IN4390 QEES GROUP 33 ASSIGNMENT1

Yuhang Tian
5219728

Mingyu Gao
5216281

December 10, 2020

1 Question 1

1.1 Experiment Explanation

In order to obtain the performance of FastRTPS transmitting policy, we tested it with 15 different sizes of data, varying from 256 bytes to 4M bytes, and recorded the transport latency between two ROS_2 nodes, and we did 300 repetitions for each group. We used Python with Pandas library to visualize the results.

1.2 Result Analysis

According to figure 1,

- When the data size is less than or equal to 64K bytes, the increasing of data size influence non-significantly on the distribution of the results.
- When the data size is greater than 64K bytes, with the increasing of data size, the variance of the results becomes larger.
- Comparing to the small data size, the latency of the larger data size is hard to predict.

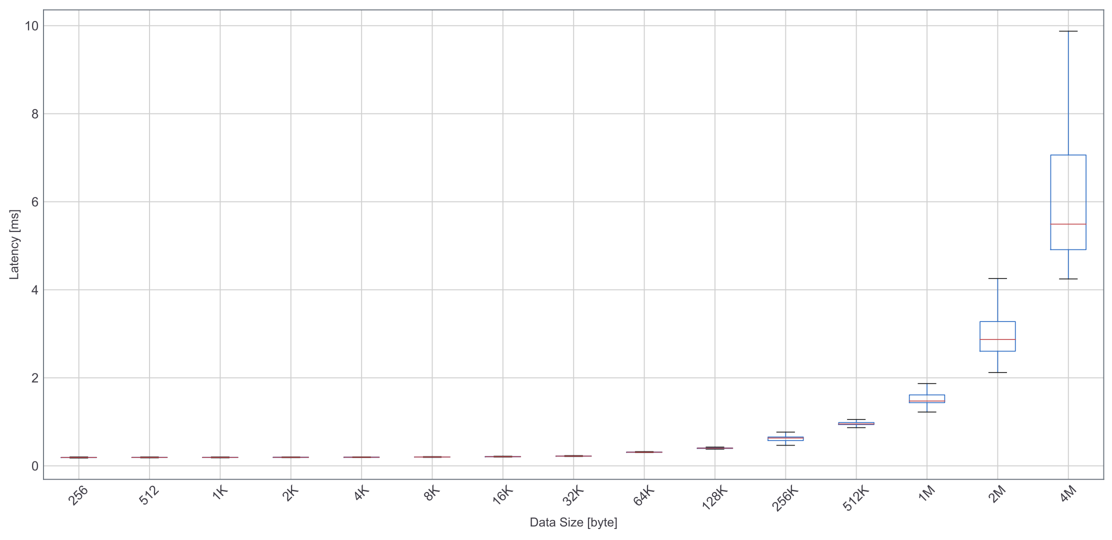

Figure 1: FastRTPS Policy with Various Data Sizes

Analysis: There should be a threshold, below which the size of the data has little influence on the transport latency. This phenomenon is caused by dividing the massage into the packet(s). If the message is larger than a standard packet size (64KB), in order to transport it, DDS will separate these long messages into several parts and marshal each of them with a packet. Dividing and transporting multiple packets will cost more time than one alone. In addition, since the sending latency is the ratio of packet size to the bandwidth, the latency is proportional to packet size.

2 Question2

2.1 Calculation Method

The premise of calculating the confidence interval is that the sample is normally distributed. The estimated mean, standard deviation, and the number of samples are μ, σ and n, respectively. And M and var are the actual mean and variance. The equation below is the confidence level, representing the probability that actual mean locating in a specific interval.

$$P(\mu - \frac{\sigma \times var}{n} < M < \mu + \frac{\sigma \times var}{n}) = 0.98 \; or \; 0.80 \tag{1}$$

Calculation of confidence interval can be divided into 4 steps as below:

Step 1: To determine the degree of freedom as equation(2)

$$df = n - 1 = 120 \tag{2}$$

Step 2: To calculate the average μ and standard variance σ of the dataset as equation(3)&(4)

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = 9.4144 \tag{3}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} = 1.7597 \tag{4}$$

Step3: To find t-values for both confidence levels, 80% and 98%, by checking the t-distribution table as equation(5)

$$t_{1-(1-98\%)/2,\; df} = 2.5378, t_{1-(1-80\%)/2.\; df} = 1.2886, \tag{5}$$

Step4:To determine the lower and upper boundaries of the confidence interval as equation(6)&(7):

$$lower_{98\%} = \mu - t \times \frac{\sigma}{n} = 9.0373, upper_{98\%} = \mu + t \times \frac{\sigma}{n} = 9.7916 \tag{6}$$

$$lower_{80\%} = \mu - t \times \frac{\sigma}{n} = 9.2083, upper_{80\%} = \mu + t \times \frac{\sigma}{n} = 9.6206 \tag{7}$$

2.2 Result and Analysis

The interval of 98% confidence level is wider than the interval of 80% confidence level, which shows that the increase of confidence level is at the expense of a searching range of the estimated interval.

Table 1: Results for Confidence Interval

Confidential Level	Lower Bound	Upper Bound
0.8	9.2083	9.6206
0.98	9.0373	9.7916

3 Question3

3.1 Boxplot Analysis

In this question, 1,2,4M bytes data are chosen to analyze and their results are shown in Figure 2:

- Without the load, the latency distribution of real and non-real-time transportation are similar, and with the increase of the data size, both median and variance grow non-significantly larger.
- If they are loaded, in terms of real-time transmission, both maximum and the third quartile (Q3) tend to become larger with the increase of data size, but the median almost stays at the same position. The overall distribution is not notably affected by the load.
- If they are loaded, turning to non-real-time transmission, the load adversely influences the transporting quality. The distribution of non-real-time transmission is dramatically expanded.

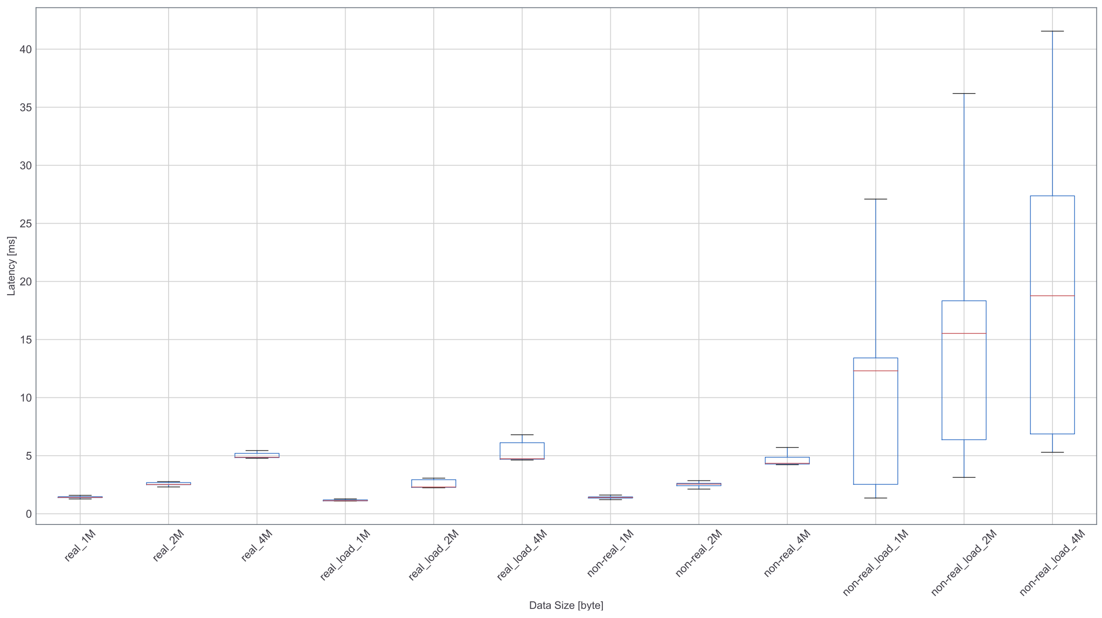

Figure 2: (non) Real-Time (un) loaded with Various Data Sizes Boxplot

Conclusion of the above observations as following:

- Without loaded, the distributions of the real-time and the non-real-time transmission are similar.
- For the real-time transmission, the effect of the loaded CPU on the transporting latency is relatively trivial.
- For non-real-time transmission, the effect of the loaded CPU significantly prolongs the latency and expands the distribution.

The reason is that the real-time transmission owns higher priorities – the real-time priorities. Therefore, although the CPU is loaded, it will put its best efforts to support the quality of the real-time transmission; whereas, without the priorities, the non-real-time transmission performs badly with the loaded CPU.

3.2 Histogram Analysis

According to figure 3, it shows four different transmitting situations, which are real-time transportation, real-time loaded transportation, non-real-time transportation, and non-real-time unloaded transportation:

- Overall, the latency is commonly increased with the growing data size for these four different transmitting categories.
- For the real-time communications without the load, the measurements are mainly concentrated at the left side of the interval, while with the data size increasing, the measurements tend to disperse to right but not significant.
- For the real-time communications with the load, the distribution of the measurements is quite similar to its unloaded one.
- For the non-real-time communications without the load, the measurements mainly locate at the Q1 position. With the increase of the data size, they tend to shift to right more.
- For the non-real-time communications with the load, the measurements locate mostly at Q1 and median, but with the increase of the data size, they tend to disperse to two extremes.

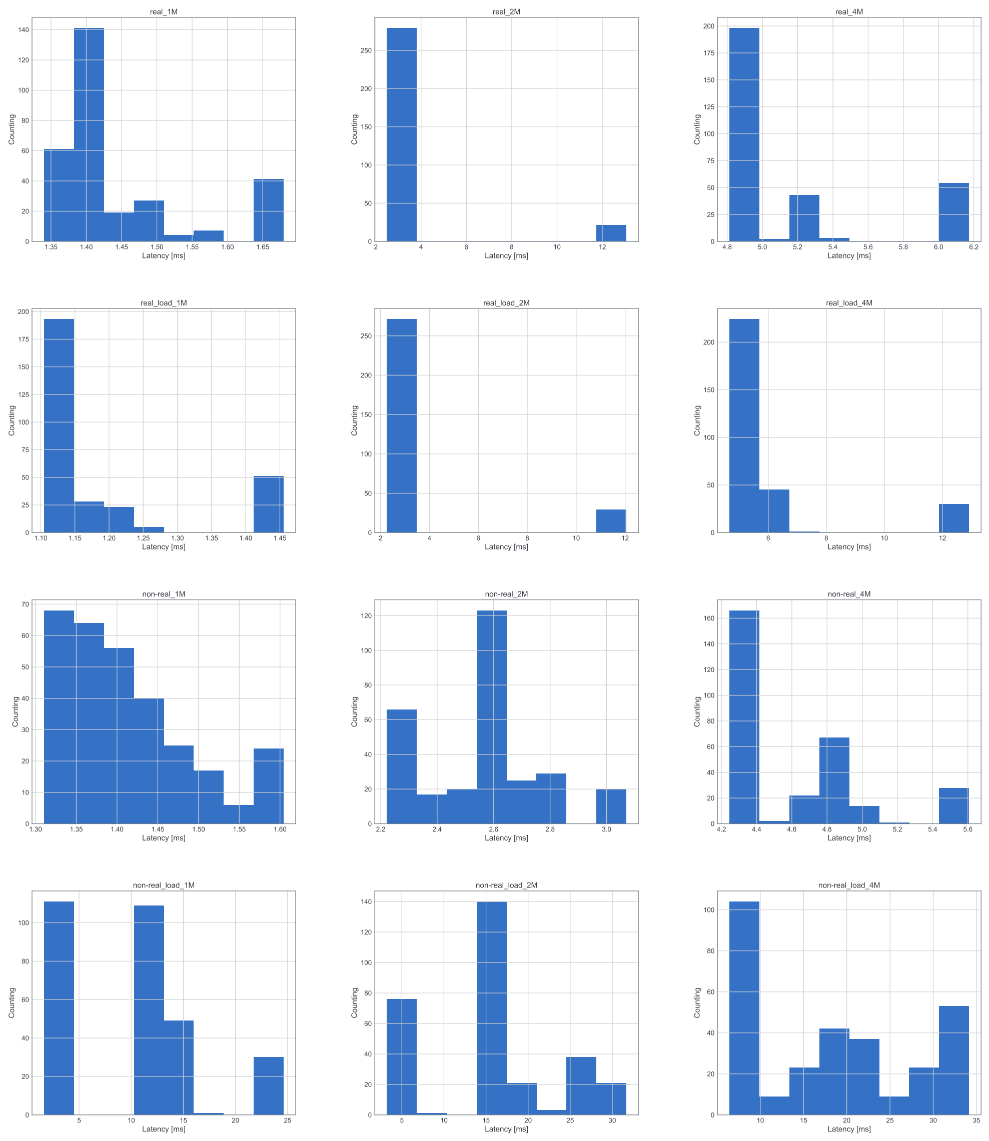

Figure 3: (Non)Real-Time (un)loaded with Various Data Sizes Hist-plot

After a comparison between these four categories, we conclude:

- In terms of the real-time transmission, no matter whether it has extra CPU load or not, latency is mostly concentrated in the lower range of the interval. In addition, the varying data size has no obvious effect on the distribution of the measurements.
- Turning to the non-real-time one, the distribution of the measurements significantly varies to data size. The distribution becomes more sparse with the increase of the data size. If adding extra load to it, the distribution tends to shift to the right side of the interval.
- Comparing real-time to non-real-time communication, it is more stable with different data size with/without the extra load, since it owns a higher priority setting.

4 Question4

4.1 Histogram Analysis

Figure 4 shows histograms of latency in two data sizes of 256KB and 128KB under three protocols - FastRTPS, OpenSlice and Connext.

In view of protocols, when the data size is 256K bytes, distributions of latency histogram under the three protocols are similar, in which most measurements locate around 0.25 ms and partial measurements locate around 0.18 ms.

When the data size is 128 K bytes, the latency of these three protocols distributes between 0.3-0.5 ms. Furthermore, their distributions under the three protocols vary from each other. Latency distribution of OpenSlice is the most average, mainly concentrating in the middle of the interval. Comparatively, distributions of FastRTPS and Connext both mainly distribute in the smaller half of the interval.

From the perspective of data size, under the three protocols, an increase of data size significantly influences the length of latency, which almost double range of latency distribution. Meanwhile, increasing the data size from 128K to 256K bytes will shift the delay distribution from a smaller interval to a larger half interval.

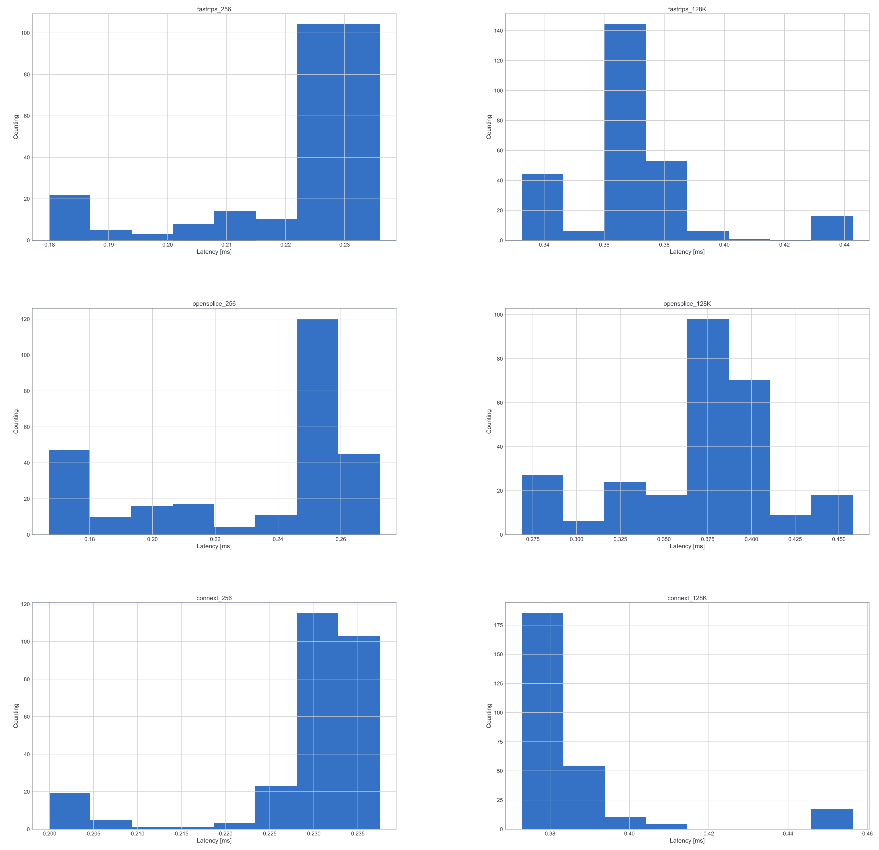

Figure 4: Various Policies with Various Data Sizes

4.2 ANOVA Analysis

Table 2: ANOVA F&P Values

	sum_sq	df	F	PR(>F)
C(DDS)	0.034492	2.0	15.530222	2.086354e-07
C(size)	8.794003	1.0	7919.143846	0.000000e+00
C(DDS):C(size)	0.024713	2.0	11.127291	1.586691e-05
Residual	1.792305	1614.0	NaN	NaN

According to table 2, the p-value is tiny, which means it has less than 0.01% percentage probability that the null hypothesis holds. In other words, it has over 99.99% confidence level to say "the variance between groups is caused by the factors, rather than measuring error".

Table 3: ANOVA Confidence Interval

		N	Mean	SD	SE	95 % Conf.	Interval
DDS	size						
connext	128KB	270	0.3853	0.0221	0.0013	0.3826	0.3879
	256B	270	0.2285	0.0158	0.0010	0.2266	0.2304
fastrtps	128KB	270	0.3698	0.0256	0.0016	0.3667	0.3728
	256B	270	0.2222	0.0201	0.0012	0.2198	0.2246
opensplice	128KB	270	0.3728	0.0499	0.0030	0.3668	0.3788
	256B	270	0.2352	0.0487	0.0030	0.2293	0.2410

Confidence Level can be regarded as "the probability of the variance/errors locating in a specific range", and this "specific range" is the confidence interval. According to table 3, it shows the 95% confidence intervals for the parameters, which do not contain the original point. Therefore, the parameters are nonzero and statistically significant.

Suggestion To The Users: If the user wants to transmit 256B size of the message, it is hard to recommend her/him to use opensplice which has the largest mean of the delay. For the rest of the two, connext has a smaller standard deviation compared with fastrtps. For a better model, it is expected within a comparably narrow range of fluctuations. Hence, we suggest her/him use connext. In terms of transmitting 128KB size of the message, the mean values of these three types of policy are close, but since connext and fastrtps both have a smaller standard deviation, we suggest the users use these two policies.

References

1. DeGlopper, D. (1992). The Art Of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling. By Raj Jain. New York: John Wiley & Sons, 1991.
2. Maruyama, Y., Kato, S., & Azumi, T. (2016, October). Exploring the performance of ROS2. In Proceedings of the 13th International Conference on Embedded Software (pp. 1-10).