IN4390 QEES GROUP 33 OPTIONAL ASSIGNMENT1

Yuhang Tian
5219728

Mingyu Gao
5216281

December 14, 2020

1 Question 5

In this question, we experimented and analyzed how CPU scaling influences the performances of data transportation. We have divided the analysis into three sections. The first section is to analyze the results of different combinations of non-real/real-time, large/small data size and on/off CPU frequency scaling through the box plots. The second section is to detail how frequency scaling impacts real-time transportation with a heavy load. In the third section, it will use ANOVA to evaluate the effect of CPU frequency scaling on the transmitting latency.

1.1 Overall Analysis

Figure 1 shows the comparison of non-real-time transmissions under different scenarios, whereas figure 2 shows the comparison of real-time ones. The varying scenarios are based on the combinations of four factors, non-real-time or real-time, small data size or large data size, CPU loaded or unloaded, and CPU scaling enabled or CPU scaling disabled.

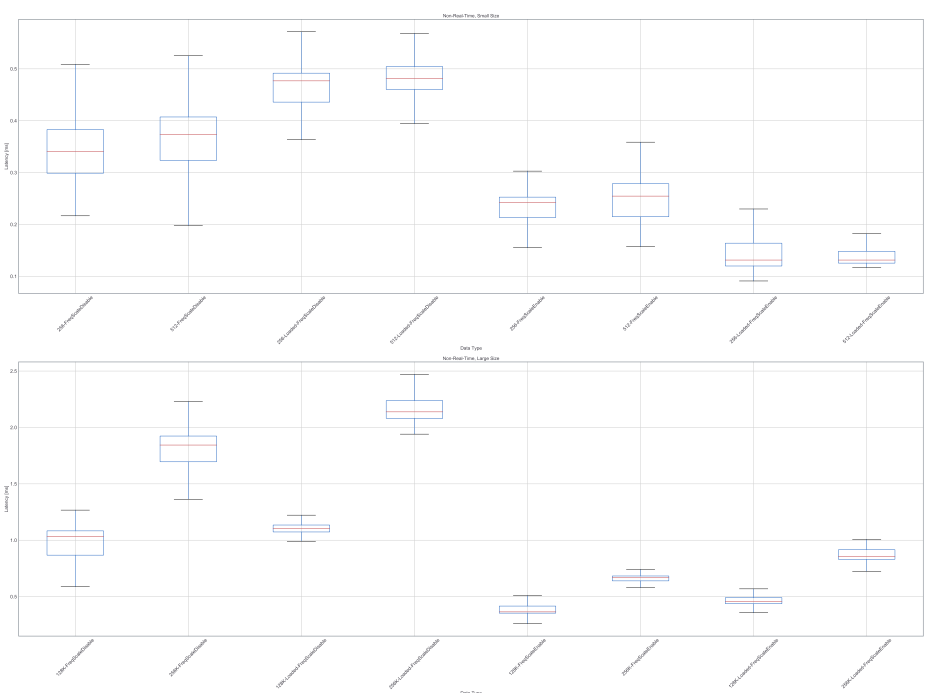


Figure 1: Non-Real-Time Boxplot

According to figure 1, for both small and large size of data with the non-real-time priority, CPU frequency scaling significantly reduces the median of the latency, the distribution of which also becomes compact.

According to figure 2, for both small and large size of data with the real-time priority, CPU frequency scaling has a non-significant impact on the median of the latency, but the distribution of the latency becomes more disperse.

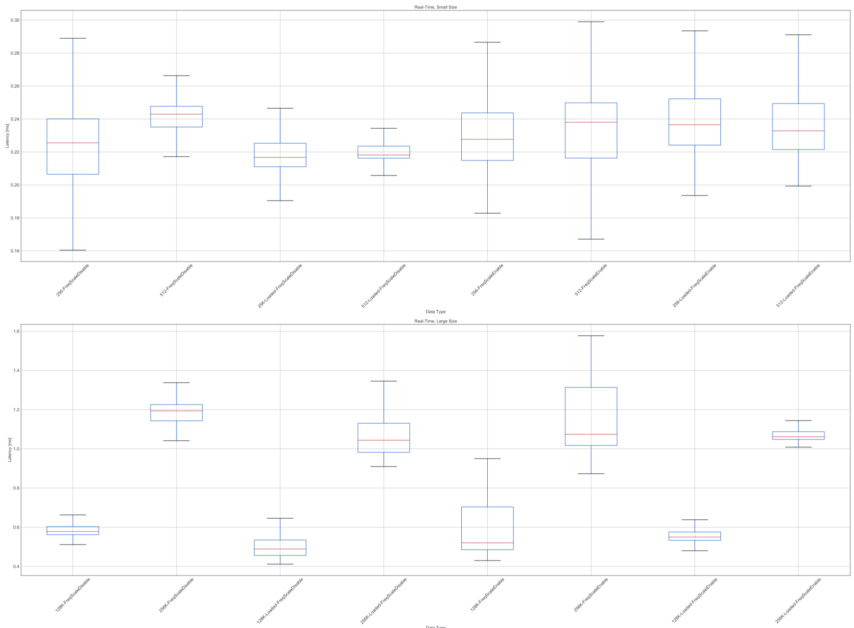


Figure 2: Real-Time Boxplot

In summary, the result reflects that CPU frequency scaling only affects the real-time transmission significantly, while its impact on the non-real-time transmission is not notable. Therefore, in the next section, it will detail the analysis of the influence of the CPU scaling on the real-time data transportation with heavy CPU load.

1.2 Loaded Real-Time Analysis

The box-plot of real-time transmission under heavy load has been shown in figure 3, while the corresponding histogram has been shown in figure 4.

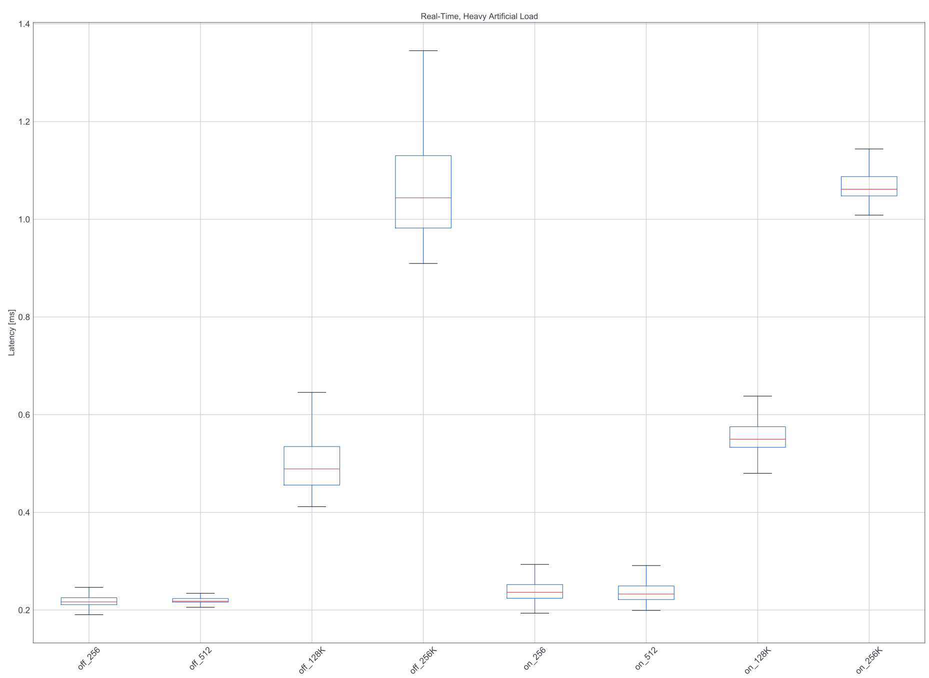


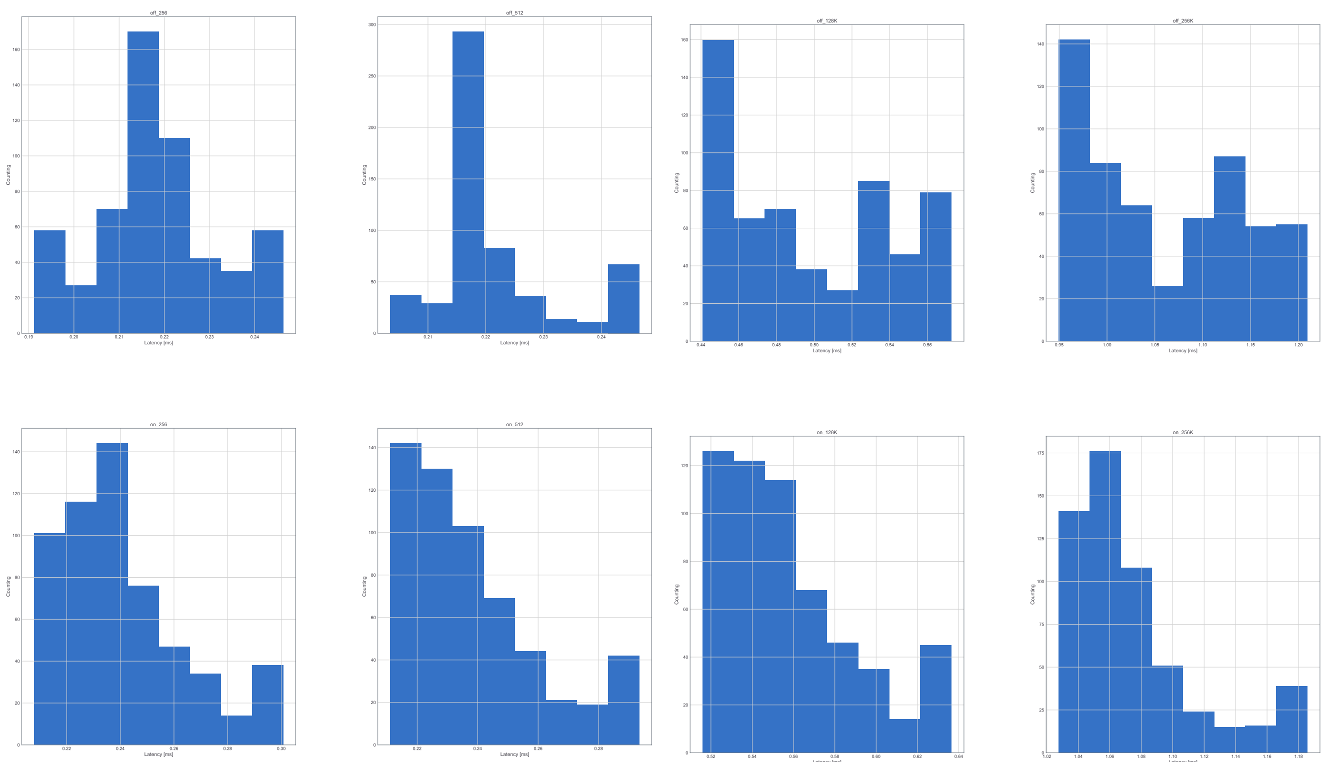
Figure 3: Boxplot: Real-Time with Heavy Load

According to both figures,

- The median latency of small and large data sizes do not vary from each other significantly, no matter whether under heavy artificial load or not, which shows that the heavy intelligent load does not notably affect the median latency of transmission.

- For small data sizes, latency distribution of the transmission with heavy artificial load is slightly more disperse than the one without heavy artificial load. However, for large data size, heavy artificial load makes the distribution of latency more compact.

CPU frequency scaling helps a computer conserve energy by slowing down its processors when the load is low, and speeding the processors up when the system load increases, which is a kind of dynamic adjustments. When a large size of data is to be transferred, detecting the increasing system's load, CPU will automatically increase its processing frequency. As a result shown in figure 3, enabling CPU frequency scaling leads to a more concentrated latency around median value, since the dynamic adjustment can make the performance more stable. However, in terms of transmitting relatively small data sizes, CPU frequency scaling does not significantly respond to the increases due to the low load; as a result, the performance of the transmission is almost identical as the one with CPU frequency disabled.



(a) Small Data Size

(b) Large Data Size

Figure 4: Histogram: Real-Time with Heavy Load

We can see more details from Figure 4 which shows the change of latency distribution when we transmit small and large data in two scenarios - turning on or off the heavy artificial load.

- When ROS2 transmits small data, the heavy artificial load could reduce the peak value in the histogram, especially for 512 bytes. Meanwhile, the distribution of latency becomes more disperse in the histogram. It is the fact that the artificial load occupies so many local-network resources that latency cannot maintain at a low-value cluster.

- However, the latency of large data transmitted by ROS2 tends to gather around the lower part of the interval when the heavy artificial load is added.

1.3 ANOVA Analysis

Table 1: ANOVA F&P Values

	sum_sq	df	F	PR(>F)
C(FreqScaling)	1.259108	1.0	4.871857	0.027311
C(Time)	0.017030	1.0	0.065892	0.797417
C(Loaded)	0.218165	2.0	0.422073	0.655692
C(Size)	3307.584597	3.0	4266.005559	0.000000
C(FreqScaling):C(Time)	0.280189	1.0	1.084134	0.297788
C(FreqScaling):C(Loaded)	0.479057	2.0	0.926807	0.395834
C(FreqScaling):C(Size)	3.225328	3.0	4.159914	0.005919
Residual	4710.420247	18226.0	NaN	NaN

According to ANOVA analysis in Table 1, it has over 97% confidence level to guarantee that the influence of CPU frequency scaling on the variations of the measurements is statistically significant. It deduces that CPU frequency scaling, as a factor, significantly impacts the variance of the measurements. The null hypothesis can be rejected, hence, the variations between each group are caused by the factors, e.g. CPU frequency scaling, not by the experiment or measuring errors.

1.4 Conclusion

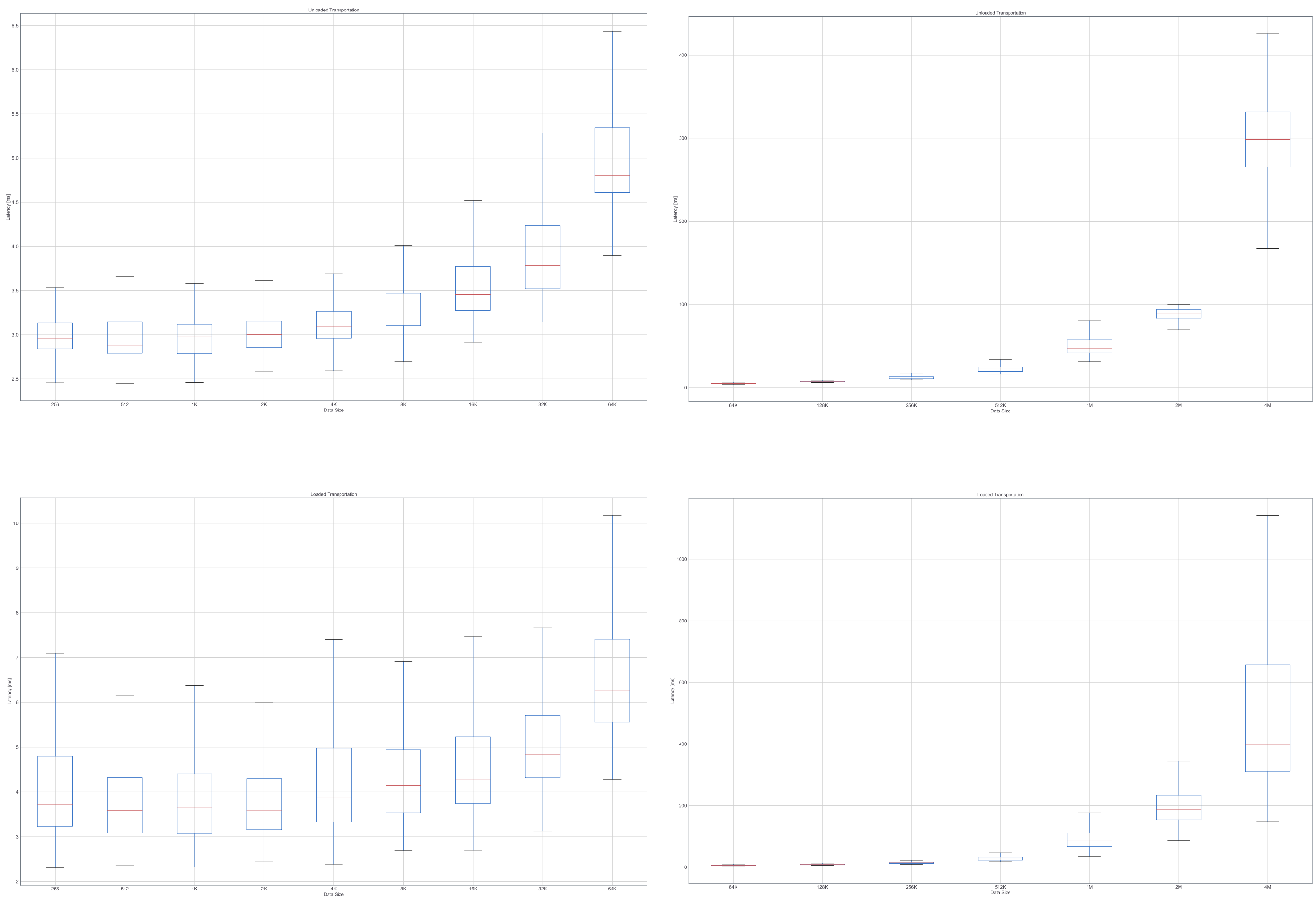
For the fifth question, we drew the following conclusions based on the experimental results:

- CPU Frequency scaling has a significant effect on reducing the median and variance of latency for non-real-time transmission.
- Turning to the real-time transmission, CPU frequency scaling remarkably affects the median latency. However, the variance of latency becomes larger, except the message with large data size and extra load.
- For the real-time transmission under heavy artificial load, turning on frequency scaling will non-significantly impact the median of the latency, but the variance is significantly reduced. For large data size, latency is mainly distributed nearby the minimum of the interval. For small-size data, whether turning on CPU frequency scaling or not, the numerical peak is near to the minimum of the interval, but enabling the CPU frequency scaling makes the distribution of latency more dispersed.
- According to ANOVA, the influence of CPU frequency scaling on latency is statistically significant.

2 Question 6

In our experiment, we carry out the remote transmission using reliable OpenSplice under two scenarios, namely, transmitting the message on a no-extra load network and extra-load network. Our experiment collects results and draws box chart and line chart to reflect the trend of latency when data size gradually increases. Figure 5 shows how the latency changes when we set different sizes of message data and network load. Figure 6 compares the mean value of latency in two different scenarios.

2.1 Box-chart Analysis



(a) Small Data Size

(b) Large Data Size

Figure 5: Opensplice Remote transmission

According to figure 5,

- With the increasing size of data, the median of the latency gradually ascends.
- The increment of the median is more considerable when the data size is larger than 16KB.

The reason for this phenomenon is that ROS2 transmits the message through Ethernet by dividing the message into 15KB packets.

When ROS2 transmits large-size data, the median and the variance of latency are proportional to the data size. In addition, after adding extra load on the network, the median of delay increases by approximately 100 ms for the large data size, however, the increment is not significant for the small size of data, approximately 1ms. Therefore, the influence of extra load on the transmission delay is more notable in large-data transmission, as transmitting small data does not occupy too much resource of the network, while transmitting large data requires more bandwidth. Furthermore, the distribution of the latency is more disperse when network is added with extra loads since these extra loads will cause the network environment unstable.

For both scenarios, the median and the variance of transmitting latency increase sharply when the data size exceeds 1M bytes, and the extra load on the network provided by the third machine has a greater impact on the median of latency than the distribution. These can be explained by the limited payload of OpenSplice, which is the maximum packet size of the IP protocol. Due to this limitation, DDS is not suitable for dealing with large-size data.

2.2 ANOVA Analysis

Table 2: ANOVA Confidence Interval

Network	Data_Size	N	Mean	SD	SE	80% Conf.	Interval
loaded	128K	600	9.1897	2.7389	0.1118	9.0463	9.3332
	16K	600	5.2305	10.9500	0.4470	4.6570	5.8040
	1K	591	4.1712	2.9793	0.1226	4.0140	4.3285
	1M	600	93.6546	36.9179	1.5072	91.7210	95.5883
	256	596	4.4006	2.5139	0.1030	4.2685	4.5327
	256K	600	15.1018	4.3621	0.1781	14.8733	15.3302
	2K	600	4.5070	7.4645	0.3047	4.1160	4.8979
	2M	600	267.7082	1121.7723	45.7962	208.9532	326.4631
	32K	600	5.6498	5.0542	0.2063	5.3851	5.9145
	4K	600	5.1201	8.1476	0.3326	4.6933	5.5468
	4M	600	3120.5878	7416.3491	302.7712	2732.1426	3509.0331
	512	596	4.2827	3.4413	0.1410	4.1019	4.4636
	512K	600	30.3029	12.9964	0.5306	29.6222	30.9836
	64K	600	11.7683	99.5013	4.0621	6.5568	16.9799
	8K	600	5.1771	7.2918	0.2977	4.7952	5.5590
unloaded	128K	599	7.1849	0.8187	0.0335	7.1420	7.2279
	16K	599	3.5549	0.3949	0.0161	3.5342	3.5756
	1K	593	3.0495	0.8556	0.0351	3.0045	3.0946
	1M	600	50.2090	11.2737	0.4602	49.6185	50.7995
	256	599	3.0147	0.4804	0.0196	2.9895	3.0399
	256K	584	11.9817	3.1904	0.1320	11.8124	12.1511
	2K	581	3.0894	0.7040	0.0292	3.0519	3.1269
	2M	600	141.4925	30.6568	1.2516	139.8868	143.0982
	32K	597	3.9693	0.8286	0.0339	3.9258	4.0128
	4K	597	3.1270	0.2838	0.0116	3.1121	3.1419
	4M	600	298.2049	57.3134	2.3398	295.2030	301.2068
	512	593	2.9884	0.3593	0.0148	2.9694	3.0073
	512K	598	23.0267	5.0232	0.2054	22.7632	23.2902
	64K	557	5.0296	0.6784	0.0287	4.9927	5.0665
	8K	592	3.3513	0.4828	0.0198	3.3259	3.3768

Table 2 shows the mean, standard variance and 80% confidence interval of different groups of the measurements. According to the table, it can deduce that:

- For both loaded network and unloaded network, with the increasing data size, the tendencies of mean, SD and confidence interval are to be larger.
- Compared with the loaded network, the confidence intervals of unloaded one are more narrow, which represents that the transmissions are more stable for the unloaded network.
- For the loaded network, when transmitting data which size is larger than 512K, the SDs and confidence intervals are significantly larger than the smaller size of data.

Table 3: ANOVA F&P Values

	sum_sq	df	F	PR(>F)
C(Network)	1.841260e+08	1.0	97.454607	6.308912e-23
C(Data_Size)	3.220041e+09	14.0	121.736469	0.000000e+00
C(Network):C(Data_Size)	2.211012e+09	14.0	83.589237	2.857745e-233
Residual	3.370981e+10	17842.0	NaN	NaN

The F-value of the situation (loaded/unloaded) of the network is large, while its P-value is small. It can deduce that the impact of the factor of unloaded/loaded transmission network is statistically significant. The variances of transmissions are affected by this factor.

2.3 Mean Curve with 80% Confidence Interval

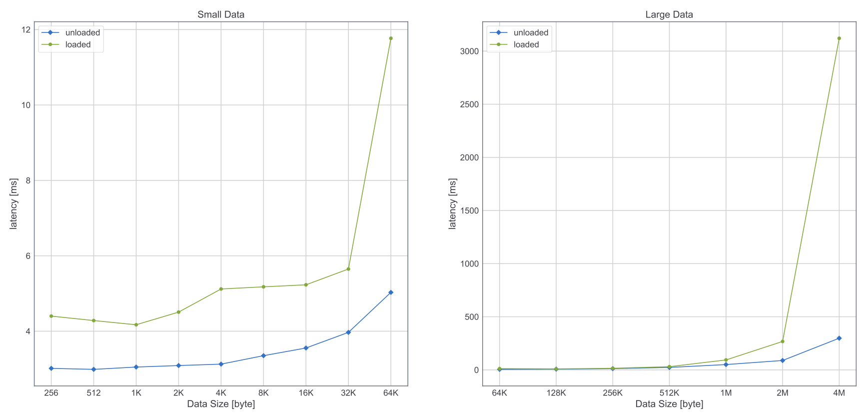


Figure 6: Average Latency for Large and Small Data on no-load and high-load network

In Figure 6, we drew the tendency of mean latency when we changed data size and network load during experiments, and drew the confidence interval for each data size under no-load and high-load network separately in Figure 7.

- Besides transmitting small data on the high-load network, the mean of latency in the other three scenarios keeps increasing along with the augmenting of data size.
- The fluctuation of the average latency for the small data size under high-load network is inconspicuous.
- In terms of large-size transmission, the difference of latency is obvious when the size goes beyond 512K bytes, as large data requires more bandwidth to transfer the message, which can be easily influenced by other simultaneously running tasks under the same network.

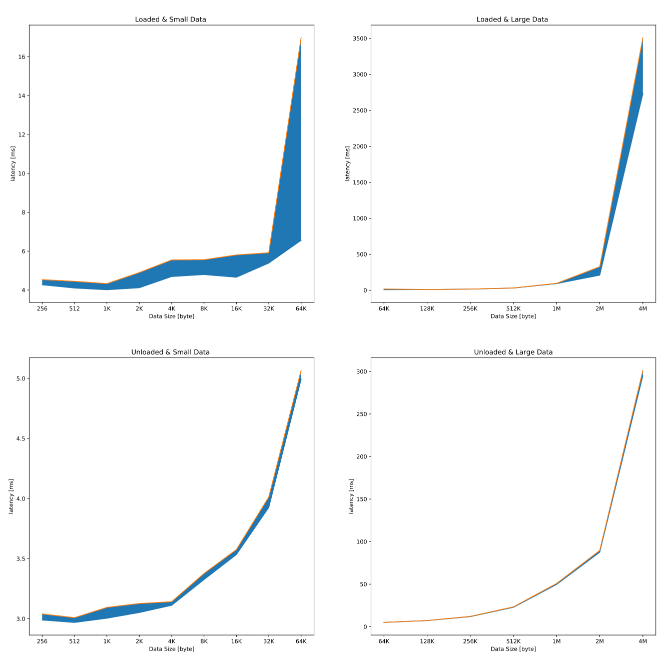


Figure 7: 80% Confidence Interval for Large and Small Data on no-load and high-load network

When it comes to analysis for the 80% confidence interval for small and large data on the no-load and high-load network, we can see the change of 80% confidence interval in Figure 7, which analysis has been interpreted in chapter 2.2.

3 Question 7

This section is mainly for exploring the influence of the sequential order of the listeners on the transmitting latency with FastRTPS policy.

3.1 Box-chart Analysis

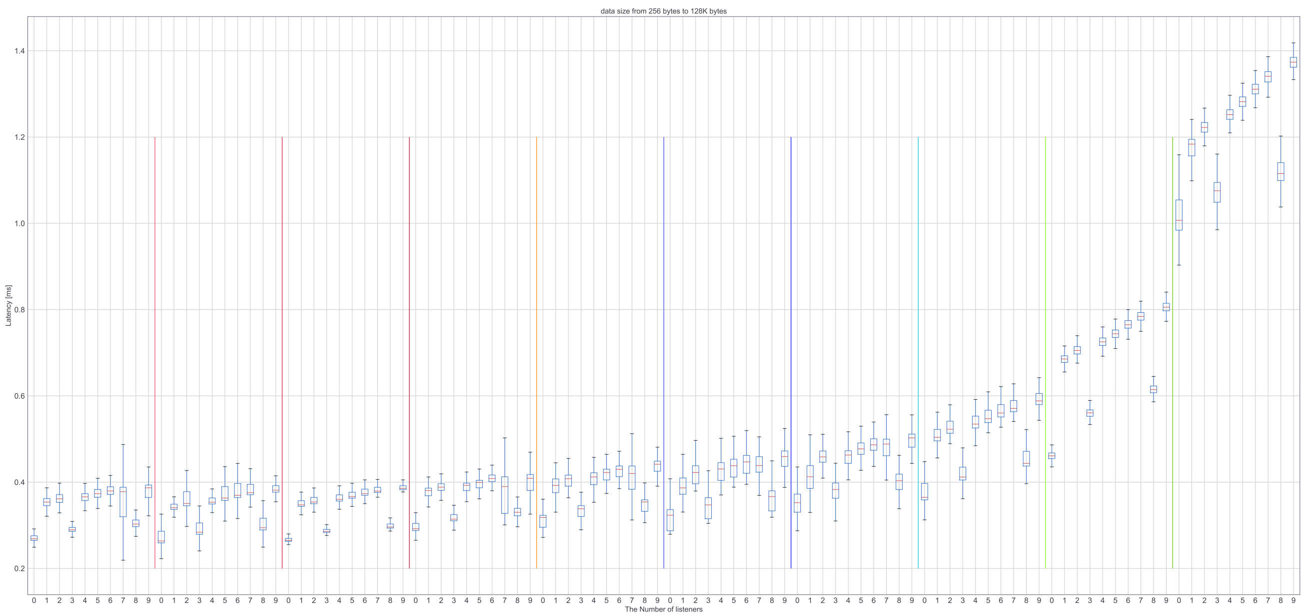


Figure 8: Multi-listeners - data size from 256 bytes to 128K bytes

Figure 8 shows the 10 subscribers listening to a single publisher at the same time. The coloured vertical lines are used for distinguishing different kinds of data size, from 256 bytes to 128K bytes. In general, the listeners at the 1st, 4th, and 9th positions always own a relatively lower latency than other listeners. The variances between listeners at the same positions are not remarkably different when the size of data is less than 64K bytes. The ranking phenomenon becomes more notable after transmitting data size is larger than 64K bytes.

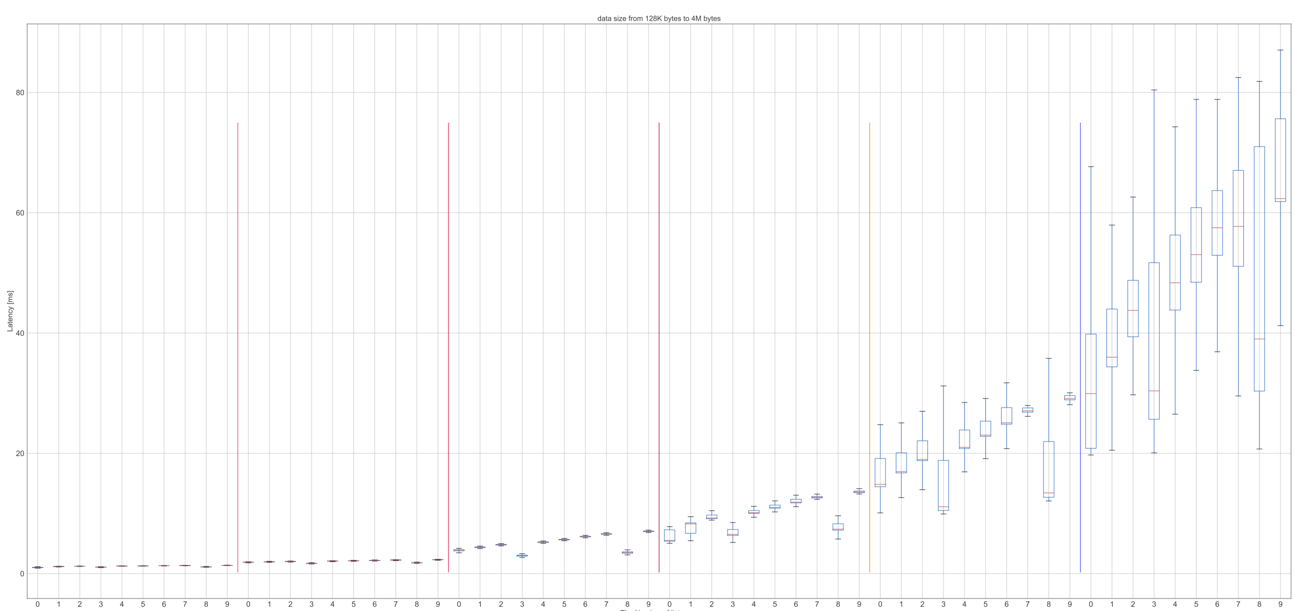


Figure 9: Multi-listeners - data size from 128K bytes to 4M bytes

Figure 9 shows when transmitting large size of data, the latency becomes more unpredictable, especially for 4M byte. In addition, the listeners at 1st, 4th and 9th positions still have a relatively low median, but their distribution becomes more disperse.

According to both figures, it can deduce that:

- If there are ten listeners, the listeners at position 1, 4, and 9 have lower average latency and their latency mainly locate at the first quartile.
- ROS2 message publication is fairer to multiple subscriber-nodes under small data transmission with FastRTPS policy.
- The listeners at the end of the line will have a higher latency, and this phenomenon is more remarkable when the data size is large.

3.2 ANOVA Analysis

Table 4: ANOVA F&P Values

	sum_sq	df	F	PR(>F)
C(Num_Listeners)	1.398071e+05	9.0	1327.770963	0.0
C(Size)	1.373698e+07	14.0	83868.631616	0.0
C(Num_Listeners):C(Size)	6.275970e+05	126.0	425.742198	0.0
Residual	9.985439e+05	85350.0	NaN	NaN

The result of 2-way ANOVA is shown in table 4. The p-value of the number of listeners approximates to zero, which means that it has more than 99.99% confidence level to guarantee that the position of the listeners significantly impacts the variations between different groups.

3.3 Conclusion

Our experimental results for question 7 could be concluded as below:

- FastRTPS policy in ROS2 cannot guarantee the fairly transmitting latency for the subscribers at different sequential positions. If users want to use a fair message publisher, the policy of FastRTPS cannot be adopted.
- With the increase of data size, the distribution of the latency becomes scattered. The truth transmitting latency becomes more unpredictable and unstable.
- Some positions have a much higher priority to subscribe to the message, and this priority is not relative to the ranking position.

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).