Performance Improvement of Multi-Robot Data Transmission in Aggregated Robot Processing Architecture With Caches and QoS Balancing Optimization

Abdul Jalil *, Jun Kobayashi , Takeshi Saitoh

Posted Date: 8 May 2023

doi: 10.20944/preprints202305.0431.v1

Keywords: Multi-Robot; Aggregated Robot Processing; Caches; QoS; Optimization; ROS 2

Article

Performance Improvement of Multi-Robot Data Transmission in Aggregated Robot Processing Architecture with Caches and QoS Balancing Optimization

Abdul Jalil [1,*,†], Jun Kobayashi [2,†] and Takeshi Saitoh [3,†]

[1] Kyushu Institute of Technology; malla.abdul-jalil545@mail.kyutech.jp
[2] Kyushu Institute of Technology; jkoba@ics.kyutech.ac.jp
[3] Kyushu Institute of Technology; saitoh@ai.kyutech.ac.jp
* Correspondence: malla.abdul-jalil545@mail.kyutech.jp
† 680-4 Kawazu, Iizuka-shi, Fukuoka, 820-8502, Japan: Kyushu Institute of Technology.

Abstract: Robot Operating System 2 (ROS 2) is a robotic software that uses a set of Quality of Service (QoS) policies to manage the quality of robot data transmissions in a network, such as the RELIABLE and KEEP_LAST options. In ROS 2 node communication, the RELIABLE connection guarantees that all message data can be properly sent from the publisher to the subscriber. However, strict reliability is not guaranteed if the RELIABLE connection uses the KEEP_LAST option to transmit the robot data in the publish-subscribe communication. This study aims to analyze the efficiency of local cache, cache control, and QoS balancing optimization to improve ROS 2 node communication when using the RELIABLE and KEEP_LAST options to transmit multi-robot data in Aggregated Robot Processing (ARP) architecture. Our idea in local cache and cache control is to streamline the sensor data output before processing it when the sensor device produces the data with the same value in a row. Furthermore, QoS balancing optimization aims to balance the DEPTH and DEADLINE QoS configuration to determine the rates and buffer size in ROS 2 node communication. This study shows that combining local cache and QoS balancing optimization improves multi-robot data transmission and cooperation in ARP architecture.

Keywords: multi-Robot; Aggregated Robot Processing; caches; QoS; optimization; ROS 2

1. Introduction

Multi-Robot Systems (MRS) consist of several robots cooperating to handle complex tasks together [1]. Each robot can exchange information data based on the node communication system in the network using Robot Operating System 2 (ROS 2) and uses a set of Quality of Service (QoS) policies to manage the quality of robot data transmission, such as RELIABILITY, HISTORY, DEPTH, DEADLINE, DURABILITY, LIVELINESS, and LEASE_DURATION [2,3]. RELIABILITY has two options to manage the mechanism of robot data transmission: RELIABLE and BEST_EFFORT. RELIABLE connection guarantees that all message data can be properly sent from the publisher and subscriber, whereas the BEST_EFFORT connection focuses only on sending message data without concern for packet loss. In the HISTORY policy, this QoS has two options to store the data sample: KEEP_ALL and KEEP_LAST. When using the KEEP_ALL option, all data transmitted from the publisher and subscriber will be stored in the buffer specified by the underlying Data Distribution Service (DDS) middleware. Furthermore, in the KEEP_LAST option, the buffer size to store the data sample is configured in DEPTH.

In ROS 2 node communication, strict reliability is not guaranteed if the RELIABLE connection uses the KEEP_LAST options to store the data sample in the publish-subscribe communication. It happens if the DEADLINE rates for transmitting the message data are not balanced with the buffer size configured in the DEPTH. If DEADLINE configures the rates with high frequency and DEPTH configures the buffer with a small size, some packets will be lost in ROS 2 node communication[4–9]. Otherwise, if

 doi:10.20944/preprints202305.0431.v1

the DEADLINE policies configure the rates with low frequency, the rates for data transmission from the publisher and subscriber will be low, affecting the real-time message data transmission between the publisher and subscriber.

Studies on improving robot data transmission using ROS 2 are interesting topics that some researchers have developed. Fernandez carried out the study on improving ROS 2 performance with different QoS and cyber security settings [4]. That study showed that a difference in QoS profiles and security settings could affect the latency and throughput of data transmission in robotic systems. Choi [10] designed the implementation of the priority-driven chain-aware scheduling (PiCAS) method for ROS 2 callbacks, nodes, and executors. The researchers used the PiCAS method to improve end-to-end message data transmission latency through a default ROS 2 scheduler. The improvement of robot data transmission in ROS 2 using a partial serialization algorithm has been developed by Wang [11]. In that study, the researchers used a partial serialization algorithm to analyze the efficiency of inter-process communication (IPC), called Toward Zero Copy (TZC), by generating and dividing the transmission of message data into two parts: socket and shared memory. The use of a priority synthesis algorithm to improve the predictability of event chains in ROS 2 has been analyzed by Randolph [12]. In that study, a priority synthesis algorithm was used to improve the predictability of ROS 2 applications in response time, jitter, and missed deadlines. Furthermore, Jiang et al. [13] have implemented the Adaptive Two-Layer Serialization Algorithm (ATSA) to optimize message passing in ROS 2. The researchers used the ATSA algorithm to minimize the total conversion and serialization costs of different message types in ROS 2.

The contribution of this study is to analyze the performance of local cache, cache control, and QoS balancing optimization to improve the quality of multi-robot communication in Aggregated Robot Processing (ARP) architecture when ROS 2 for robot data transmission uses the RELIABLE and KEEP_LAST options to transmit the robot data between publisher and subscriber. ARP is an architecture in robotic systems that centralizes robot data processes on a computer, called the Computer Environment Dedicated to Data Processing (CEDDP) [14]. In the ARP architecture, the robot computer and CEDDP can exchange message data through wireless networks. Our idea in the local cache is to streamline the sensor data output on the robot computer before sending it to the CEDDP. Then in the cache control, the streamlining of sensor data is executed in CEDDP before processing it for robot localization and path planning. Furthermore, QoS balancing optimization aims to balance the DEADLINE and DEPTH QoS configurations to determine the rates and buffer size in publish-subscribe communication.

This paper consists of five chapters. In the second chapter, we elaborate on the materials and methods: aggregated robot processing, local cache, cache control, and optimization. In the third and fourth chapters, we discuss the experimental results and the discussion. Furthermore, the fifth chapter shows the conclusion of this study.

2. Materials and Methods

2.1. Aggregated Robot Processing

The flow of robot data processes typically consists of three components: sensing, planning, and actuation [15]. These components can be connected as node communication systems in the network using ROS 2. ROS 2 is a robotic software built on top of the Data Distribution Service (DDS) [2]. Based on ROS 2 node communication, the function of a node in the sensing component is to manage the sensor hardware, read the sensor output as input information for the robot, and send the sensor data to a node in the planning component through a topic. After that, the node in the planning component will process the sensor data to determine the robot's action, localization, and path planning, then send the result to a node in the actuation component to control the robot actuators. The topic is a bus used in ROS to transmit message data between the publisher and the subscriber [16]. The ARP architecture separates the robot data processes between the sensing, planning, and actuation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

components. Specifically, the sensing and actuation components run on the robot computer, and the planning component runs on CEDDP. Figure 1 shows the MRS data transmission in the ARP architecture based on the ROS 2 node communication mechanism.

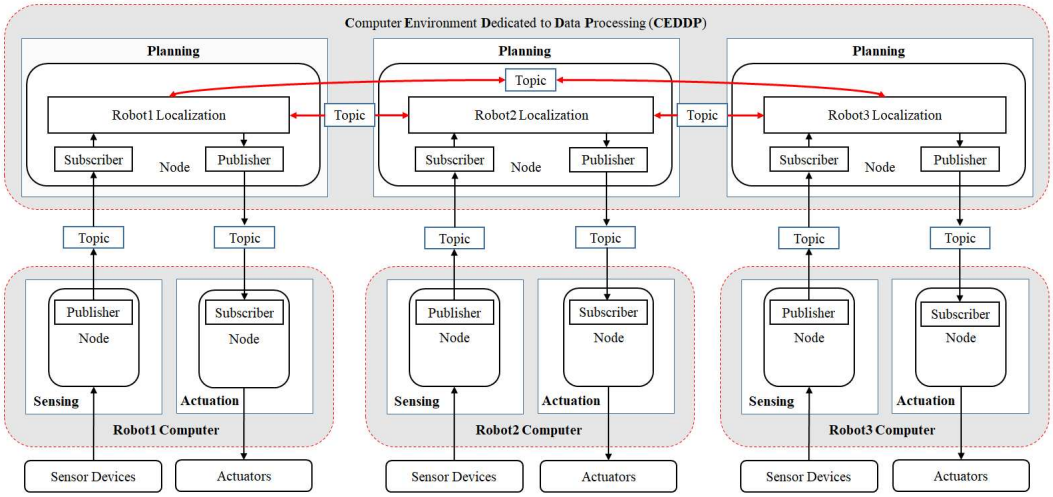

Figure 1. Aggregated robot processing architecture.

Based on the information in Figure 1, it can be seen that the data communication system between robots in MRS was exchanged in CEDDP, not in a wireless network. The function of the wireless network in ARP architecture is to transfer sensor data between the sensing and planning components and to transmit the localization result from the planning to the actuation components.

2.2. Local Cache

The local cache function in this study is to streamline the transmission of sensor data from the sensing components to the planning components when the sensor device produces the data output with the same value in a row. Figure 2 shows the publish-subscriber mechanism in ROS 2 and how the local cache works to streamline sensor data transmission between the sensing and planning components.

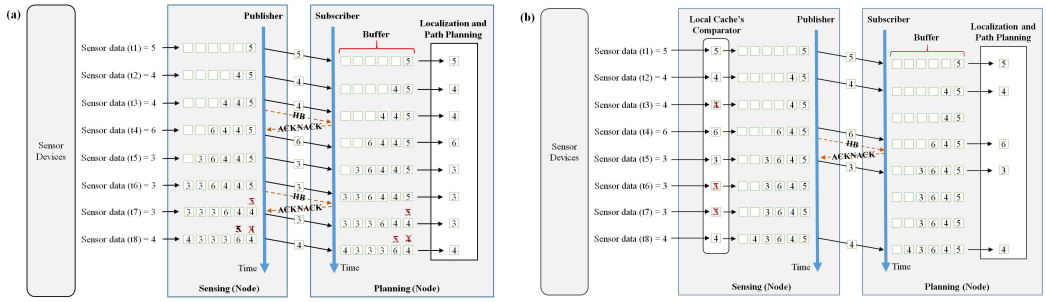

Figure 2. (a) Publish-subscribe mechanism in ROS 2; **(b)** Local cache works in a sensing component to streamline the sensor data before publishing it to the subscriber.

In ROS 2 node communication, the RELIABLE connection uses the Heartbeat (HB) and Acknowledgment (ACKNACK) mechanism to transmit the data sample between the publisher and the subscriber [4,5]. When the publisher sends a data sample to the subscriber, the publisher will transmit the data with a sub-message HB to announce that the subscriber should receive the data sent from the publisher. After that, the subscriber sends the ACKNACK to the publisher to confirm that the subscriber has already received all data samples. In that case, if the publisher fails to transmit a data sample, the subscriber will send an ACKNACK to the publisher to confirm the failure of the data

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

transfer. After that, the publisher sends the lost data sample to the subscriber in the following data transmission.

Figure 2(a) shows the publish-subscribe mechanism in ROS 2 and illustrates how the buffer stores and discards data samples in node communication. In the KEEP_LAST option, if the DEPTH policy configures the buffer with a small queue size, the buffer cannot store all data samples sent from the publisher and received by the subscriber, so the buffer will discard the oldest data sample in the queue to make space for the newer ones. In a RELIABLE connection, if the publisher fails to send a data sample to a subscriber, the publisher will retrieve the lost data sample from the buffer and re-transmit it. In that case, if the publisher's buffer cannot accommodate storing the lost data sample due to the buffer size, strict reliability is not guaranteed in a RELIBALE connection due to the lost data sample already discarded in the publisher's buffer.

Furthermore, Figure 2(b) shows the local cache mechanism to streamline the sensor data before the publisher in a sensing component sends it to the subscriber in a planning component. It can be seen that the local cache holds the sensor data transmission if the sensor device produces the data with the same value in a row. Based on a RELIABLE connection mechanism, we propose the local cache method to improve MRS data transmission by reducing the HB and ACKNACK rates and helping the buffer to reduce the data stored in the queue and not store the data with the same value in a row. Figure 3 shows the local cache flowchart to streamline sensor data transmission in a sensing component. The local cache works to streamline the sensor data in each iteration in a sensing component node until it is shut down.

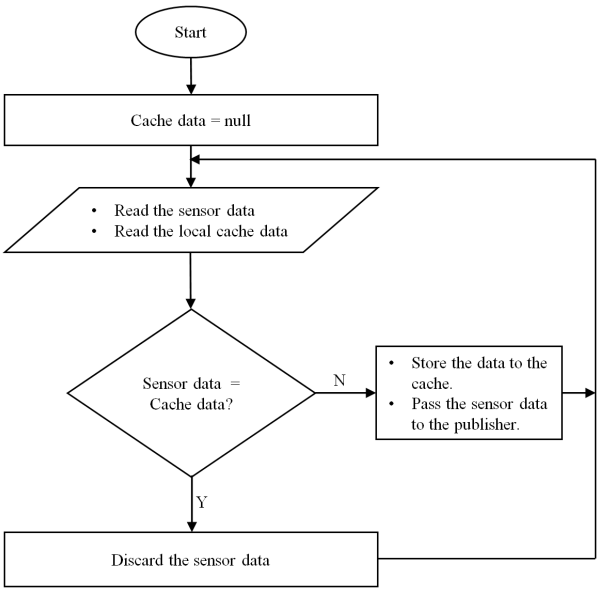

Figure 3. Local cache flowchart.

2.3. Cache Control

Figure 4 shows the cache control mechanism to streamline sensor data transmission in CEDDP before processing it in the planning component. In ARP architecture, each robot in MRS cooperates based on the communication of the planning component in CEDDP. Our idea in cache control is to improve the MRS communication performance in CEDDP by reducing the planning component work to receive the sensor data transmission sent from the sensing component.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

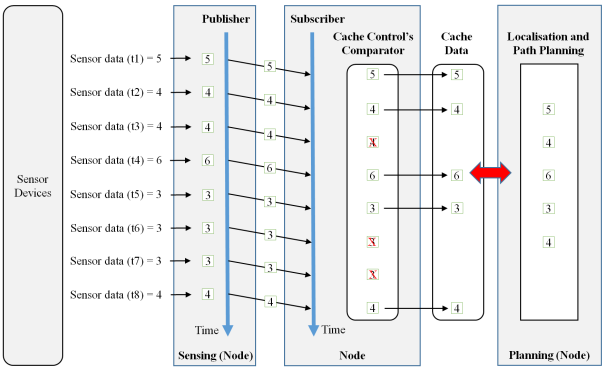

Figure 4. Cache control works in the CEDDP.

Furthermore, Figure 5 shows the cache control flow chart to streamline the sensor data sent from the sensing component. To run the cache control in CEDDP, we used a node in CEDDP to subscribe to the sensor data sent from the sensing component. After that, the cache control will compare the sensor and cache data values. If the sensor data are the same as the cache data value, then the cache control will discard the sensor data. Then if the sensor data value differs from the cache data, the cache control will store the sensor data in the cache and read the next sensor data transferred from the sensing component. This cache control mechanism is run repeatedly until the node is shut down.

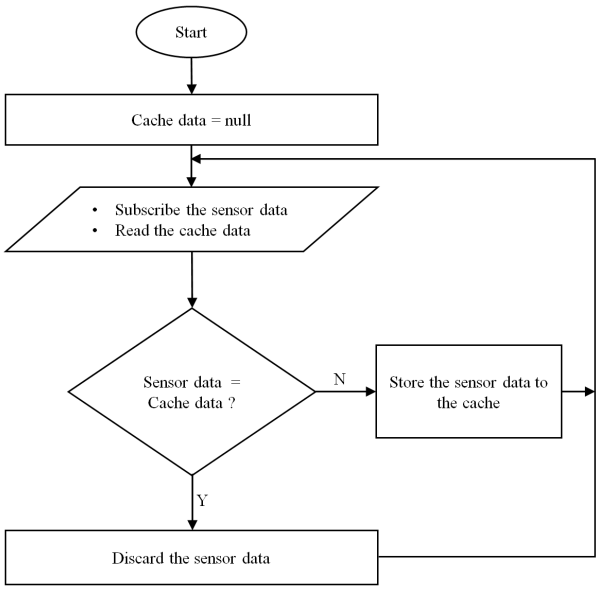

Figure 5. Cache control flowchart.

2.4. QoS Balancing Optimization

Figure 6 shows MRS communication in the ARP architecture, illustrating DEPTH and DEADLINE functions. DEPTH is a QoS policy in ROS 2 to configure the buffer size when the RELIABLE connection uses the KEEP_LAST option to transmit the data sample between the publisher and the subscriber. Furthermore, DEADLINE is a QoS policy for configuring data transmission rates between the publisher and the subscriber via topic. Figure 7 shows the DEPTH and DEADLINE illustration to configure the buffer size and rates in the publish-subscribe communication.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

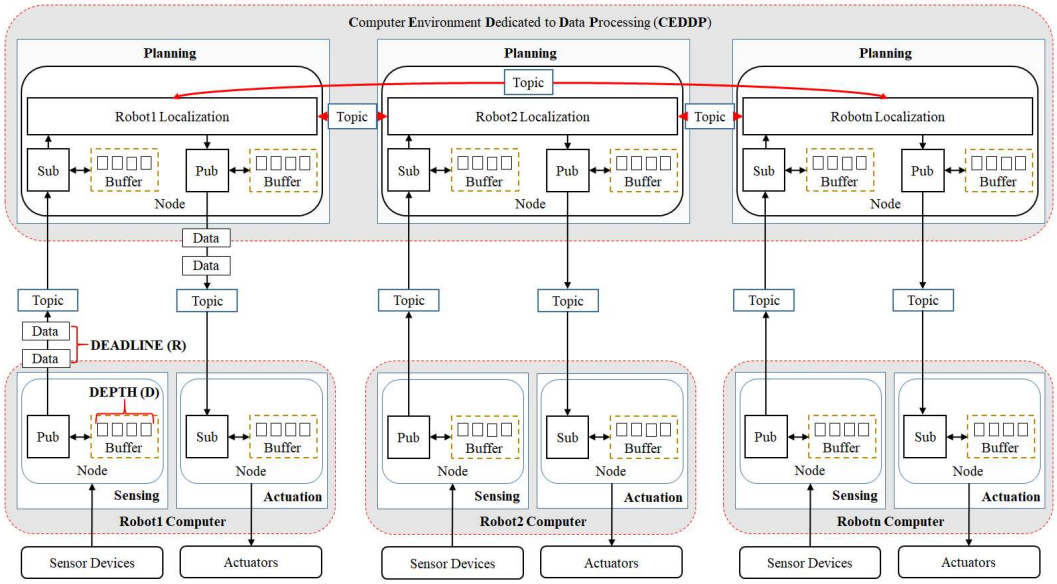

Figure 6. QoS balancing optimization for MRS communication in ARP architecture.

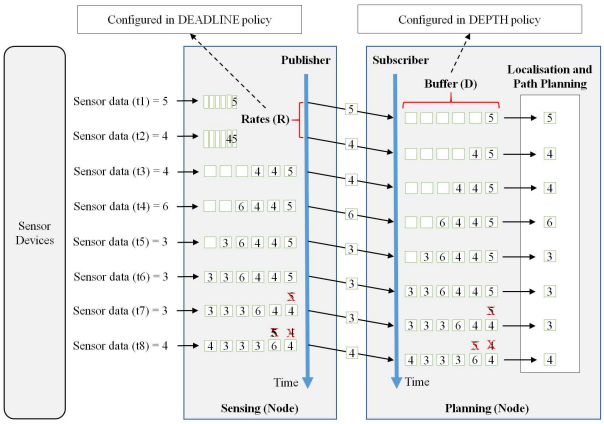

Figure 7. DEPTH and DEADLIENE QoS configuration illustration.

Based on the illustration shown in Figures 6 and 7, each robot or agent in MRS sends the sensor data from a node in the sensing component to a node in the planning component through a topic. After that, receive the localization result sent from the planning component to the actuation component through a topic. In this study, to find the optimal value of DEADLINE and DEPTH, in the first step, we need to identify the total topic used to transmit the data sample from each agent.

$$T = \sum_{i=1}^{n} t_i \tag{1}$$

where T is the total topic of each agent, t is the topic used to send sensor data and receive the localization result, and n is the number of topics (1, 2, 3,..., n). After getting the total topic from each agent, calculate the average topic used to transmit the data sample in MRS.

$$T_{avg} = (\sum_{a=1}^{A} T_a)/A \tag{2}$$

where T_{avg} is the average of topics in MRS, and A is the number of agents (1, 2, 3,..., A). In this optimization, our idea to find the optimal value of DEADLINE R is to divide the maximum data transmission rate R_{max} by the average topic used to transmit the data sample in MRS. Where R_{max}

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

is the maximum rate when only one topic is used to transmit the data sample in MRS, furthermore, create our idea to determine the DEADLINE R with the equation:

$$R = \frac{Rmax}{T_{avg}} \tag{3}$$

In Eq. 3, we divide the maximum data transmission rate $Rmax$ by the average topic in MRS T_{avg} to balance the rates with all topics to transmit the data sample between the publisher and the subscriber in MRS. Based on the idea of Eq. 3, we create the first constraint in this optimization to find the optimal value of DEADLINE R with:

$$\frac{Rmax}{T_{avg}} - R \geq Rmin \tag{4}$$

In the first constraint, the optimal value of DEADLINE R should be greater than or equal to the minimum data transmission rate $Rmin$, which means that the transmission of the data sample between the publisher and the subscriber is satisfied when the optimal value of DEADLINE is greater than or equal to the minimum data transmission rate. The next step is to find the optimal value of DEPTH D to determine the buffer size. Our idea here to find the optimal value of the DEPTH is to balance it with a DEADLINE tune. If the DEADLINE tune is large and close to the maximum rate, the DEPTH tune will also be large and close to the maximum buffer size $Dmax$. Otherwise, if the DEADLINE tune is low, the DEPTH tune will also be low and close to the minimum buffer size $Dmin$. $Dmax$ is the maximum buffer size to store the data sample in the publish-subscribe communication, that is, 5000 [17]. Based on this idea, we create the second constraint in this optimization to find the optimal value of DEPTH D with:

$$Dmax \times \frac{R}{Rmax} - D \geq Dmin \tag{5}$$

Next, for the following constraints, we bound the DEADLINE variable R not to exceed or equal the maximum rate and not less than or equal to the minimum rate $Rmin \leq R \leq Rmax$. Furthermore, we bound the DEPTH variable D not to be larger than or equal to the maximum buffer size and not less than or equal to the minimum buffer size $Dmin \leq D \leq Dmax$. For the objective function of this optimization, we use the multi-objective optimization technique [18,19] to determine the maximum configuration of DEADLINE R and DEPTH D. Finally, create the optimization of this study to find the optimal value of DEADLINE R and DEPTH D to ensure their QoS balance with:

$$\begin{aligned} \max \quad & R, D \\ \text{s.t.} \quad & \frac{Rmax}{T_{avg}} - R \geq Rmin \\ & Dmax \times \frac{R}{Rmax} - D \geq Dmin \\ & Rmin \leq R \leq Rmax \\ & Dmin \leq D \leq Dmax \end{aligned} \tag{6}$$

In this study, we maximize the DEADLINE R in the objective function to make the data transfer rates in MRS high and close to the real-time data transfer. Then maximize the buffer size in DEPTH D to adjust the change in the data transfer rate between the publisher and the subscriber. Furthermore, to implement the optimization, we used CVXPY to solve the problem in our optimization. CVXPY is an open source Python-embedded modeling language to solve the problem in convex optimization [20].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

3. Results

3.1. Experimental Result in Actual Machine

This experiment analyzes the performance of MRS data transmission when the robot computer/machine sends various sensor data with local cache, cache control, and QoS balancing optimization in the ARP architecture. We did this experiment in a static environment with objects moving around the sensor devices to get various data values or conditions sent from the sensor devices to the robot machine. To perform the experiment, we used three Raspberry Pi 4 as robot machines in MRS with a Quad-Core Cortex A72 (ARM v8) processor @ 1.5GHz and 8GB of memory, respectively. Then a laptop computer with an Intel Core i5 processor @ 2.60GHz x 4 and 12GB memory as a CEDDP. The OS installed on the MRS machine and CEDDP is a Linux Ubuntu 20.04 LTS, ROS 2 Foxy Fitzroy for robotic software, and Fast-RTPS DDS for ROS 2 node communication in the ARP network.

In this experiment, we used several sensor devices connected to each Raspberry Pi as an actual machine in MRS. The sensor devices connected to actual machine 1 are LIDAR (SLAMTEC, A2M8), light detection sensor (VKLSVAN, photosensitive sensor module), IR flame detection sensor (AYNEF, flame sensor module), temperature and humidity sensor (HiLetgo, DHT11), IMU sensor (KKHMF, MPU-6050), and PIR sensor (VKLSVAN, HC-SR501). Furthermore, for machines 2 and 3, the sensor devices connected to the machines are the IR flame detection sensor (AYNEF, flame sensor module), PIR sensor (VKLSVAN, HC-SR501), ultrasonic distance sensor (ELEGOO, HC-SR04), temperature and humidity sensor (HiLetgo, DHT11), and light detection sensor (VKLSVAN, photosensitive sensor module). Figure 8 shows the experimental setup and the sensor devices connected to the machines.

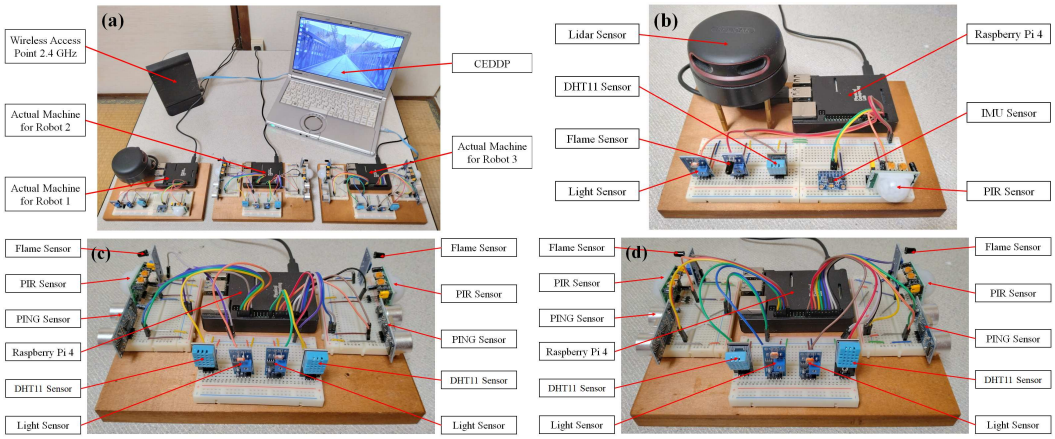

Figure 8. (a) Experimental setup on the actual machine; **(b)** Sensor devices connected to the actual machine 1; **(c)** Sensor devices connected to the actual machine 2; **(d)** Sensor devices connected to the actual machine 3.

Table 1 shows the sensor devices, message data type, and data size used in our experiment. The data type used in our experiment to transmit sensor data from nodes in the sensing component to a node in the planning component is the default data type of the sensor output (LIDAR, Flame, DHT11, IMU, and Ultrasonic). Furthermore, for the PIR and light sensors, we used a string-type message with "detect" or "not detect" if the PIR sensor output is "1" or "0", then "light" or "dark" when the light sensor output is "1" or "0", respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

Table 1. Sensor devices, message data type, and data size.

Sensor Devices	Message Data Type	Data Size (Bytes)
LIDAR	Float	24
Flame	Boolean	28
DHT11	Float	24
IMU	Float	24
Ultrasonic	Float	24
PIR	String	59
Light	String	54

Furthermore, Table 2 shows the QoS configurations used in the experiment. In this study, we focus on analyzing the performance of MRS data transmission when the RELIABLE and KEEP_LAST options were used to transmit robot data in ROS 2 node communication. We analyze it because strict reliability is not guaranteed if the DDS as a core for ROS 2 node communication uses the RELIABLE and KEEP_LAST options to transmit the data samples between the publisher and the subscriber.

Table 2. Configuration of QoS policies in the experiment.

QoS Policies	Options
RELIABILITY	RELIABLE
HISTORY	KEEP_LAST
DEPTH	1, 5, 10, 100, 1000, 5000, Opt (D)
DEADLINE	100 Hz, 200 Hz, 500 Hz, 1000 Hz, Opt (R)
DURABILITY	VOLATILE
LIVELINESS	AUTOMATIC

Figure 9 shows the illustration of this actual machine experiment. In the experiment, each robot machine sends the sensor data information from the sensing component nodes to the panning component node in CEDDP with local cache, cache control, and QoS balancing optimization runs on those sensor data transmissions, respectively. After that, analyze the efficiency of local cache, cache control, and QoS balancing optimization to improve the performance of MRS data transmission when robot1 machine successfully sent and received the "Hello" message data transmitted from a node in the sensing component to a node in the actuation component through the localization result of MRS in CEDDP and robot3 machine.

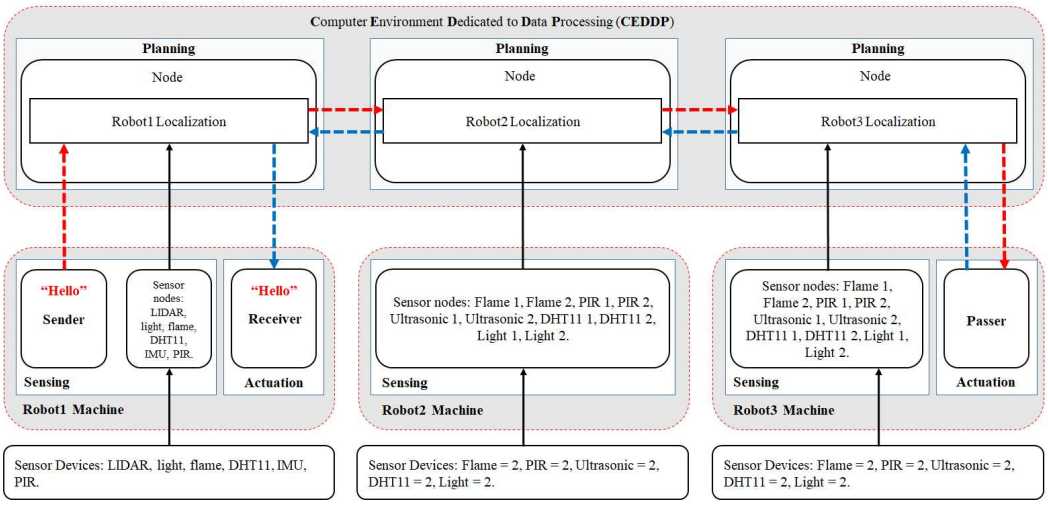

Figure 9. Experimental illustration in actual machine.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

To analyze the performance of MRS data transmission, we measure the latency and calculate the total packet loss of the "Hello" message data transmitted in MRS communication. Figures 10 and 11 show the latency and packet loss analysis results, respectively. Based on the results of latency and packet loss, it can be seen that the combination of local cache and QoS balancing optimization effectively improves latency and reduces packet loss in MRS data transmission, compared to the situation without the implementation of optimization and combined with cache control. Furthermore, from those results we conclude that if DEADLINE configures the rates with high frequency and DEPTH configures the buffer with a small size, it can increase the latency of data transmission and affect packet loss in MRS data transmission.

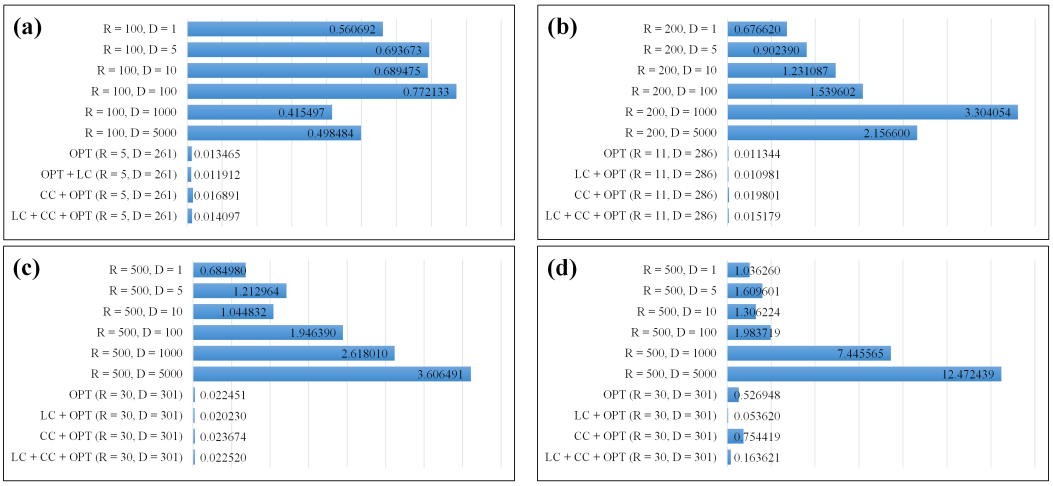

Figure 10. (a) Latency when the maximum data transmission rate R_{max} = 100 Hz; **(b)** Latency when the maximum data transmission rate R_{max} = 200 Hz; **(c)** Latency when the maximum data transmission rate R_{max} = 500 Hz; **(d)** Latency when the maximum data transmission rate R_{max} = 1000 Hz.

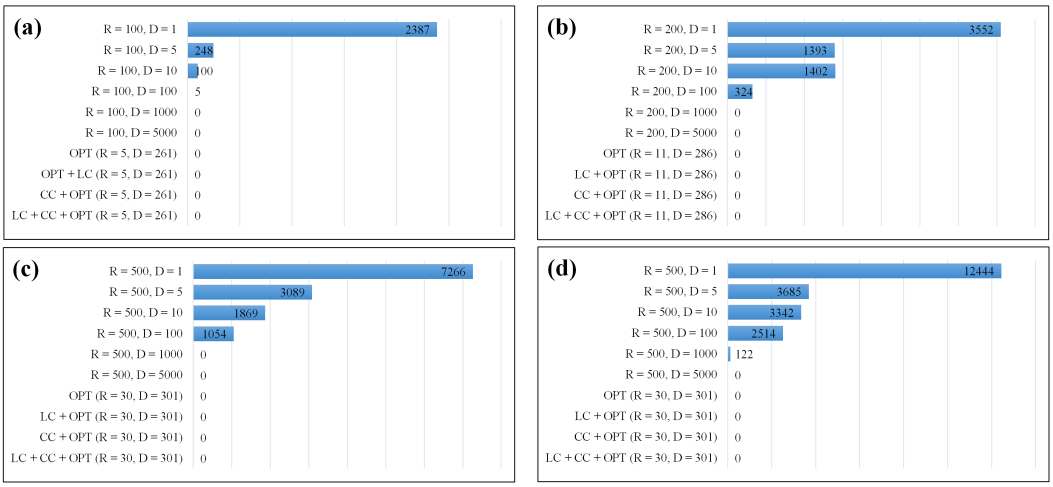

Figure 11. (a) Packet loss when the maximum data transmission rate R_{max} = 100 Hz; **(b)** Packet loss when the maximum data transmission rate R_{max} = 200 Hz; **(c)** Packet loss when the maximum data transmission rate R_{max} = 500 Hz; **(d)** Packet loss when the maximum data transmission rate R_{max} = 1000 Hz.

3.2. Experimental Result in Simulation

This experiment was carried out to analyze the performance of MRS cooperation in simulation when local cache, cache control, and QoS balancing optimization were used to improve robot data communication in the ARP architecture. To implement the experiment, we used three Raspberry Pi 4

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

as robot computers to perform the robot simulation, CEDDP to process MRS data and communicate MRS, and access point 2.4 GHz to transmit robot data between the Raspberry Pi and CEDDP. The OS installed on the Raspberry Pi and CEDDP is Linux Ubuntu 20.04 LTS, ROS 2 Foxy Fitzroy for robotic software, and the Gazebo application for robot simulation. Figure 12 shows the experimental setup in simulation, Figure 13 shows the robot design, and Figure 14 presents the MRS tasks in the experiment.

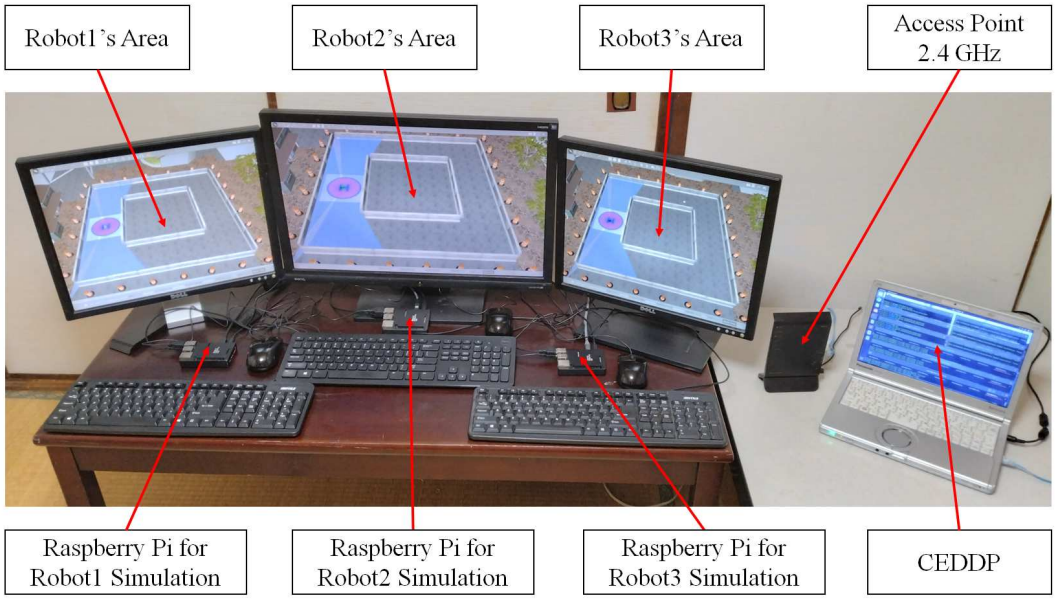

Figure 12. Experimental setup in simulation.

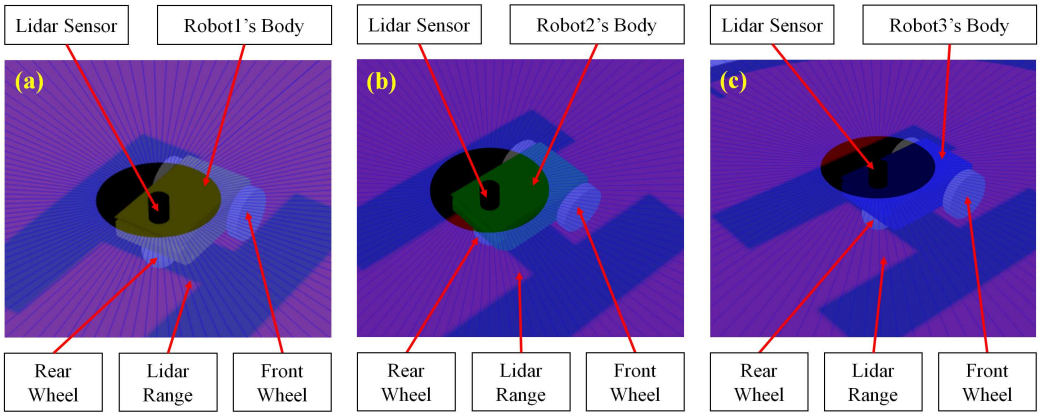

Figure 13. (a) Robot1 design; **(b)** Robot2 design; **(c)** Robot3 design.

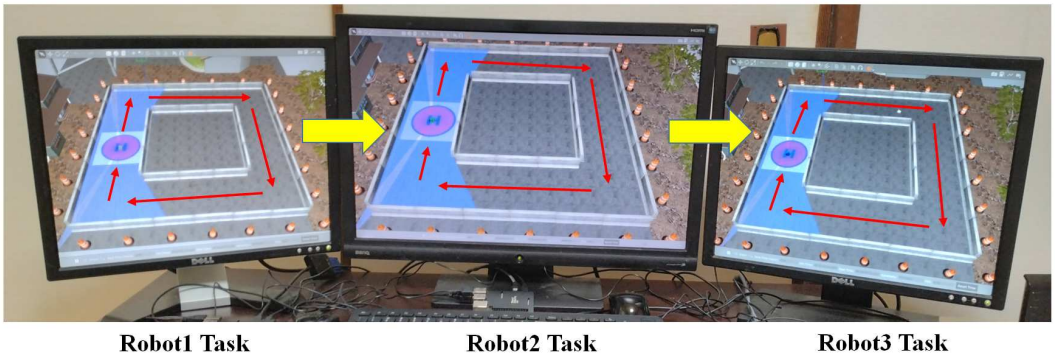

Figure 14. Multi-robot tasks in the experiment.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

Based on the experimental setup shown in Figure 12, three Raspberry Pis were used to run the robot simulation using the Gazebo application, then CEDDP to process the LIDAR data for robot localization and to communicate the MRS based on the localization results from each robot. Raspberry Pi and CEDDP can exchange MRS data through a wireless network. Furthermore, according to the MRS design shown in Figure 13, we used three mobile robots with the same design and specifications, and each robot was equipped with a LIDAR sensor to perceive the environment. To analyze the performance of MRS cooperation, we have designed the MRS task based on the robot path shown in Figure 14. The task of MRS in this simulation is to move the robots by parallels in different areas based on the robot1 moves as a leader in MRS. Furthermore, the velocity of MRS moving in the simulation is constant. Figure 15 shows the experimental illustration in the MRS simulation.

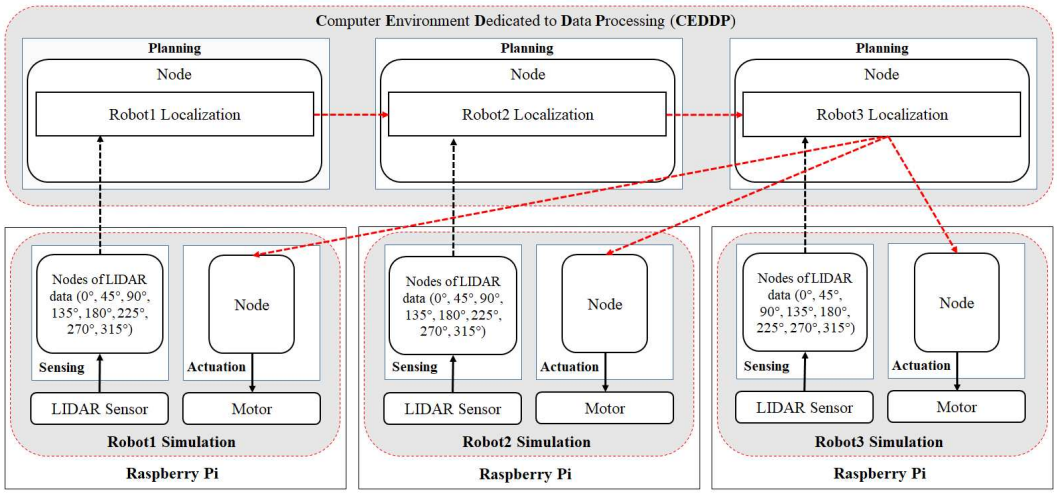

Figure 15. Experimental illustration in MRS simulation.

Referring to the experimental illustration shown in Figure 15, each robot in the simulation sends eight LIDAR sensor data to CEDDP for robot localization through eight nodes, respectively. The LIDAR data sent from each robot to CEDDP for robot localization are 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. The data type used to transmit the LIDAR data from the sensing to the planning components was float data type with a capacity of 24 Bytes, respectively. For the QoS configuration, we analyze the performance of MRS cooperation when the QoS policy for ROS 2 node communication was configured based on the QoS configurations shown in Table 2.

In the experiment, after robot1 sends eight LIDAR data to the planning node in the CEDDP, the planning node will process the LIDAR data to determine the MRS moves in the environment in robot1 localization. After that, send the localization result to robot2's and robot3's localization, respectively. In this MRS cooperation, if robot2's and robot3's localization was not detected as an obstacle in their areas, the robot3 localization then sent a command to all robots to move based on the navigation results sent from robot1. This MRS task was performed until the MRS successfully navigated the hallway in one round.

Figure 16 shows the simulation results when the MRS was communicated without optimization, with optimization (OPT), and combined with the local cache (LC) and cache control (CC). Based on the simulation results shown in Figure 16, it can be seen that QoS balancing optimization combined with a local cache method effectively improves the performance of MRS cooperation by successfully completing the tasks. Furthermore, Figure 17 shows the results of the MRS simulation when the robots successfully navigated the hallway in one round. It can be seen that the MRS successfully kept the moves in a parallel way in the environment. Then Figure 18 shows the results of the MRS simulation when the robots failed to navigate the hallway in one round. Based on the information in Figure 18, the MRS cannot complete the task because robot2 detected an obstacle when robot2 navigated its

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

environment. This happened because the data transmission rate and buffer size were unstable in MRS communication.

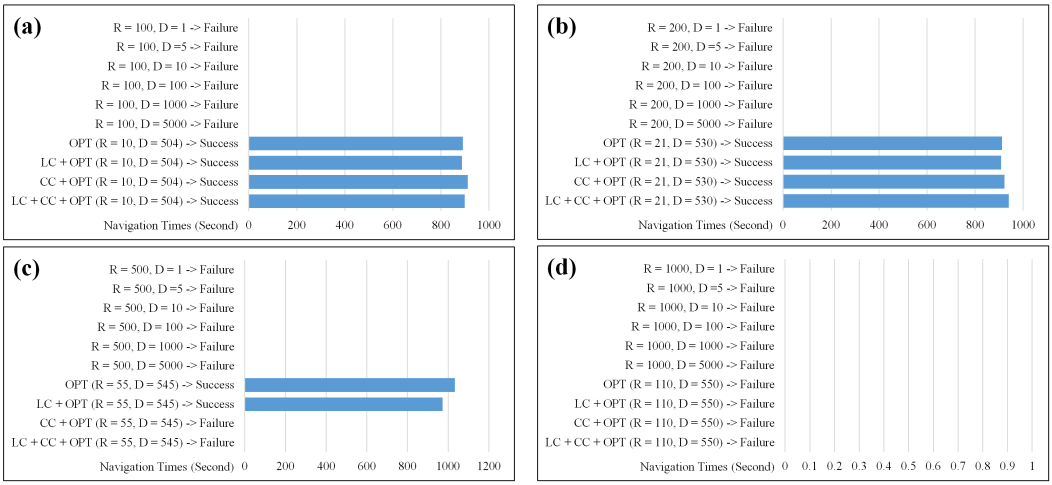

Figure 16. **(a)** Simulation result when R_{max} = 100 Hz; **(b)** Simulation result when R_{max} = 200 Hz; **(c)** Simulation result when R_{max} = 500 Hz; **(d)** Simulation result when R_{max} = 1000 Hz.

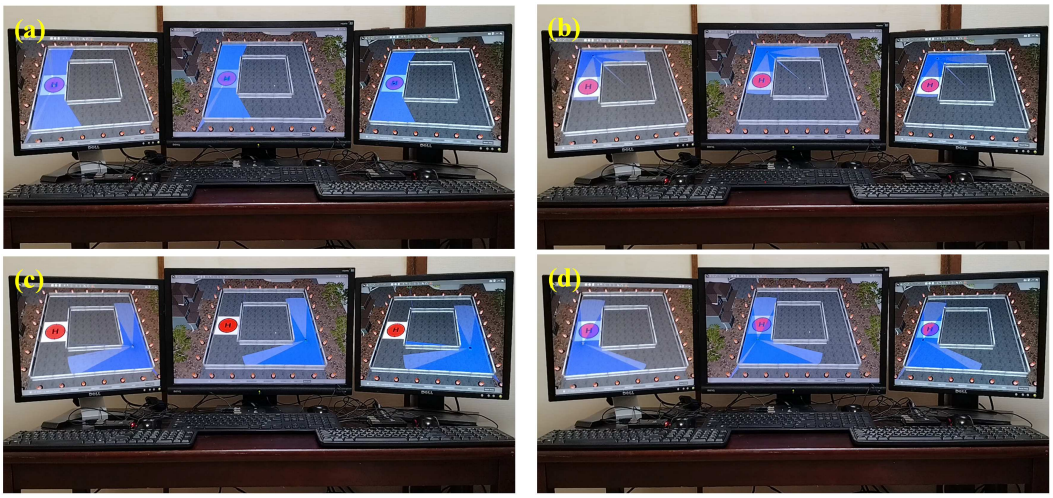

Figure 17. **(a)** MRS ready to navigate the hallway; **(b)** MRS turned right in the hallway; **(c)** MRS successfully navigated the hallway; **(d)** MRS successfully returned home.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2023 doi:10.20944/preprints202305.0431.v1

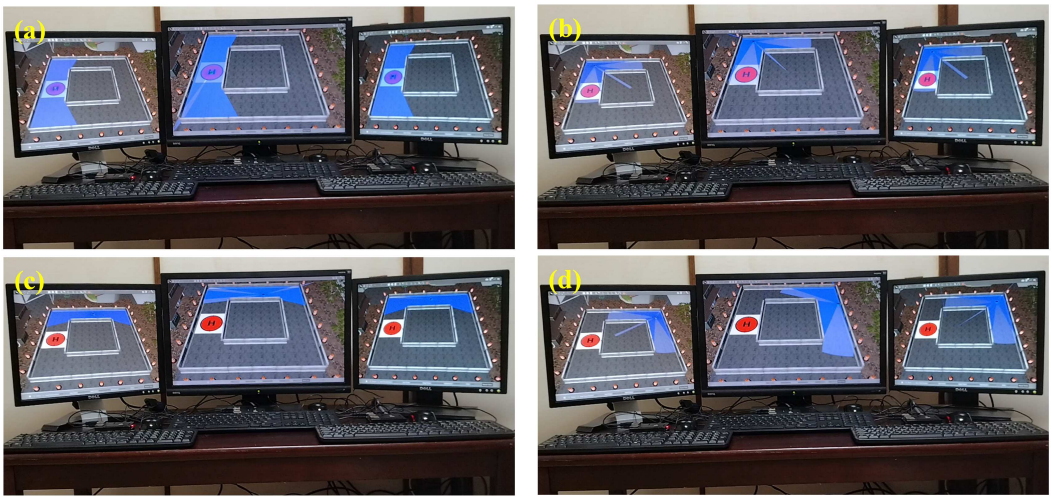

Figure 18. **(a)** MRS ready to navigate the hallway; **(b)** MRS turned right in the hallway; **(c)** MRS failed to navigate the hallway; **(d)** MRS failed to complete the task.

4. Discussion

This study analyzed the efficiency of local cache, cache control, and QoS balancing optimization to improve the performance of MRS data communication in the ARP architecture. We analyze it when ROS 2 for robotic software in this study uses the QoS RELIABLE and KEEP_LAST options to transmit the robot data between the publisher and the subscriber. In ROS 2 node communication, a RELIABLE connection guarantees that all packet data can be sent properly from the publisher to the subscriber. However, strict RELIABILITY is not guaranteed if the RELIABLE connection uses the KEEP_LAST option to transmit the data sample. This occurs when the data transfer rate between the publisher and the subscriber is too high and the buffer size is not large enough to store old data samples, affecting packet loss in a RELIABLE connection [4–9].

Our idea in the local cache method is to streamline the sensor data transmission between sensing and planning components when the robotic sensor produces the data output with the same value in a row. This method can reduce the heartbeat and acknowledgment rates in a RELIABLE connection and help the buffer not store the data sample with the same value in a row, so it can make more space in the data queue stored in the buffer. Furthermore, our idea in the cache control method is to streamline sensor data transmission in CEDDP before processing it in the planning component. We streamline it to improve MRS communication in the ARP architecture by reducing the planning component work to not receive the sensor data directly from the sensing component and focusing only on processing the robot data and communicating the MRS in CEDDP. For QoS balancing optimization, our idea in this optimization is to improve MRS communication by balancing the data transmission rates and buffer size in publish-subscribe communication, configured in QoS DEADLINE for the rates and DEPTH for buffer size, respectively.

We have analyzed the performance of MRS communication in the ARP architecture when the robot data are transmitted without caches and optimization, with local cache, with cache control, with optimization, and combined between local cache, cache control, and optimization. We analyze it when MRS data transmission was implemented in actual machines, and MRS cooperation was implemented in the robot simulation using Gazebo. The results of this study show that the combination of local cache and QoS balancing optimization effectively improves the latency of MRS data transmission and reduces packet loss in a RELIABLE connection. Furthermore, QoS balancing optimization combined with the local cache method improves MRS cooperation compared to when the MRS is communicated without optimization and cache control. However, our proposed study is effective depending on the task, computer performance, wireless communication speed, number of sensors, and sensor types used in our experiment.

In the future, we will analyze the efficiency of local cache, cache control, and QoS balancing optimization when implementing in real multi-robot cooperation with a different area and distance. Then analyze it when implemented on different computer performances, network types, and Internet of Things technology.

5. Conclusions

In this study, we have analyzed the efficiency of local cache, cache control, and QoS balancing optimization to improve MRS communication and cooperation in ARP architecture when ROS 2 was used to transmit robot data with QoS RELIABLE and KEPP_LAST options. The result of this study shows that the combination of local cache and QoS balancing optimization effectively improves the latency of data transmission in MRS, reduces packet loss, and improves the performance of MRS cooperation. The idea in the local cache is to streamline the sensor data transmission when the robotic sensor produces the data output with the same value in a row. Then the QoS balancing optimization to determine the rates and buffer size with the balancing settings configured in DEADLINE and DEPTH of the QoS policies in ROS 2. For the next study, we will analyze the performance of local cache, cache control, and QoS balancing optimization when implemented in a real Multi-Robot communication.

Author Contributions: Conceptualization, A.J. and J.K.; methodology, A.J. and J.K.; software, A.J.; validation, T.S.; formal analysis, A.J.; investigation, A.J.; resources, J.K. and T.S.; data curation, A.J.; writing—original draft preparation, A.J.; writing—review and editing, A.J. and T.S.; visualization, A.J. and T.S.; supervision, J.K. and T.S; project administration, A.J.; funding acquisition, T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gautam, A.; Mohan, S. A review of research in multi-robot systems. 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), Chennai, India, Date of Conference (06-09 August 2012). DOI: 10.1109/ICIInfS.2012.6304778.
2. Maruyama, Y.; Kato, S.; Azumi, T. Exploring the Performance of ROS2. 2016 International Conference on Embedded Software (EMSOFT), Pittsburgh, PA, USA, Date of Conference (02-07 October 2016). DOI: https://doi.org/10.1145/2968478.2968502.
3. About Quality of Service settings. Available online: https://docs.ros.org/en/foxy/Concepts/About-Quality-of-Service-Settings.html (accessed on April 12, 2023).
4. Fernandez, J.; Allen, B.; Thulasiraman, P.; Bingham, B. Performance Study of the Robot Operating System 2 with QoS and Cyber Security Settings. 2020 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, Date of Conference (24 August 2020 - 20 September 2020). DOI: 10.1109/SysCon47679.2020.9275872.
5. Thulasiraman, P.; Chen, Z.; Allen, B.; Bingham, B. Evaluation of the Robot Operating System 2 in Lossy Unmanned Networks. 2020 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, Date of Conference (24 August 2020 - 20 September 2020). DOI: 10.1109/SysCon47679.2020.9275849.
6. Chen, Z. Performance Analysis of ROS 2 Networks Using Variable Quality of Service and Security Constraints for Autonomous Systems. Thesis, NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIFORNIA, September 2019.
7. Park, J.; Delgado, R.; Choi, B.W. Real-Time Characteristics of ROS 2.0 in Multiagent Robot Systems: An Empirical Study. *IEEE Access* **2020**, *8*, 154637–154651. DOI: 10.1109/ACCESS.2020.3018122.
8. Jalil, A.; Kobayashi, J. Efficacy of Local Cache for Performance Improvement of Reliable Data Transmission in Aggregated Robot Processing Architecture. 2022 22nd International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, Date of Conference (27 November 2022 - 01 December 2022). DOI: 10.23919/ICCAS55662.2022.10003765.

9. Jalil, A; Kobayashi, J; Saitoh, T. Optimization Algorithm for Balancing QoS Configuration in Aggregated Robot Processing Architecture. The 2023 International Conference on Artificial Life and Robotics (ICAROB2023), Oita, Japan, Date of Conference (9-12 February 2023).
10. Choi, H.; Xiang, Y.; Kim, H. PiCAS: New Design of Priority-Driven Chain-Aware Scheduling for ROS2. 2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS), Nashville, TN, USA, Date of Conference (18-21 May 2021). DOI: 10.1109/RTAS52030.2021.00028.
11. Wang, Y.P.; Tan, W.; Hu, X.Q.; Manocha, D.; Hu, S.M. TZC: Efficient Inter-Process Communication for Robotics Middleware with Partial Serialization. *arXiv:1810.00556 [cs.RO]* **2020**. DOI: https://doi.org/10.48550/arXiv.1810.00556.
12. Randolph, C. Improving the Predictability of Event Chains in ROS 2. Master of Science Thesis in Embedded Systems, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands, 15-3-2021.
13. Jiang, Z.; Gong, Y.; Zhai, J.; Wang, Y.P.; Liu, W.; Wu, H.; Jin, J. Message Passing Optimization in Robot Operating System. *International Journal of Parallel Programming* **2020**, *48*, 119–136. DOI: 10.1007/s10766-019-00647-w.
14. Jalil, A.; Kobayashi, J. Experimental Analyses of an Efficient Aggregated Robot Processing with Cache-Control for Multi-Robot System. 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea (South), Date of Conference (13-16 October 2020). DOI: 10.23919/ICCAS50221.2020.9268225,
15. Staschulat, J.; Lütkebohle, I.; Lange, R. The rclc Executor: Domain-specific deterministic scheduling mechanisms for ROS applications on microcontrollers: work-in-progress. 2020 International Conference on Embedded Software (EMSOFT), Shanghai, China, Date of Conference (20-25 September 2020). DOI: 10.1109/EMSOFT51651.2020.9244014.
16. Understanding topics. Available online: https://docs.ros.org/en/foxy/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Topics/Understanding-ROS2-Topics.html (accessed on April 20, 2023).
17. eProsima. *Fast DDS Documentation*, Release 2.10.1. Apr 25, 2023.
18. Gunantara, N. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering* **2018**, *5:1*, 1–16. DOI: 10.1080/23311916.2018.1502242.
19. Ehrgott, M. *Multicriteria optimization*, Germany: Springer, 2005.
20. Diamond, S.; Boyd, S. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research* **2016**, *17*, 2909–2913.