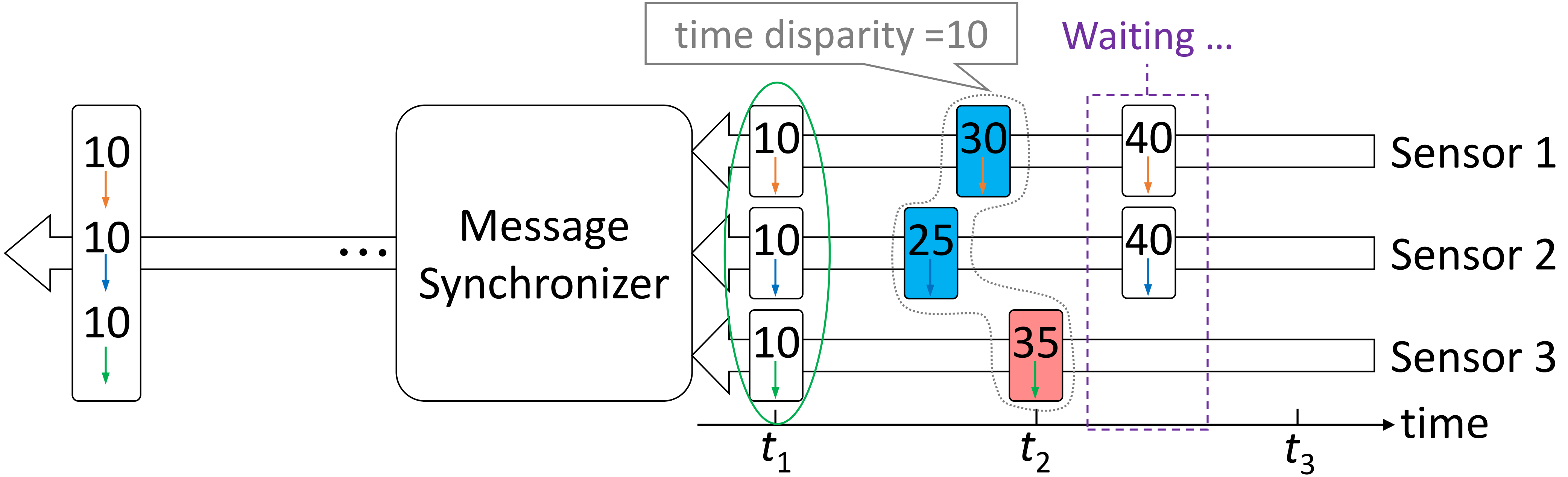
What happens when Pursuing the Smallest Disparity?

- Traditional methods in ROS
  - Called the ApproximateTime Algorithm
  - A sophisticated message synchronization policy
    - Predict the arrival time of messages
    - Wait for incoming messages to pursue the smallest time disparity



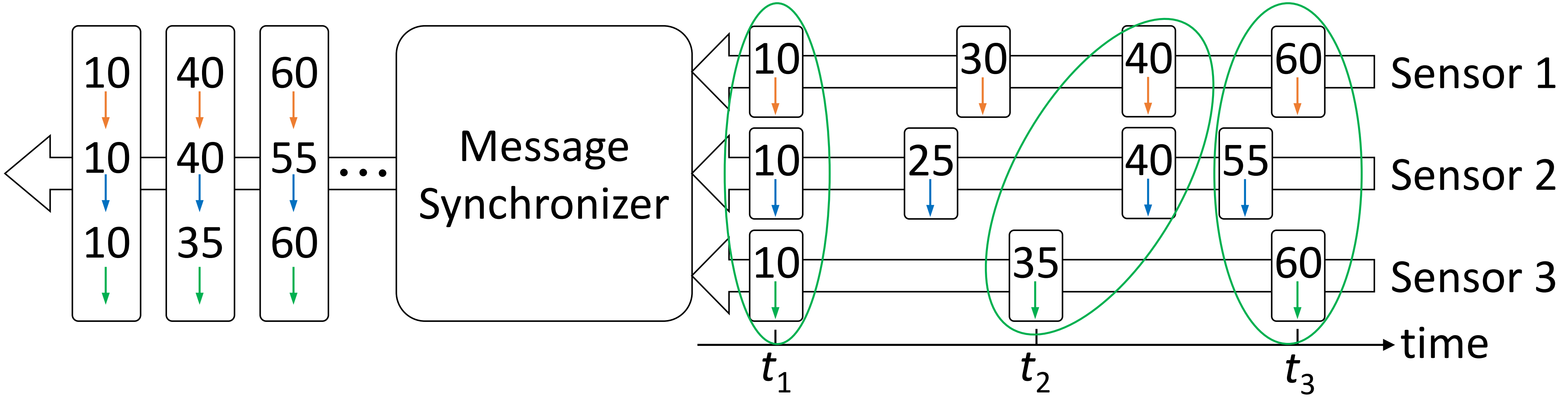
What happens when Pursuing the Smallest Disparity?

- Traditional methods in ROS
  - Called the ApproximateTime Algorithm
  - A sophisticated message synchronization policy
    - Predict the arrival time of messages
    - Wait for incoming messages to pursue the smallest time disparity



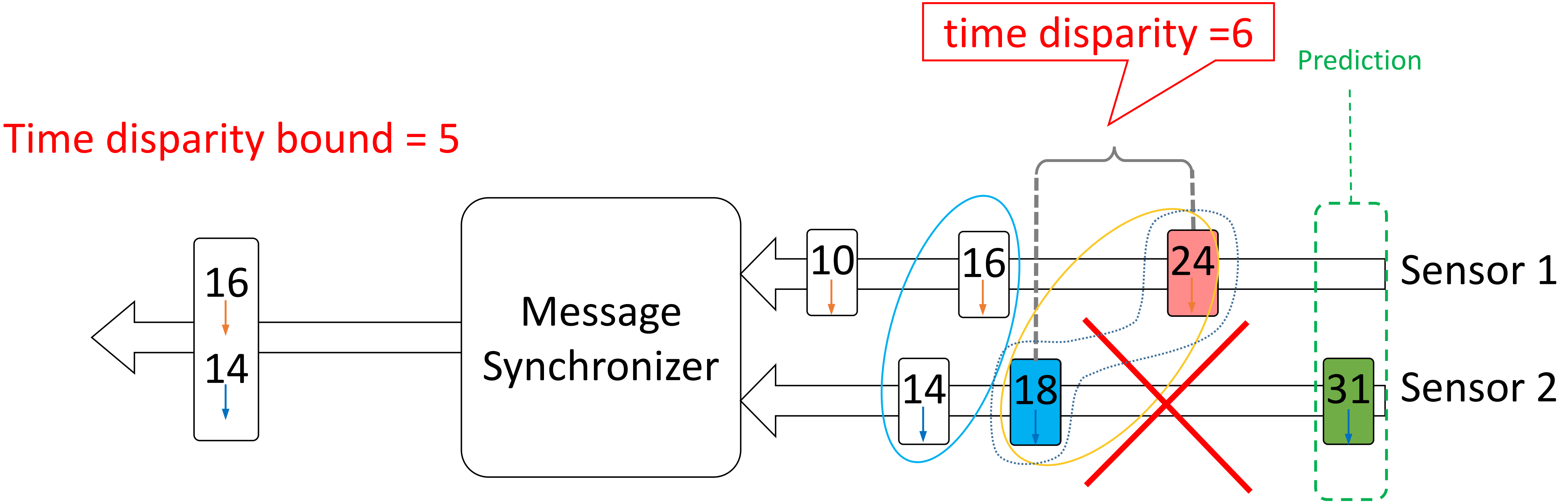
What happens when Pursuing the Smallest Disparity?

- Traditional methods in ROS
  - Called the ApproximateTime Algorithm
  - A sophisticated message synchronization policy
    - Predict the arrival time of messages
    - Wait for incoming messages to pursue the smallest time disparity



What happens when Pursuing the Smallest Disparity?

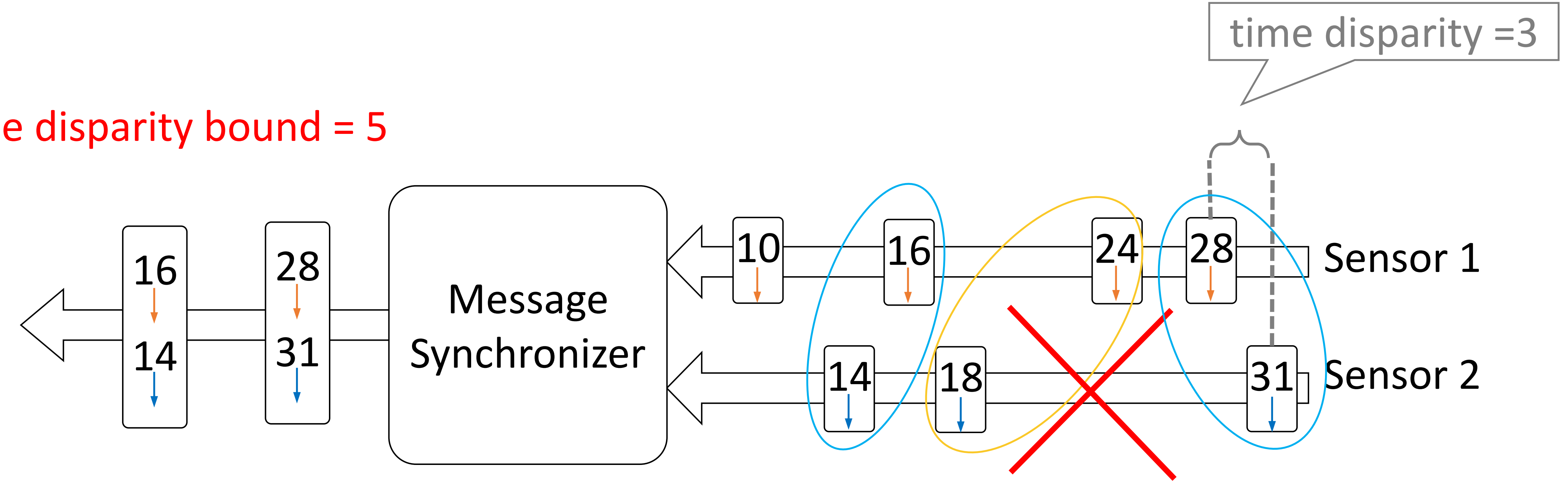
- ApproximateTime algorithm fails to meet timing disparity constraints
  - To pursue the smallest time disparity at an iteration
  - May worsen time disparity at future iterations



What happens when Pursuing the Smallest Disparity?

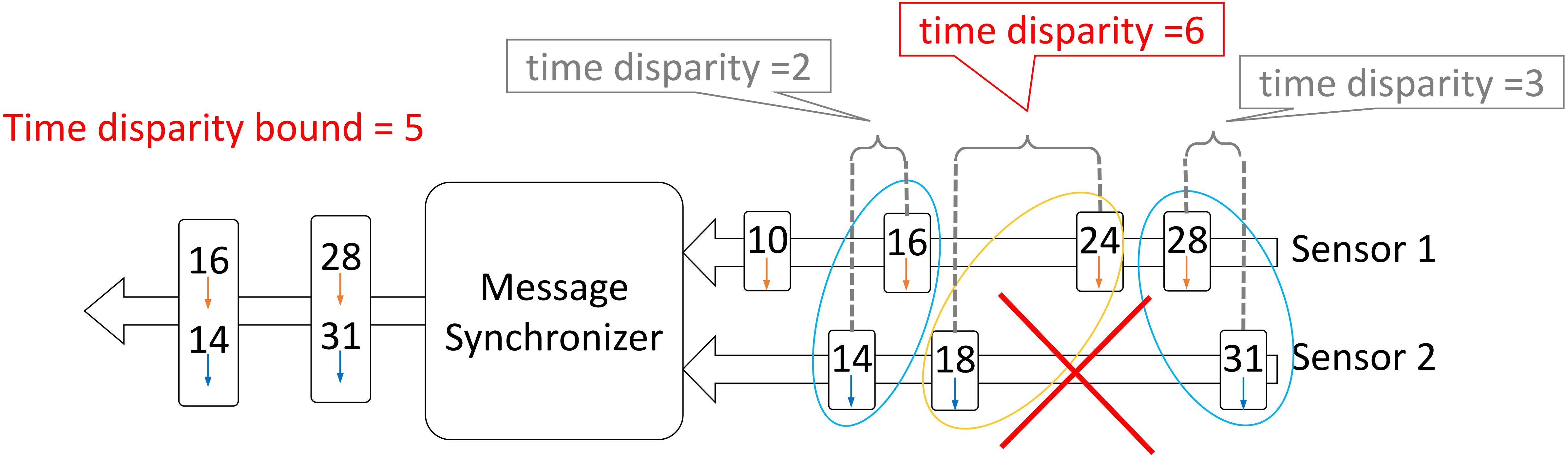
- ApproximateTime algorithm fails to meet timing disparity constraints
  - To pursue the smallest time disparity at an iteration
  - May worsen time disparity at future iterations

Time disparity bound = 5



What happens when Pursuing the Smallest Disparity?

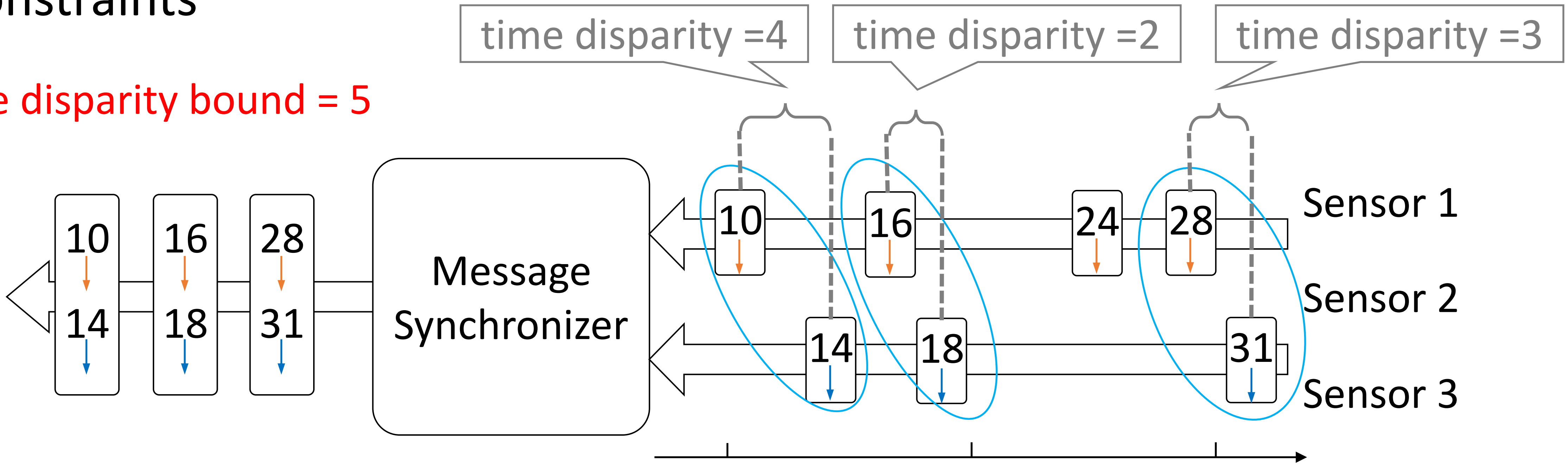
- ApproximateTime algorithm fails to meet timing disparity constraints
  - To pursue the smallest time disparity at an iteration
  - May worsen time disparity at future iterations



What happens when Pursuing the Smallest Disparity?

- ApproximateTime algorithm fails to meet timing disparity constraints
  - To pursue the smallest time disparity at an iteration
  - May worsen time disparity at future iterations
- There indeed exists fusion scheme that fulfills timing disparity constraints

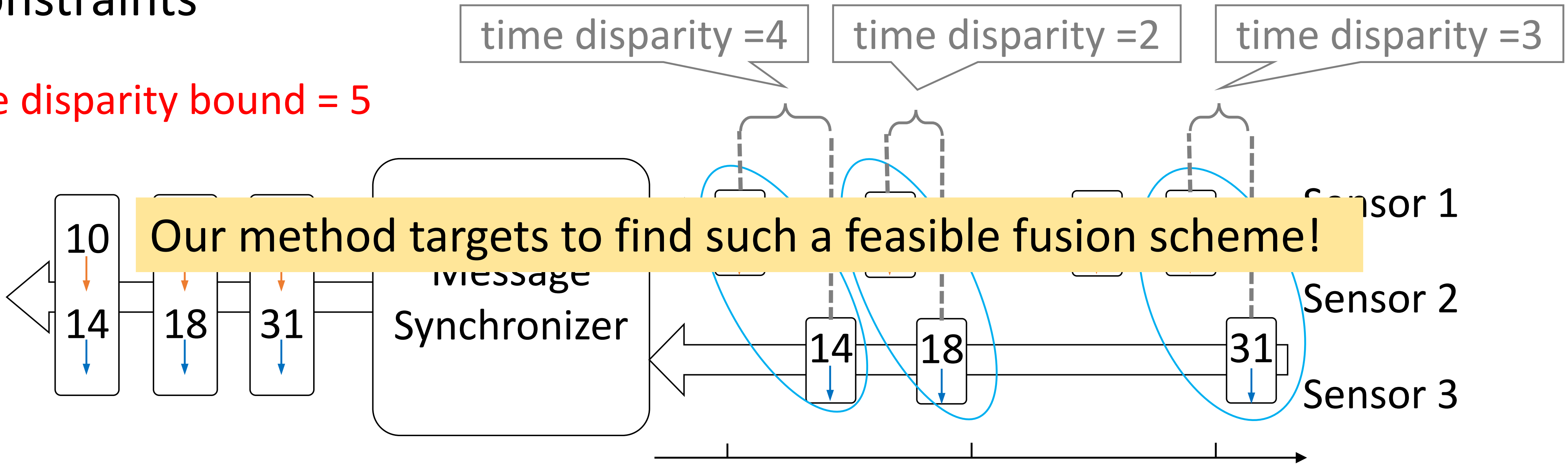
Time disparity bound = 5



What happens when Pursuing the Smallest Disparity?

- ApproximateTime algorithm fails to meet timing disparity constraints
  - To pursue the smallest time disparity at an iteration
  - May worsen time disparity at future iterations
- There indeed exists fusion scheme that fulfills timing disparity constraints

Time disparity bound = 5



Innovation of This Work

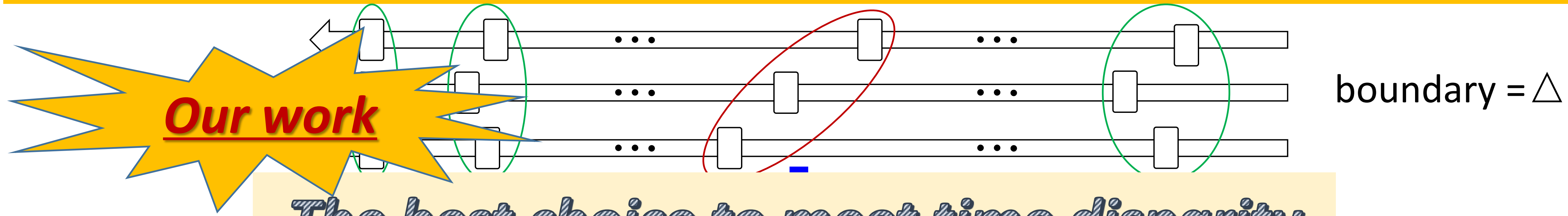
- More deeply rooted in practical applications

- Try to design more effective fusion mechanisms

Try to design more effective fusion mechanisms

Open Problem

If ApproximateTime Algorithm fails, is there better algorithms to meet time disparity constraints?



The best choice to meet time disparity constraints

success

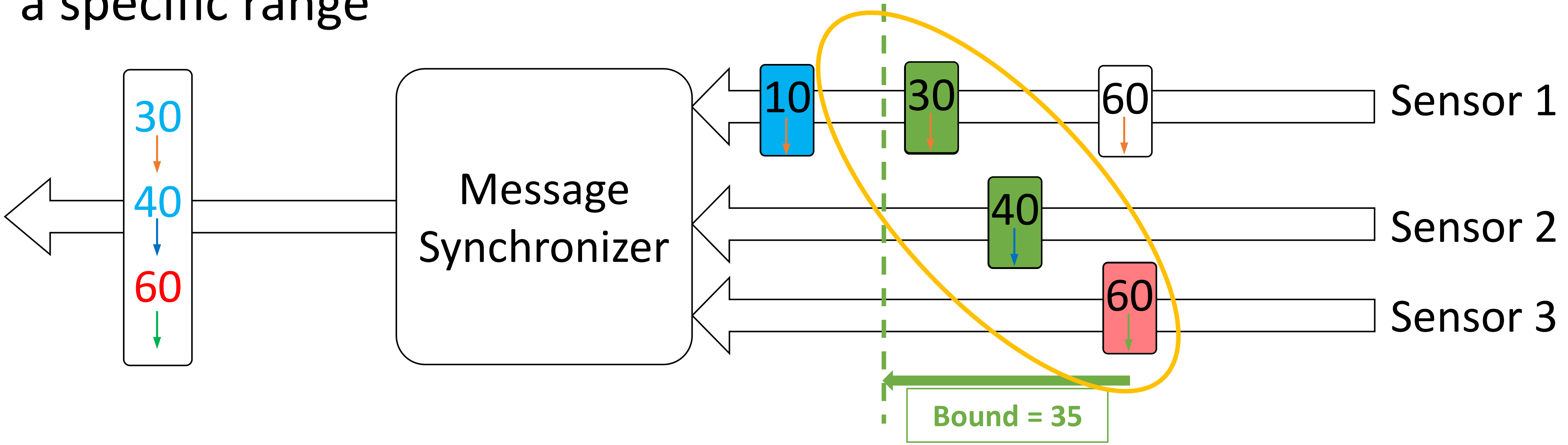
fail

Any better algorithm?

The SEAM Algorithm

Synchronize Earliest Arrival Messages (SEAM)

- Target: group the messages with reasonable time disparity
- Do this online:
  - Buffer arrived messages
  - Group the earliest arrival messages once their time disparity falls within a specific range



Evaluation Result

- Compare the Success Rate and Computation Time between SEAM and ApproximateTime
  - **Success Rate:** the percent of valid output message sets
  - **Computation Time**: from message selection to output message set generation
- SEAM outperforms ApproximatTime algorithm
  - Greater stability in success rate
  - Better real-time capbility:

    the computation time of SEAM is only 2/5 of that required by ApproximateTime on average

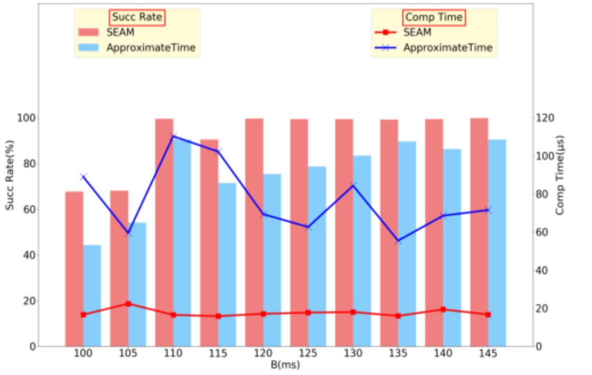


Fig. 10. Evaluation result for different thresholds B with configuration: $C = 100\text{ms}$, $N = 6$, $P = [30 - 100]\text{ms}$, $\alpha = 1.2$, $T = 50\text{s}$.

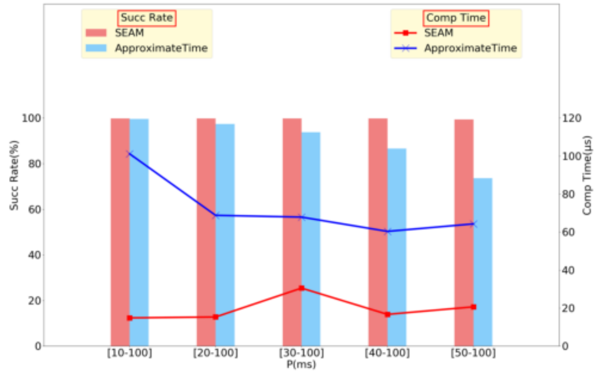


Fig. 12. Evaluation result for different time separations P with configuration: $C = 100\text{ms}$, $B = 125\text{ms}$, $N = 6$, $\alpha = 1.2$, $T = 50\text{s}$.

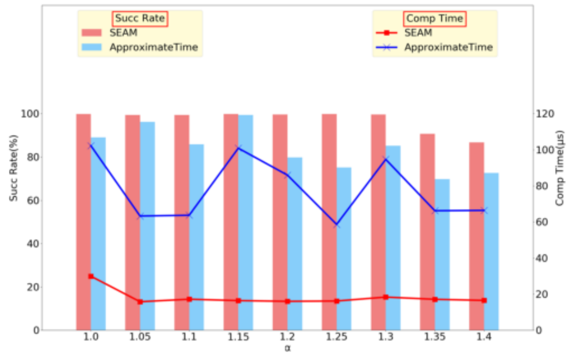


Fig. 13. Evaluation result for different jitter ratio α with configuration: $C = 100\text{ms}$, $B = 125\text{ms}$, $N = 6$, $P = [30 - 100]\text{ms}$, $T = 50\text{s}$.

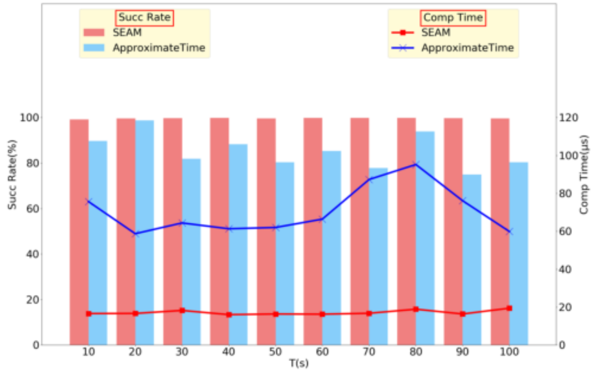


Fig. 14. Evaluation result for different interval lengths T with configuration: $C = 100\text{ms}$, $B = 125\text{ms}$, $N = 6$, $P = [30 - 100]\text{ms}$, $\alpha = 1.2$.

Summary

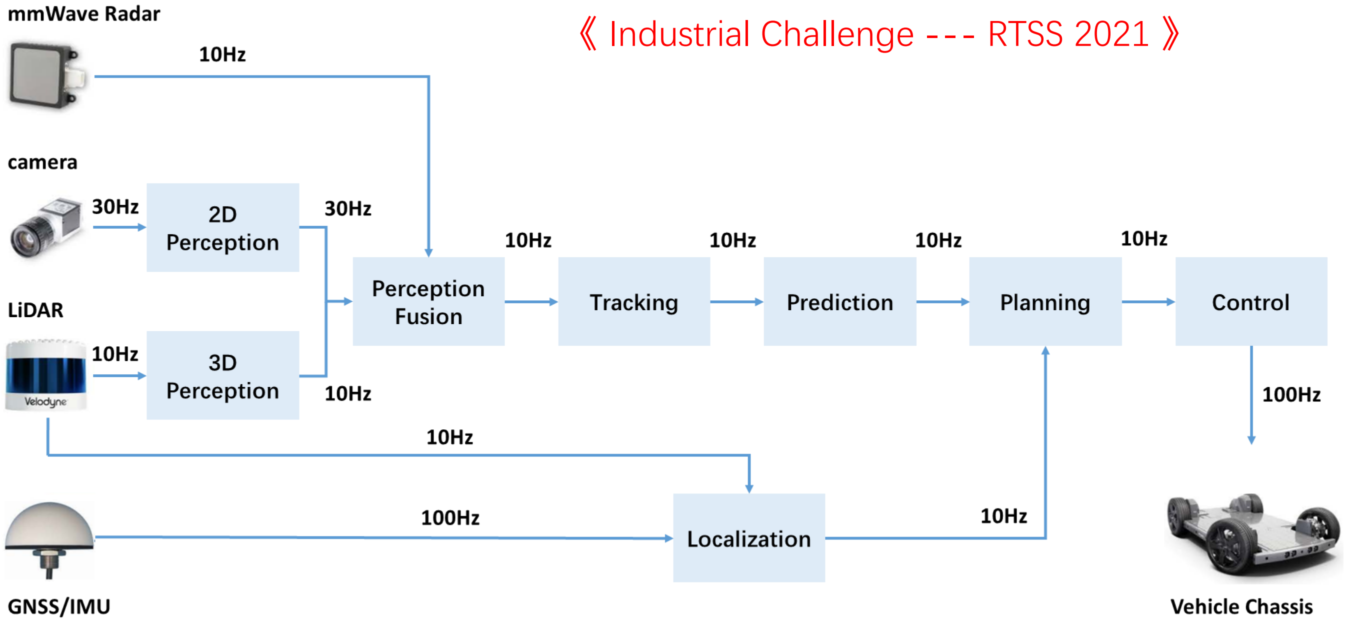
- Propose a novel message synchronization policy (SEAM)
- Prove the SEAM algorithm is optimal
  - Output maximum number of message sets satisfying time disparity constraints
- Implement SEAM into ROS 2

Jointly Ensuring Timing Disparity and End-to-End Latency Constraints in Hybrid DAGs

Jinghao Sun[1], Xisheng Li[1], Mingyang Gong[1], Nan Guan[2],
Zhishan Guo[3], Mingsong Chen[4], Jun Zhao[1] and Qingxu Deng[5]

Common Features of Autonomous Machines

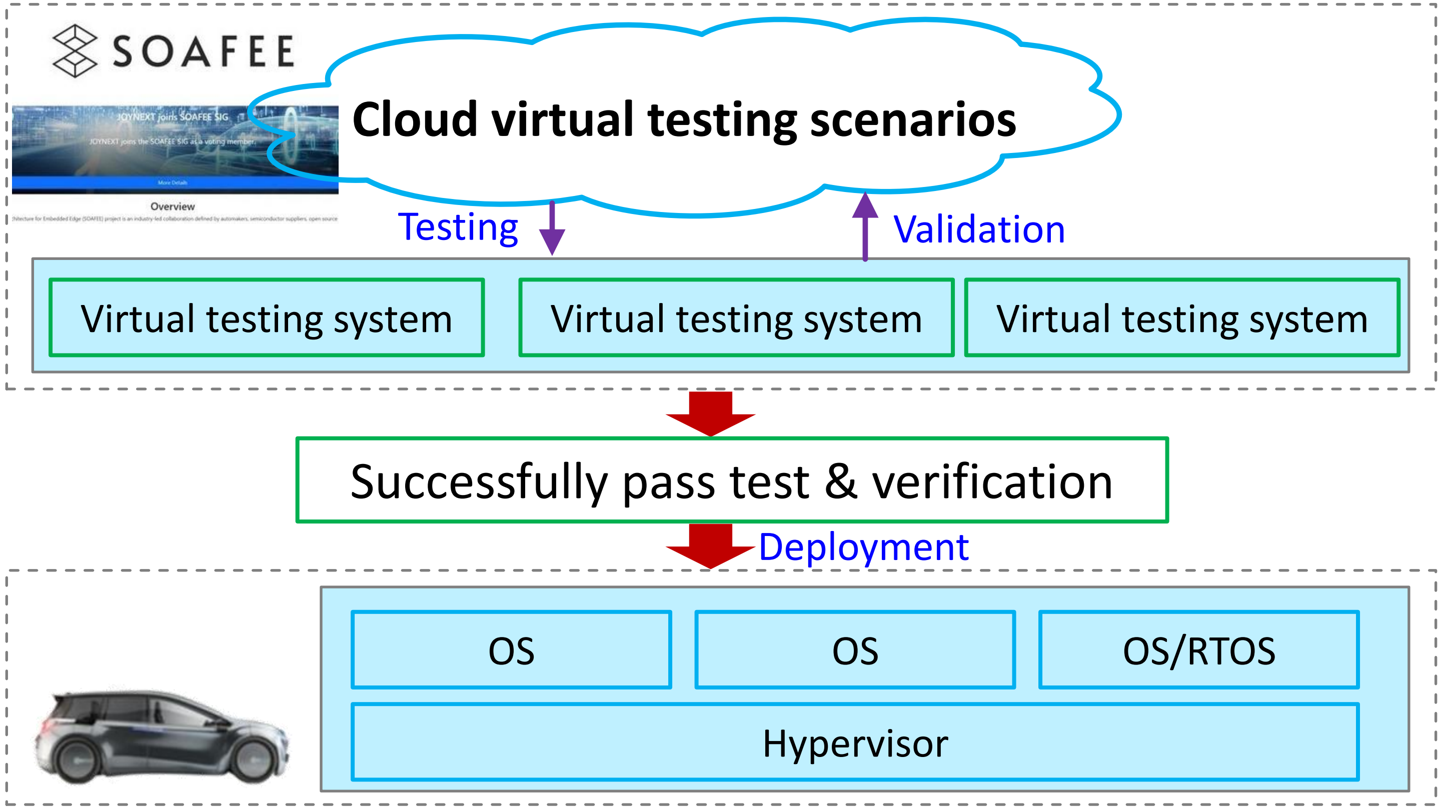
- React to open environment adaptively
  - Use multiple sensors to capture events, and then
  - Controllers react to these events in real time



Processing graph of an autonomous driving system

Common Features of Autonomous Machines

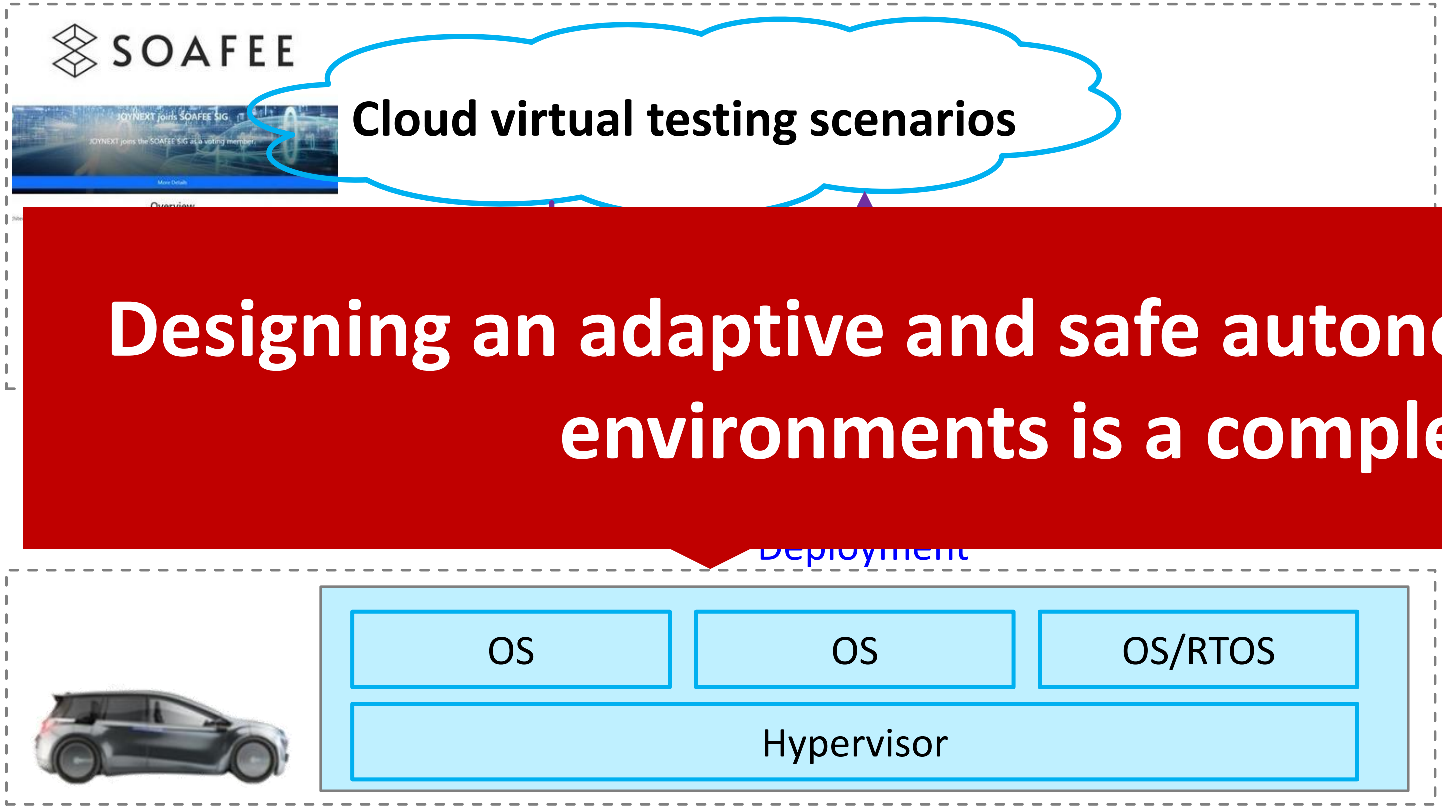
- Safety: Major Concern of Autonomous Machines
  - New digraphms, e.g., SOAFEE, is proposed to develop safe-critical autonomous machines



- Before deployment at the edge, AM software is virtualized and fully tested in the cloud
- Ensure it responds correctly and safely to real-world environmental conditions during runtime

Common Features of Autonomous Machines

- Safety: Major Concern of Autonomous Machines
  - New digraphms, e.g., SOAFEE, is proposed to develop safe-critical autonomous machines



Designing an adaptive and safe autonomous machine for open environments is a complex challenge

- Before deployment at the edge,

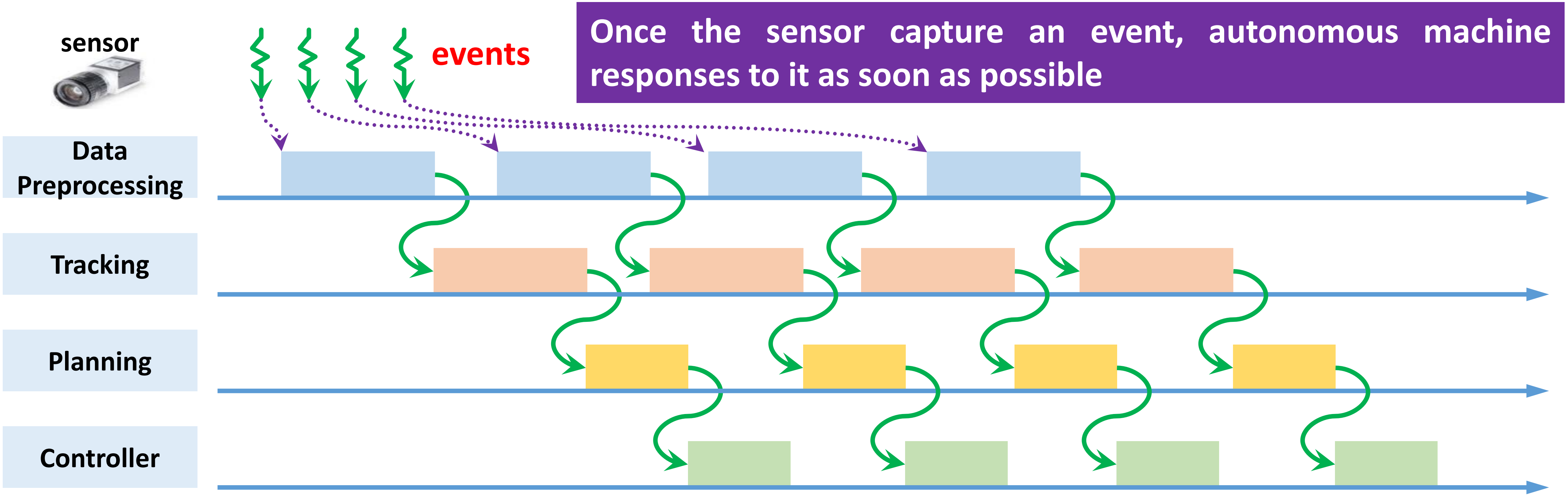
d

d

safely to real-world environmental conditions during runtime

Building Adaptive and Safe Autonomous Machines

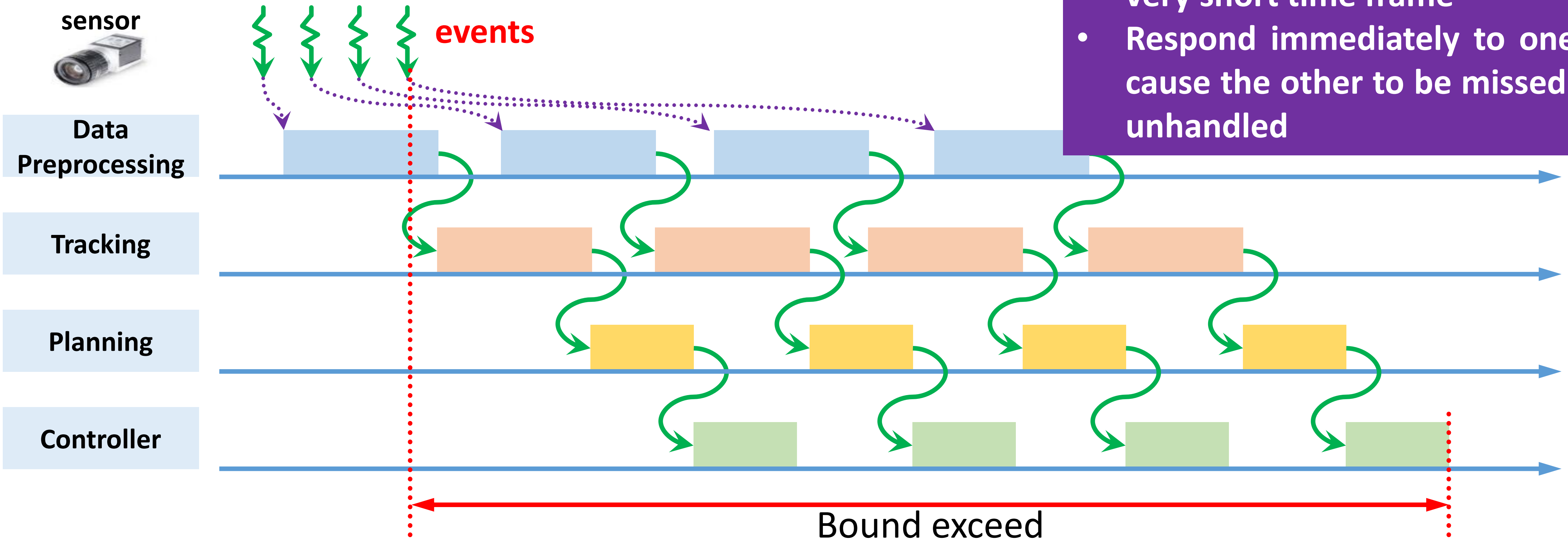
- Online methods seems an effcient way to implement adaptive autonomous machines



Building Adaptive and Safe Autonomous Machines

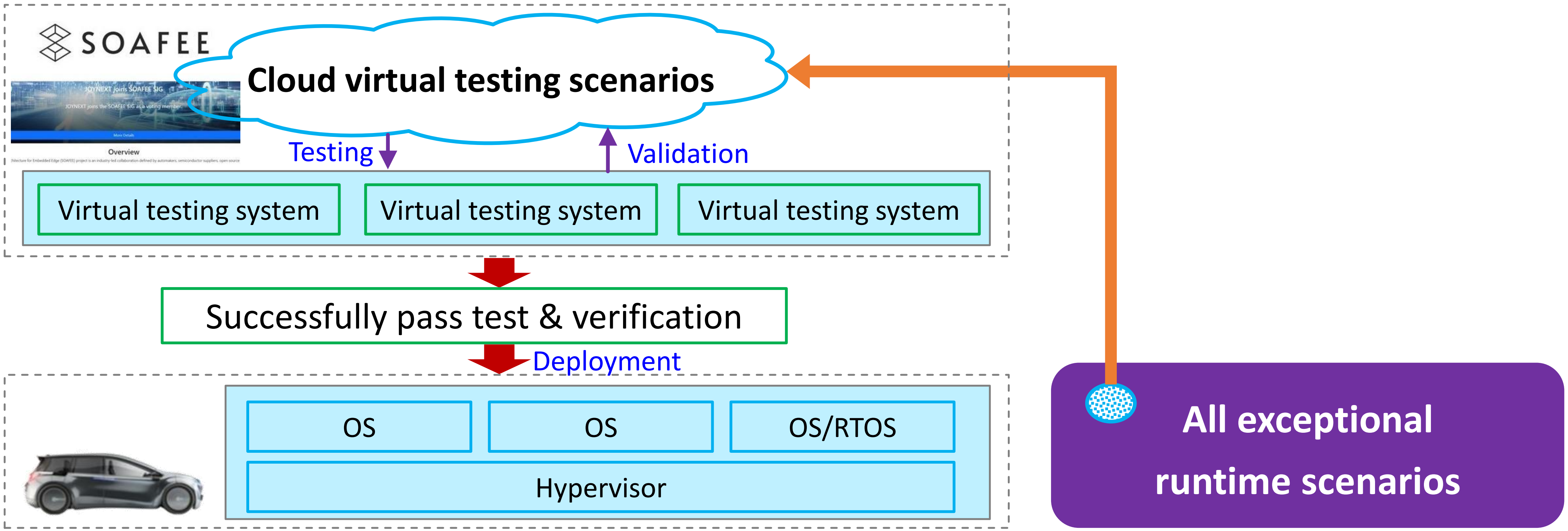
- Online methods introduce non-determinism
- Pose challenges for autonomous machine development and testing during the design phase

- Suppose two events occur within a very short time frame
- Respond immediately to one may cause the other to be missed or go unhandled



Building Adaptive and Safe Autonomous Machines

- In SOAFEE, even with extensive testing in the virtual cloud environment, it is impossible to cover all runtime scenarios

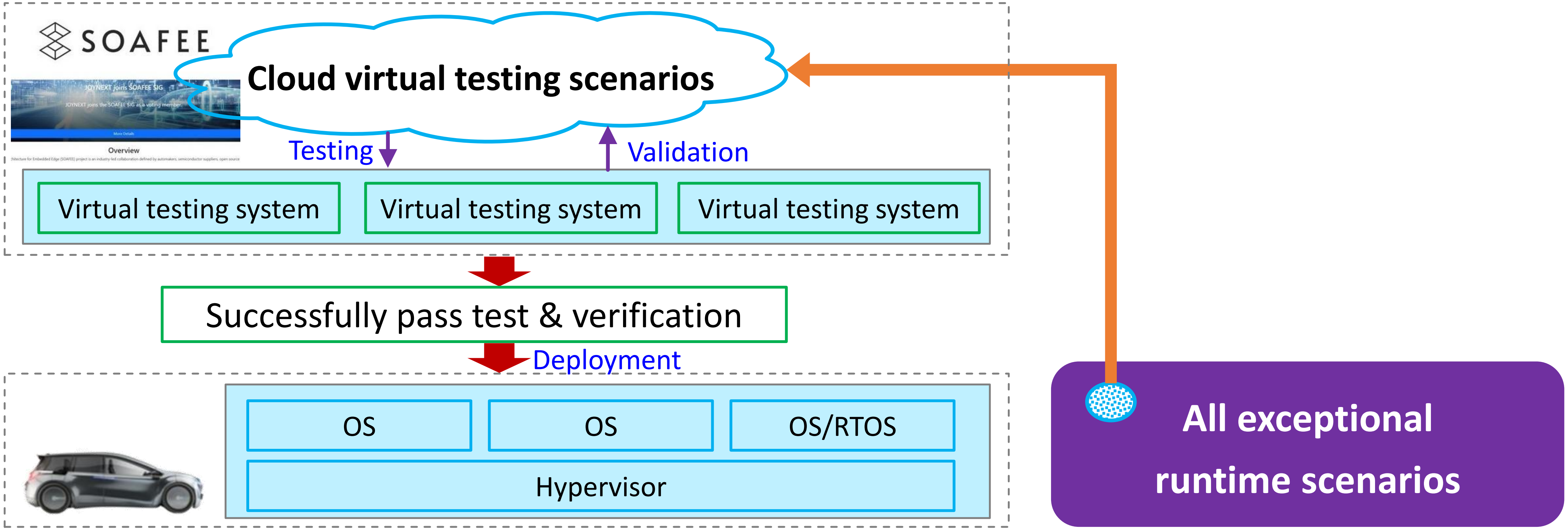


Building Adaptive and Safe Autonomous Machines

- A worthwhile direction
  - Simulate more realistic runtime scenarios
  - Enumerate additional corner cases

Better reflect real-world operating conditions

Enable more comprehensive and robust testing coverage

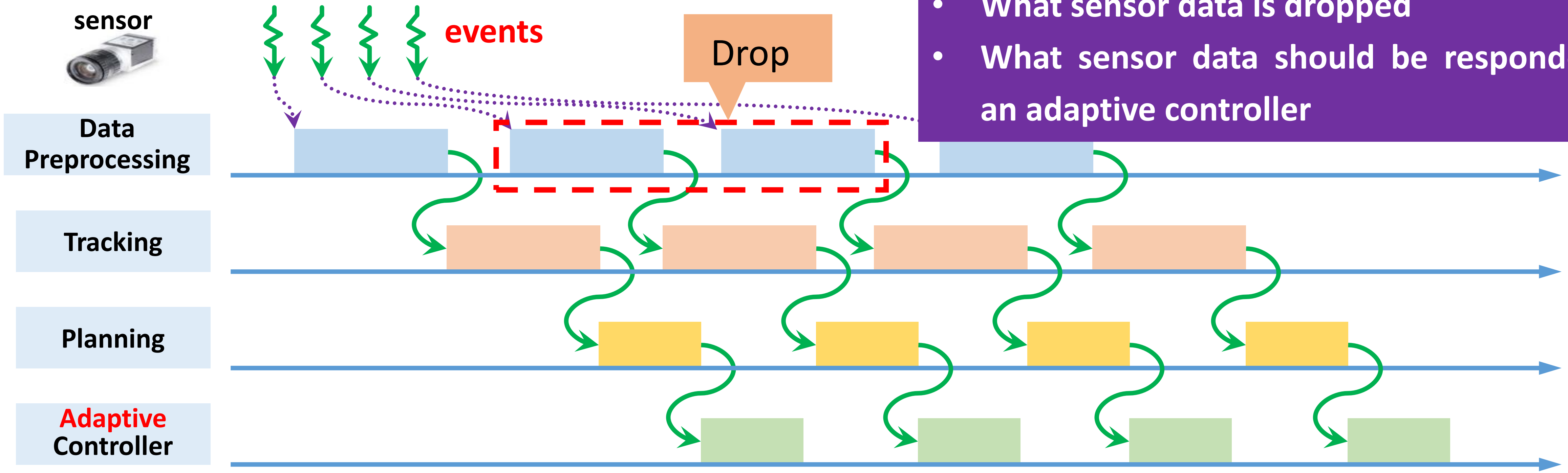


Building Adaptive and Safe Autonomous Machines

- An promising direction:
  - Offline + Adaptive Control (both deterministic and flexibility)

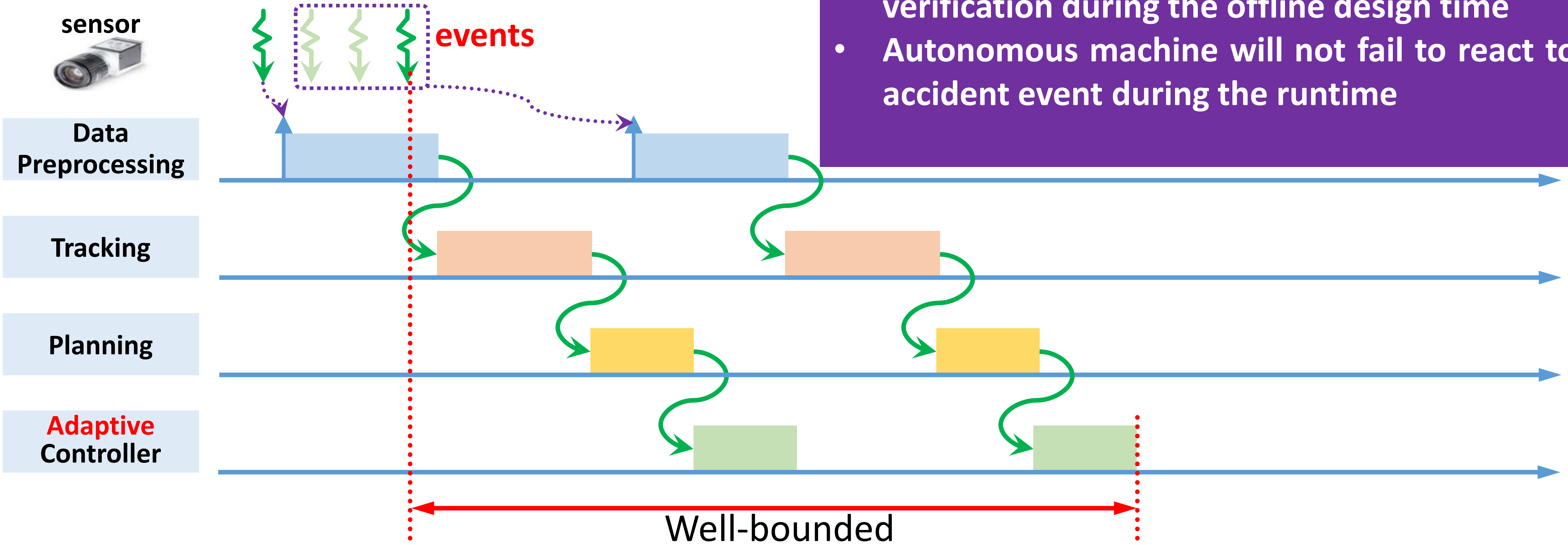
At the design phase, we determine

- What sensor data is dropped
- What sensor data should be responded by an adaptive controller



Building Adaptive and Safe Autonomous Machines

- An promising direction:
  - Offline + Adaptive Control (both deterministic and flexibility)



- Once the sensor data selection scheme is pass the verification during the offline design time
- Autonomous machine will not fail to react to any accident event during the runtime

Content

- **System Model**
- Offline Mapping
- Online Transmission
- Evaluation Work
- Conclusion

Our contribution

- To support cloud virtual testing & verification in SOAFEE, we need more deterministic timing behaviors in adaptive systems
- This paper propose an offline data mapping method (combined with online transmission) to guarantee system’s predictable time performance
- Experimental work shows that our method achieves a higher success rate than the built-in ROS 2 algorithm

最新录用的相关研究成果

[RTSS 2025] Submission #161 "WCDFP Analysis for Real-Time Tasks with..."

发件人： "RTSS 2025 HotCRP" <noreply-rtss25@hotcrp.com>

收件人： "Shining Sun" <20194745@stu.neu.edu.cn> "Deng qingxu" <dengqx@mail.neu.edu.cn> "Xu Jiang" <jian "Nan Guan" <nanguan@cityu.edu.hk>

抄　送： rtss25-tpc-chair@mpi-sws.org

Dear authors,

The 46th IEEE Real-Time Systems Symposium (RTSS 2025) program committee is delighted to inform you that your submission #161 has been accepted to appear in the conference.

* Title: WCDFP Analysis for Real-Time Tasks with Stochastic Release Patterns using Chernoff Bound
* Authors: Shining Sun (Northeastern University);

[RTSS 2025] Submission #64 "Investigating Jitter Propagation in Task..."

发件人： "RTSS 2025 HotCRP" <noreply-rtss25@hotcrp.com>

收件人： "Deng qingxu" <dengqx@mail.neu.edu.cn> "enrico bini" <enrico.bini@unito.it> "Martina Maggio" <n "Shumo Wang" <wangshumo0102@gmail.com>

抄　送： rtss25-tpc-chair@mpi-sws.org

Dear authors,

The 46th IEEE Real-Time Systems Symposium (RTSS 2025) program committee is delighted to inform you that your submission #64 has been conditionally accepted to appear in the conference, subject to shepherd approval.

* Title: Investigating Jitter Propagation in Task Chains
* Authors: Shumo Wang (Northeastern University, Shenyang, China);
  Enrico Bini (University of Turin, Italy);
  Qingxu Deng (Northeastern University, Shenyang, China);
  Martina Maggio (Saarland University, Germany)
* Site: https://rtss25.hotcrp.com/paper/64?

[CODES+ISSS 2025] Congratulations for Best Paper Nomination

发件人： "CODES+ISSS 2025 HotCRP" <noreply-codesisss25@hotcrp.com>

收件人： "Deng qingxu" <dengqx@mail.neu.edu.cn>

抄　送： pbogdan@usc.edu

Dear Qingxu,

On behalf of the CODES+ISSS Best Paper Award Committee (BPA), we are pleased to inform you that your paper has been nominated as a best paper candidate.

Paper ID: 11
Title: Re-thinking Memory-Bound Limitations in CGRAs

The BPA committee has selected three best paper candidates. The committee has read your paper and also aware of the reviews, responses, TPC discussion, and scores. The committee will attend your presentation as well as the poster session to clearly understand the contribution. The committee will make the final ranking considering everything and the winner will be announced during the award ceremony on Wednesday (Oct 1) at 8:30 am. Please plan to attend the award ceremony.

Your can maximize your chances by making a perfect 16-minute presentation and 4-minute Q/A followed by a great poster presentation. Please note that the committee information will remain confidential until we announce the winner during the award ceremony.

Thank you!

感谢聆听，欢迎交流指导！

更多了解可以查阅我们的网站

www.neu-rtes.org

dengqx@mail.neu.edu.cn