

2025-2026 CSEN 493 Directed Research Report

- Yinmingren Fu, 2025.12.26

Introduction

- The report is divided into two phases: completed work conducted in Fall 2025, and planned work to be carried out in Spring 2026.

Problem Statement

In ROS2, each task is assumed to include parameters such as computation time and communication delay, and is subject to a fixed deadline (e.g., 100 milliseconds). Therefore, maximizing the quality of execution for each task during continuous system operation is crucial. The variability of network latency is a key challenge. When network latency increases, the remaining execution time budget for computation decreases, making it difficult to guarantee task completion before the deadline. Furthermore, tasks have varying levels of importance and different time constraints, and may form execution chains, where the overall Quality of Service (QoS) depends on the performance of individual tasks and system-level scheduling behavior.

Therefore, a scheduling framework is needed that can dynamically determine task priorities to ensure timely completion within a fixed time budget, while providing a unified evaluation framework across system, task chain, and task levels.

Completed Work (Fall 2025)

System Design

During this semester, I designed a priority-aware task scheduling framework built on ROS2, based on several papers and documents, aiming to operate under a client-server execution model within a strict time budget. Each task is described by parameters such as its arrival time, deadline, expected execution time, and network latency, and may belong to a larger execution chain.

The core design principle is to dynamically calculate task priorities based on the remaining execution time of the tasks. Given a fixed time budget, the scheduler adjusts the actual execution time based on observed or estimated network latency to ensure that tasks can still be scheduled before their deadlines. Tasks with tighter deadlines or less remaining time budget will be given higher priority.

The system should support:

- Dynamic priority calculation upon many parameters
- Priority-based preemptive task scheduling

QoS is evaluated at three levels: single-task QoS (deadline and timing accuracy), chain-level QoS (accuracy and data freshness), and system-level QoS (throughput and data freshness within a sliding time window).

Implementation Details

This scheduler is based on the open-source PICAS priority scheduling framework, implemented in C++ on an Ubuntu system, and integrated into the ROS2 Galactic environment. This implementation extends the rclcpp

execution layer, introducing a custom scheduler to intercept task execution requests and enforce a priority-based scheduling policy.

Each incoming task (e.g., timer callbacks or other ROS2 executables) is assigned a priority, calculated based on its network latency, estimated execution time, arrival time, and deadline. Tasks are inserted into a priority queue and scheduled non-preemptively according to their calculated priority.

During this semester, I have implemented and functionally verified the custom scheduler, demonstrating its correct priority assignment logic and priority scheduling. Concurrently, I designed a comprehensive QoS evaluation framework to measure system-level, chain-level, and task-level performance metrics.

Planned Work

Expected Outcomes

During the Spring phase, I still have some tasks to complete:

1. I will build a more complete and extensible priority-aware scheduling framework based on ROS2. First, I will use the designed QoS evaluation framework to further refine and verify the scheduler implementation. I will try to write some test cases to systematically evaluate system-level, chain-level, and task-level QoS, ensuring that the scheduler can schedule tasks to maximize task QoS within a fixed time budget.
2. For heterogeneous computing resources (e.g., CPU/GPU) and preemptive execution mechanisms, I will attempt to implement them based on the custom scheduler.

Related Research

1. Choi H, Xiang Y, Kim H. PiCAS: New design of priority-driven chain-aware scheduling for ROS2[C]//2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE, 2021: 251-263.
2. Liu S, Wagle R, Ahmed S, et al. ROSRT: Enabling Flexible Scheduling in ROS 2[C]//Proceedings of the 46th IEEE Real-Time Systems Symposium. to appear. 2025.
3. Sobhani H, Choi H, Kim H. Timing analysis and priority-driven enhancements of ROS 2 multi-threaded executors[C]//2023 IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE, 2023: 106-118.
4. Staschulat J, Lange R, Dasari D N. Budget-based real-time executor for micro-ros[J]. arXiv preprint arXiv:2105.05590, 2021.
5. Jiang X, Ji D, Guan N, et al. Real-time scheduling and analysis of processing chains on multi-threaded executor in ros 2[C]//2022 IEEE Real-Time Systems Symposium (RTSS). IEEE, 2022: 27-39.