Optimization Algorithm for Balancing QoS Configuration in Aggregated Robot Processing Architecture

Abdul Jalil[*], Jun Kobayashi, Takeshi Saitoh
Kyushu Institute of Technology,
680-4 Kawazu, Iizuka-shi, Fukuoka, 820-8502, Japan
E-mail: malla.abdul-jalil545@mail.kyutech.jp, jkoba@ics.kyutech.ac.jp, saitoh@ai.kyutech.ac.jp.
www.kyutech.ac.jp

Abstract

Quality of Service (QoS) manages the data traffic to reduce packet loss, latency, and jitter in the network. This study aims to design an optimization algorithm to find the balance of QoS configuration to set the rates and buffer size while the robot data processes are communicated in the Aggregated Robot Processing (ARP) architecture. This study implements optimization to manage the DEPTH and DEADLINE QoS configuration in Robot Operating System 2 (ROS 2) node communication. Unbalancing DEPTH and DEADLINE configurations can affect the high latency time of message data transmission and packet loss in RELIABLE connections. The results of this study show that the optimization algorithm can determine the optimal value of DEPTH and DEADLINE by balancing the QoS configuration to improve the robot data transmission in the ARP architecture.

Keywords: Optimization, Quality of Service, ROS 2, Aggregated Robot Processing

1. Introduction

Robot data processing flows generally have three components: sensing, planning, and actuation [1]. These components can be connected as node communication in the network and use Quality of Service (QoS) policies to manage the quality of data transmission between components. QoS manages data traffic on the network to reduce packet loss, latency, and jitter. However, the unbalanced QoS configuration on the network can influence the performance of robot data transmission between components, such as packet loss and latency.

This study aims to develop an optimization algorithm to find the balancing QoS configuration of the rates and buffer size while the robot data processes are communicated in the Aggregated Robot Processing (ARP) architecture. We implement this optimization to find the optimal value of DEPTH and DEADLINE when the robot data processes between sensing, planning, and actuation components are communicated using ROS 2 (Robot Operating System 2). ROS 2 is built on top of Data Distribution Service (DDS) and uses a set of QoS policies to tune node communication [2].

Some researchers have analyzed the effectiveness of QoS policies configuration in ROS 2. They analyzed latency [2][3][4][5][6], throughput [2][4], packet loss [4][5][6][7], and memory consumption [2]. Several studies show that when DEPTH configures the buffer with a small size and DEADLINE configures the rates with high frequency, some of the packets will be lost in ROS 2 node communication [4][5][6][7]. Furthermore, when DEPTH configures the buffer with a large size, some memory space will be used in that configuration [4][8], and when DEADLINE sets the rates with a low frequency, the data transfer rate becomes low, and this will affect the real-time of the message data transfer between nodes.

The contribution of this paper is to improve robot data transmission by balancing the DEPTH and DEADLINE QoS configuration when the ROS 2 nodes transmit the data using the RELIABLE and KEEP_LAST options. We analyze it because strict reliability is not guaranteed if the DDS uses the RELIABLE and KEEP_LAST options for data transmission between nodes [9].

2. Methods

Figure 1 shows the ARP architecture developed in our study [10]. In ARP architecture, the robot data processes for localization and path planning are executed in the Computer Environment Dedicated to Data Processing (CEDDP). The robot computer function reads the sensor data in the sensing component and then sends them to the CEDDP, which also drives the robot actuator based on the path-planning result sent from the CEDDP in the actuation component. The robot computer and CEDDP can exchange data through a wireless network.

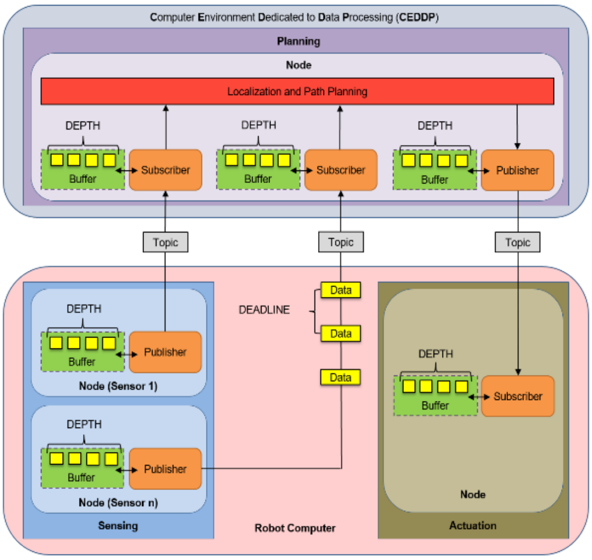

Fig.1.Aggregated Robot Processing architecture

Based on the illustration shown in Fig.1, the sensor node in the sensing component transmits sensor data to a node in the planning component through a topic SPt. Then the node in the planning component transmits the robot localization and path planning result to a node in the actuation component through a topic PAt. Therefore, our idea here to find the optimal value of DEADLINE R is to divide the maximum data transmission rate $Rmax$ by the total topic used to transmit sensor data from a node in the sensing component to a node in the planning component $\sum SPt$, added with the total topic used to transmit the data from a node in the planning component to a node in the actuation component $\sum PAt$.

$$R = \frac{Rmax}{\sum SPt + \sum PAt} \tag{1}$$

In Eq. (1) we divide $Rmax$ by $\sum SPt + \sum PAt$ to balance the rate configuration in DEADLINE R with all topics when transmitting the message data from the publisher to the subscriber. In this study, $Rmax$ is the maximum rate when only one topic is used to transfer message data from the publisher to the subscriber. Furthermore, based on the idea of Eq. (1) we create the first constraint in our optimization with the following:

$$\frac{Rmax}{\sum SPt + \sum PAt} - R \geq Rmin \tag{2}$$

In Eq. (2) the value of R should be greater than or equal to the minimum transmission rate $Rmin$, which means that the communication of message data between the publisher and the subscriber is satisfied when R is greater than or equal to $Rmin$.

Next, 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 of the DEADLINE, the DEPTH tune will also be high and close to the maximum value of DEPTH $Dmax$. Otherwise, if the DEADLINE tune is low, the DEPTH tune will also be low and close to the minimum value of the DEPTH $Dmin$. Here, $Dmax$ is the maximum queue size for storing data samples in the buffer when the KEEP_LAST option is chosen, that is, 5000 [8]. Based on this idea, we create the second constraint to find the optimal value of DEPTH with the following:

$$Dmax\,\frac{R}{Rmax} - D \geq Dmin \tag{3}$$

For the next constraints, bound the variable R not to be greater than or equal to the maximum rate and not less than or equal to the minimum rate $Rmin \leq R \leq Rmax$. After that, bound the variable D not to be greater than or equal to the maximum DEPTH and not less than or equal to the minimum DEPTH $Dmin \leq D \leq Dmax$. Finally, create the optimization equation to find the optimal value of DEPTH and DEADLINE with the following:

$$\begin{aligned} & max\ R + D \\ s.t.\ & \frac{Rmax}{\sum SPt + \sum PAt} - R \geq Rmin \\ & Dmax\,\frac{R}{Rmax} - D \geq Dmin \\ & Rmin \leq R \leq Rmax \\ & Dmin \leq D \leq Dmax \end{aligned} \tag{4}$$

In this study, we used CVXPY to implement the optimization algorithm. CVXPY is an open-source Python-embedded modeling language that solves the problem of convex optimization [11]. The following algorithm is our proposed optimization to find the optimal value of DEPTH and DEADLINE based on the optimization shown in equation 4.

```
Algorithm: Optimization algorithm
Require: $\sum SPt, \sum PAt, Rmax, Rmin, Dmax, Dmin$
Variable: $R, D$
Constraints:
    [$\frac{Rmax}{\sum SPt + \sum PAt} - R \geq Rmin$,
    $Dmax \frac{R}{Rmax} - D \geq Dmin$,
    $Dmin \leq R \leq Dmax$,
    $Dmin \leq D \leq Dmax$]
Objective_Function: $max\ R + D$
Problem: (Objective_Function, Constraints)
Problem.solve()
```

3. Results and Discussion

Figure 2 shows the experimental illustration of our study. To perform the analysis, we used Raspberry Pi 4, which has a processor of 1.5GHz with 8 GB memory for the robot computer, and a laptop computer with an Intel Core i5 @ 2.60GHz x 4 with 12 GB memory as a CEDDP. The operating system installed on these computers was Linux Ubuntu 20.04 LTS, Foxy Fitzroy for the ROS 2 distribution, and Fast-RTPS for the DDS middleware. Furthermore, we used a wireless access point with a frequency of 2.4 GHz to exchange the message data between the robot computer and CEDDP.

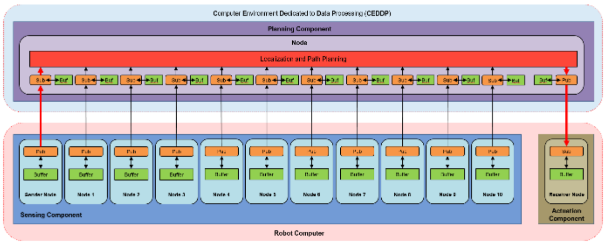

Fig.2. Experimental illustration

Based on the illustration shown in Fig.2, we analyze the performance of our optimization when the node between the sensing, planning, and actuation components is communicated based on the QoS configurations shown in Table 1.

Table 1. QoS configurations in experiments.

RELIABILITY	RELIABLE
HISTORY	KEEP_LAST
DEPTH	1, 5000, Optimization (D)
DEADLINE	100 Hz, 200 Hz, 500 Hz, 1000 Hz, Optimization (R)
DURABILITY	VOLATILE
LIVELINESS	AUTOMATIC

In the experiment, we evaluated the optimization algorithm if each node in the sensing, planning, and actuation component exchanges the message data with a size of 10 bytes, 100 bytes, and 1000 bytes. Furthermore, Table 2 shows the requirement value entered into the optimization algorithm.

Table 2. Input value to optimization algorithm.

	1st Exp	**2nd Exp**	**3rd Exp**	**4th Exp**
$\sum SPt$	11	11	11	11
$\sum PAt$	1	1	1	1
$Rmax$	100 Hz	200 Hz	500 Hz	1000 Hz
$Rmin$	1 Hz	1 Hz	1 Hz	1 Hz
$Dmax$	5000	5000	5000	5000
$Dmin$	1	1	1	1
Opt (D)	365	390	405	410
Opt (R)	7 Hz	15 Hz	40 Hz	82 Hz

Based on the information shown in Fig.2 and Table 2, $\sum SPt = 11$ is the total of topics used to transmit data from 11 nodes in the sensing component to a node in the planning component. $\sum PAt = 1$ is the total number of topics used to transmit data from a node in the planning component to a node in the actuation component. $Rmax$ = 100 Hz, 200 Hz, 500 Hz, and 1000 Hz is the maximum rate when only one topic is used to transmit the data between the publisher and the subscriber, respectively. $Rmin$ = 1 Hz is the minimum data transmission rate, $Dmax$ = 5000 is the maximum queue to store the data sample in the buffer, and $Dmin$ = 1 is the minimum queue in the buffer. Furthermore, Opt (D) and Opt (R) are the optimization results of DEPTH D and DEADLINE R.

In this study, we analyze the optimization efficiency by measuring the latency time of message data transmission and calculating the packet loss while the sender node in the sensing component sends the message data to the receiver node through a node in the planning component. We measured the latency by calculating the time that elapsed until the receiver node in the actuation component received the message data sent from the sender node. Figure 3 shows the latency result in this experiment.

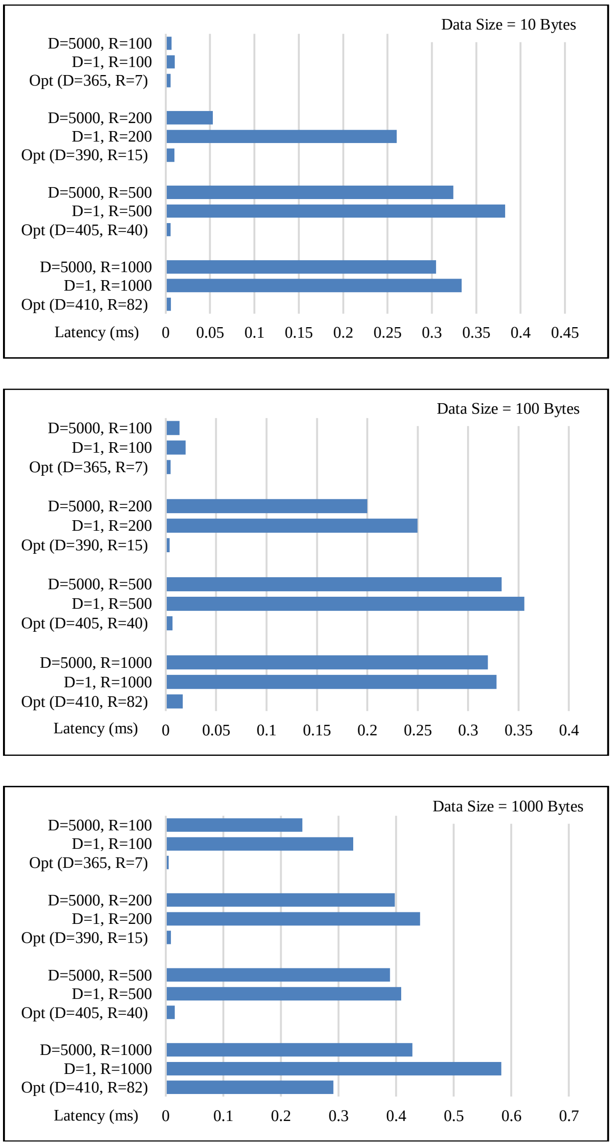

Fig.3. Latency results.

Based on the analysis results shown in Fig.3, the balance of the QoS configuration can improve the latency of message data transmission compared to when the node transmits the data with the maximum rate in the DEADLINE and the maximum/minimum configuration in the DEPTH. Furthermore, Fig.4 shows the packet loss when the sender node transmits the data to the receiver node through a node in the planning component.

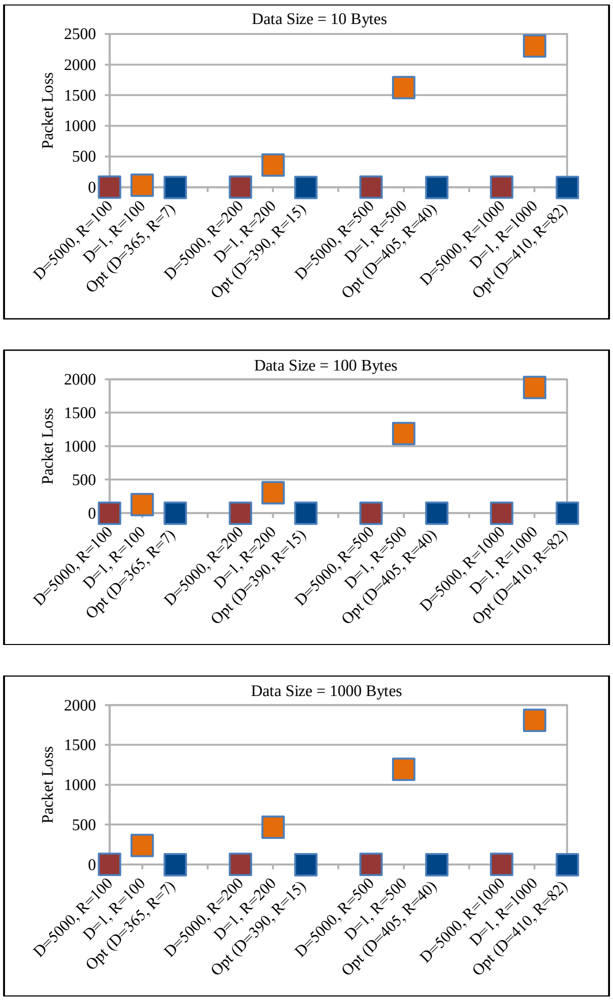

Fig.4. Packet loss results.

Based on the information results shown in Fig.4, the unbalanced QoS configuration increased the packet loss in node communication. Furthermore, the balance of QoS configuration can reduce packet loss.

4. Conclusion

This study has designed an optimization algorithm to find the balance of DEPTH and DEADLINE QoS configuration to improve ROS 2 node communication in the ARP architecture. The balance of DEPTH and DEADLINE QoS policies shown in this study can improve the latency time of message data transmission and reduce packet loss in a RELIABLE connection. Next, we will implement this optimization algorithm to improve the multi-robot data transmission in ARP architecture.

References

1. J. Staschulat, I. Lütkebohle, and R. Lange, “The rclc Executor: Domain-specific deterministic scheduling mechanisms for ROS applications on microcontrollers: work-in-progress”, International Conference on Embedded Software, EMSOFT, IEEE, 2020.
DOI: 10.1109/EMSOFT51651.2020.9244014.
2. Y. Maruyama, S. Kato, and T. Azumi, “Exploring the Performance of ROS2”, International Conference on Embedded Software, EMSOFT, IEEE, 2016.
DOI: 10.1145/2968478.2968502.
3. T. Kronauer, J. Pohlmann, M. Matthé, T. Smejkal, and G.~Fettweis, “Latency Analysis of ROS2 Multi-Node Systems”, International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), IEEE, 2021.
DOI: 10.1109/MFI52462.2021.9591166.
4. J. Fernandez, B. Allen, P. Thulasiraman, and B. Bingham, “Performance Study of the Robot Operating System 2 with QoS and Cyber Security Settings”, International Systems Conference (SysCon), IEEE, 2020.
DOI: 10.1109/SysCon47679.2020.9275872.
5. P. Thulasiraman, Z. Chen, B. Allen, and B. Bingham, “Evaluation of the Robot Operating System 2 in Lossy Unmanned Networks”, International Systems Conference (SysCon), IEEE, 2020.
DOI: 10.1109/SysCon47679.2020.9275849.
6. Z. Chen, “Performance Analysis of ROS 2 Networks Using Variable Quality of Service and Security Constraints for Autonomous Systems”, Naval Postgraduate School, 2019.
7. J. Park, R. Delgado and B.W. Choi, “Real-Time Characteristics of ROS 2.0 in Multiagent Robot Systems: An Empirical Study”, IEEE Access, vol. 8, pp. 154637-154651, 2020.
DOI: 10.1109/ACCESS.2020.3018122.
8. eProsima, “Fast DDS Documentation”, Release 2.8.1, December 2022.
9. https://community.rti.com/static/documentation/connext-dds/5.2.0/doc/manuals/connext_dds/html_files/RTI_ConnextDDS_CoreLibraries_UsersManual/Content/UsersManual/HISTORY_QosPolicy.htm.
10. A. Jalil and J. Kobayashi, “Experimental Analyses of an Efficient Aggregated Robot Processing with Cache-Control for Multi-Robot System”, 20th International Conference on Control, Automation and Systems (ICCAS), IEEE, pp. 1105-1109, 2020.
DOI: 10.23919/ICCAS50221.2020.9268225.
11. S. Diamond and S. Boyd, “CVXPY: A Python-Embedded Modeling Language for Convex Optimization”. Journal of Machine Learning Research, Vol. 17, Num 83, pp. 1-5, 2016.

Authors Introduction

Mr. Abdul Jalil

He received Master’s and Bachelor’s degree from the Department of Computer Systems, STMIK Handayani Makassar, Indonesia in 2015 and 2012, respectively. Currently, he is a Doctoral student at Department of Computer Science and Systems Engineering, Kyushu Institute of Technology, Japan.

Dr. Jun Kobayashi

He is an Associate Professor at Department of Intelligent and Control Systems, Kyushu Institute of Technology. He obtained his PhD degree at Kyushu Institute of Technology in 1999. Currently, his main interests are Robotics, Physiological Computing, and Cybernetic Training.

Prof. Takeshi Saitoh

He is Professor of Department of Artificial Intelligence, Kyushu Institute of Technology. He received B.Eng., M.Eng., and D.Eng. degrees from Toyohashi University of Technology, Japan. His research field includes image processing and pattern recognition.