

## The Evaluation QS-WFQ Scheduling Algorithm For IoT Transmission To Cloud

Hirzen Hasfani<sup>\*1</sup>, Mardhani Riasetiawan<sup>2</sup>

<sup>1</sup>Master Program of Computer Science and Electronics, FMIPA, UGM, Yogyakarta Indonesia

<sup>2</sup>Department of Computer Science and Electronics, FMIPA, UGM, Yogyakarta Indonesia

e-mail: <sup>\*1</sup>[hirzen.hasfani@mail.ugm.ac.id](mailto:hirzen.hasfani@mail.ugm.ac.id), <sup>2</sup>[mardhani@ugm.ac.id](mailto:mardhani@ugm.ac.id)

### Abstrak

Dalam penelitian ini digunakan algoritme penjadwalan *Weighted Fair Queueing* di mana bobotnya dapat berubah dan dihitung berdasarkan perubahan rata-rata ukuran antrian dalam buffer. Algoritme ini membagi prioritas dari setiap sensor menjadi tiga prioritas yaitu prioritas tinggi, sedang, dan rendah. Masing-masing antrian diberikan bobot yang disesuaikan dengan kebutuhan resource masing-masing trafik. Data dengan prioritas tinggi akan didahulukan tetapi data dengan prioritas sedang dan rendah akan tetap terlayani dan mendapat jaminan resource dari jaringan.

Hasil penelitian ini menunjukkan *packet loss ratio* ketika rasio jumlah buffer dan jumlah data 1:3 dengan variasi jumlah buffer prioritas tinggi, sedang, dan rendah 75:75:150 dan 50:50:200 sebesar 0 %. Waktu delay pengiriman data pada buffer prioritas tinggi dan prioritas sedang memiliki waktu delay yang hampir sama, sedangkan pada prioritas rendah terjadi peningkatan waktu delay.

**Kata kunci**— *Internet of Things, Weighted Fair Queueing, Buffer, QS-WFQ*

### Abstract

This study using the *Weighted Fair Queue* scheduling algorithm when the weights can change and calculated based on changes in the average queue size in the buffer. This algorithm divides the priorities of each sensor into three priorities, namely high, medium and low priority. Each queue is given a weight that is adjusted to the resource requirements of each traffic. High priority data will take precedence, but medium and low priority data will remain underserved and guaranteed by network resources.

The results of this study show *packet loss ratio* when the ratio of the number of buffers and the amount of data is 1: 3 with variations in the number of high, medium and low priority buffers 75: 75: 150 and 50: 50: 200 is 0%. The delay time in the high priority and the medium priority buffer has almost the same delay time when data is transmitted, whereas for the low priority buffer increased in the delay time.

**Keywords**—*Internet of Things, Weighted Fair Queueing, Buffer, QS-WFQ*

## 1. INTRODUCTION

The Internet of Things (IoT) has received great attention since its first appearance. The IoT is the physical devices that communicate each other through an internet network [1]. IoT can also be applied to several other research sectors, for example in research [2] that makes smart video surveillance and measurement, research [3] that makes smart homes and shopping centers equipped with certain device security, research [4] about medical applications, research [5] that makes disaster early warning assistants and others. However, devices in IoT are small devices that are widely distributed and have limitations in terms of storage capacity and computing capabilities. This creates concerns in reliability, performance, security, and privacy in the services that will be built [6]. To overcome this, IoT devices can be integrated into other systems that have better storage capacity and computing capabilities. One system used is a Cloud-based computing system.

One of the IoT Cloud applications is when the sensor data is entered into an IoT device, the data will be collected and generate a lot of data in different sizes, then the data will be transmitted to the cloud. After the sensor data is sent to the cloud, each sensor's data can be accessed widely without having to be in that place [7]. Although integration between IoT and Cloud has various advantages, it still leaves a problem, namely in the case of transmitting IoT data to the cloud for a certain period of time, so that a way to transmit data is needed, one of which is by doing a data scheduling mechanism. The several factors that must be considered in data transmission are the data sent is not lost and the data transmission time is small [8]. Therefore we need a data transmission method that can reduce the risk of losing data packets and reduce data transmission time.

In this study, a method of packet queue scheduling in data transmission will be proposed using the Weighted Fair Queue scheduling algorithm whose weights can vary and are calculated based on the average queue size in the buffer using the Dynamic Weight Standardization calculation. In research [9] shows the Weighted Fair Queue algorithm is better than the Weighted Round Robin algorithm where packet loss on WFQ is 13% compared to the WRR algorithm of 25%. Then the delay in WFQ and WRR is 4 ms and 1.9 s, respectively. This research will be carried out in areas prone to disasters and poor internet connections. Sensors will be divided into several priorities, including high priority, medium priority, and low priority. The first arriving package will be classified and assigned to one queue class based on sensor priority. Each queue is given a weight that is adjusted for the resource requirements of each traffic. The weight of each queue will be calculated based on the priority level. So in this scheduling method, sensor traffic with medium and low priority levels will be served and guaranteed by network resources that can reduce lost data packets and data transmission delay time.

## 2. METHODS

### *2.1 System Design*

The focus of this study will be to analyze the method of transmitting sensor data from the IoT gateway to the cloud using the Queue Size based Weighted Fair Queue scheduling algorithm, whose weights are calculated based on the average size of the queue in the buffer using dynamic weight standardization.

#### *2.1.1 System in General*

The focus of this research is on scheduling IoT sensor data transmission from the IoT gateway to the cloud. This research uses Raspberry Pi 3 as an IoT gateway and the cloud

platform used is Digital Ocean Server. Figure 1 shows a general system diagram that will be built:

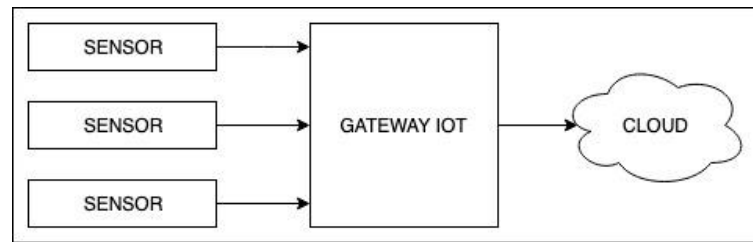


Figure 1 System Diagram in General

Sensors will detect physical changes from the research environment and produce data. Data generated from various sensors will be sent to the IoT gateway. IoT Gateway is an intermediary device between sensors, devices, and cloud that functions to manage varied or heterogeneous IoT device data and store data temporarily. Data from the sensor is then transmitted inside the IoT gateway which will immediately be sent to cloud storage [10].

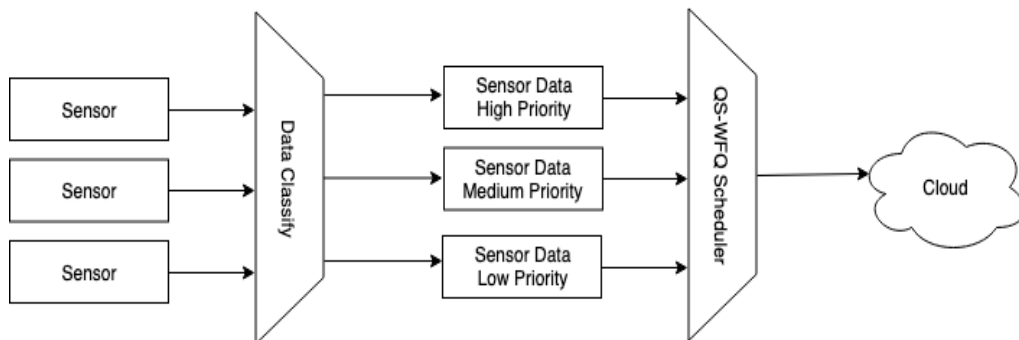


Figure 2 QS-WFQ – Queue Scheduling

The Queue Size based Weighted Fair Queue scheduling mechanism can be seen in Figure 2 starting in the initial process, ie each sensor will send heterogeneous data to the IoT gateway. In IoT gateways the sensor data will be classified based on shipping priorities, namely sensors with high priority, medium priority, and low priority. Sensors with high priority are used sensors GY-521 MPU 6050. Sensors with a priority are used anemometer sensors. While sensors with low priority are used SHT11 sensors. This priority sensor classification will be tailored to the needs in the field. After that the sensor data will enter into each buffer according to the sensor priority. Then the sensor data is scheduled to transmit data based on the Queue Size based Weighted Fair Queue algorithm. Sensor data that has a smaller finishing time will be sent first to the cloud storage so that it forms a data transmission queue based on the calculation of the algorithm.

### 2. 1.2 QS-WFQ Data Transmission Flowchart

This is a data transmission flow diagram using the WFQ algorithm shown in Figure 3.

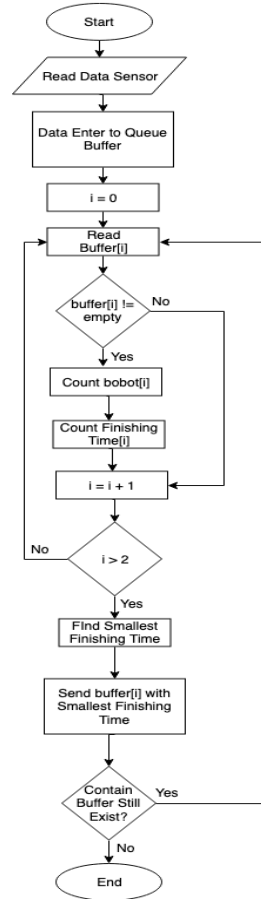


Figure 3 QS-WFQ Data Transmission Flowchart

Figure 3 is the stage of queue scheduling using the Weighted Fair Queue algorithm can be explained as follows:

1. Sensor data will be read in the IoT gateway. Then in the IoT gateway, sensor data will be entered into the buffer of each priority.
2. Then set the initial priority value  $i$  is 0.
3. Then check the contents of the buffer is empty or not. If the contents of the buffer are empty then the priority value  $i$  will be  $i + 1$ . If the contents of the buffer are not empty, weights will be calculated  $[i]$ .
4. If the priority value  $i = 0$ , then weights will be calculated on priority 0. After that, finishing time 0 is calculated, then setting the priority value  $i = i + 1$ . After that, check again whether the value of  $i$  is more than 2. If  $i$  is less than 2, it will return to the process of checking the contents of the buffer and the priority value  $i$  becomes  $i + 1$ .
5. If priority value  $i = 1$ , then weights will be calculated in priority 1. After that finishing time 1 priority calculation is done, then setting the priority value  $i = i + 1$ . After that, check again whether the value of  $i$  is more than 2. If  $i$  is less than 2, it will return to the process of checking the contents of the buffer and the priority value  $i$  becomes  $i + 1$ .
6. If priority value  $i = 2$ , then weights will be calculated in priority 2. After that, finishing time 2 is calculated, then set the priority value  $i = i + 1$ . After that, check again whether the value of  $i$  is more than 2. If  $i$  is less than 2, it will return to the process of checking the contents of the buffer and the priority value  $i$  becomes  $i + 1$ .
7. Then if  $i$  is more than equal to 2 it will be done looking for the smallest finishing time value. Then the data that has the smallest finishing time will be sent first to cloud storage. If the finishing time value of each sensor data in each buffer is the same, then the sensor data that has the highest priority will first be sent to cloud storage.

### 3. RESULT AND DISCUSSION

This research was conducted in 5 experimental scenarios. The first trial scenario uses 300 data with 100 data each priority. The second trial scenario uses 600 data with 200 data each priority. The third trial scenario uses 900 data with 300 data each priority. The fourth trial scenario uses 1200 data with 400 data each priority. And the last fifth trial scenario uses 1500 data with 500 data each priority.

#### 3.1 Packet Loss Ratio – Testing

##### 3.1.1 Testing scenario in 300 Data

The first scenario is sending 300 data, each of which sends 100 data each priority. Table 1 shows the results of testing packet loss ratios when the data sent is 300 data.

Table 1 Data Testing with 300 Data

Algorithm	P1			P2			P3		
	PT	PS	PR	PT	PS	PR	PT	PS	PR
QS-WFQ	0	0	0	0	0	0	0	0	0
QS-WRR	0	0	0	0	0	0	0	0	0

Based on Table 1 on the ratio of data to the amount of data and the number of buffers of 1: 1, there was no loss of data from each priority on the QS-WFQ and QS-WRR algorithms.

##### 3.1.2 Testing scenario in 600 Data

The second scenario is sending data as much as 600 data, each of which sends 200 data each priority. Table 2 shows the results of testing packet loss ratios when the data sent is 600 data. PT shows high priority, PS shows moderate priority, and PR shows low priority.

Table 2 Data Testing with 600 Data

Algorithm	P1			P2			P3		
	PT	PS	PR	PT	PS	PR	PT	PS	PR
QS-WFQ	0	0	0	0	0	0	0	0	0
QS-WRR	0	0	0	0	0	0	0	0	0

Based on Table 2 on the ratio of data to the amount of data and the number of buffers of 1: 2, there is no loss of data from each priority on the QS-WFQ and QS-WRR algorithms.

##### 3.1.3 Testing scenario in 900 Data

The third scenario is sending data of 900 data, each of which sends 300 data each priority. Table 3 shows the results of testing the packet loss ratio when the data sent is 900 data. PT shows high priority, PS shows moderate priority, and PR shows low priority.

Table 3 Data Testing with 900 Data

Algorithm	P1			P2			P3		
	PT	PS	PR	PT	PS	PR	PT	PS	PR
QS-WFQ	0	0	0	0	0	0	0	0	0
QS-WRR	0	0	3,67	0	0	3,33	0	0	3,33

Based on Table 3 on the ratio of data amount and number of buffers to 1: 3, there is no loss of data from each priority on the QS-WFQ algorithm, while QS-WRR data loss occurs at low priority.

##### 3.1.4 Testing scenario in 1200 Data

The fourth scenario is sending data as much as 1200 data, each of which sends 400 data each priority. Table 4 shows the results of testing the packet loss ratio when the data sent is 1200 data. PT shows high priority, PS shows moderate priority, and PR shows low priority.

Table 4 Data Testing with 1200 Data

Algorithm	P1			P2			P3		
	PT	PS	PR	PT	PS	PR	PT	PS	PR
QS-WFQ	0	0	4,5	0	0	4	0	0	5
QS-WRR	0	3	14,75	0	3,25	14,25	0	3	15,25

Based on Table 4 on the ratio of data amount and number of buffers to 1: 4, there is a loss of data at low priority on the QS-WFQ algorithm, while QS-WRR occurs in data loss at medium and low priority.

### 3. 1.5 Testing scenario in 1500 Data

The fifth scenario is sending data in 1500 data, each of which sends 500 data each priority. Table 5 shows the results of testing packet loss ratios when the data sent is 1500 data. PT shows high priority, PS shows moderate priority, and PR shows low priority.

Table 5 Data Testing with 1500 Data

Algorithm	P1			P2			P3		
	PT	PS	PR	PT	PS	PR	PT	PS	PR
QS-WFQ	5,4	5,6	11,8	6,2	6,2	11	5,8	6	11,4
QS-WRR	0	7,2	25,2	0	7,8	24,6	0	8	23,8

Based on Table 5 on the ratio of the data amount of data and the number of buffers of 1: 5, there is a loss of data in each priority on the QS-WFQ algorithm, while QS-WRR occurs in the medium and low priority data loss.

### 3. 2 Delay Time QS-WFQ Algorithm Testing

#### 3. 2.1 Testing scenario in 300 Data

The first scenario is sending 300 data, each of which sends 100 data each priority. Figure 5 shows the results of testing the packet loss ratio when the data sent is 300 data.

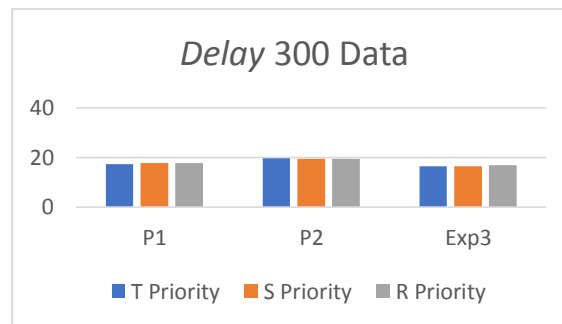


Figure 4 Data Testing with 300 Data

Based on Figure 4, the results of testing 300 data delay times for 100 data for each priority are obtained. In experiment 1, the maximum delay time achieved in T priority was 17.34 s, S priority was 17.65 s and low priority was 17.73 s. In experiment 2, the maximum delay time achieved in priority T is 19.73 s, priority S is 19.45 s and low priority is 19.34 s. In experiment 3, the maximum delay time achieved in T priority is 16.34 s, S priority is 16.32 s and low priority is 16.87 s.

### 3. 2.2 Testing scenario in 600 Data

The second scenario is sending data as much as 600 data, each of which sends 200 data each priority. Figure 6 shows the results of testing packet loss ratios when the data sent is 600 data.

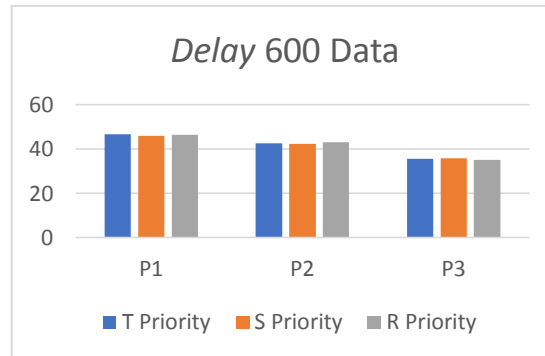


Figure 5 Data Testing with 600 Data

Based on Figure 5, the results of testing 600 data delay for each of the 200 data for each priority are obtained. In experiment 1, the maximum delay time achieved in T priority was 46.67 s, priority S was 45.98 s and low priority was 46.34 s. In experiment 2, the maximum delay time achieved at T priority was 42.56 s, S priority was 42.34 s and low priority was 42.98 s. In experiment 3, the maximum delay time achieved in T priority was 35.6 s, priority S was 35.8 s and low priority was 35.19 s.

### 3. 2.3 Testing scenario in 900 Data

The third scenario is sending data of 900 data, each of which sends 300 data each priority. Figure 7 shows the results of packet loss ratio testing when the data sent is 900 data.

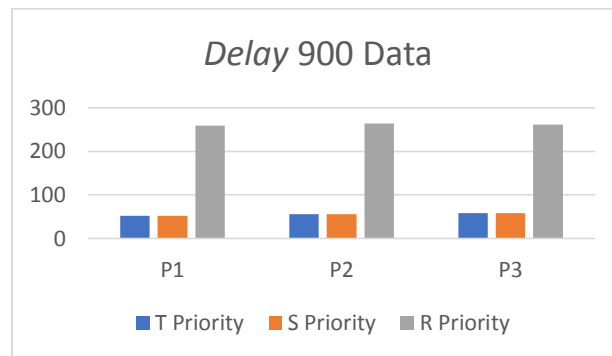


Figure 6 Data Testing with 900 Data

Based on Figure 6, the results of testing 900 data delay for each 300 data for each priority are obtained. In experiment 1, the maximum delay time achieved in T priority was 51.94 s, priority S was 52.08 s and low priority was 258.43 s. In experiment 2, the maximum delay time achieved in T priority was 55.76 s, priority S was 55.23 s and low priority was 264.33 s. In experiment 3, the maximum delay time achieved on T priority is 58.33 s, S priority is 58.54 s and low priority is 261.28 s.

### 3. 2.4 Testing scenario in 1200 Data

The fourth scenario is sending data as much as 1200 data, each of which sends 400 data each priority. Figure 8 shows the results of testing the packet loss ratio when the data sent is 1200 data.

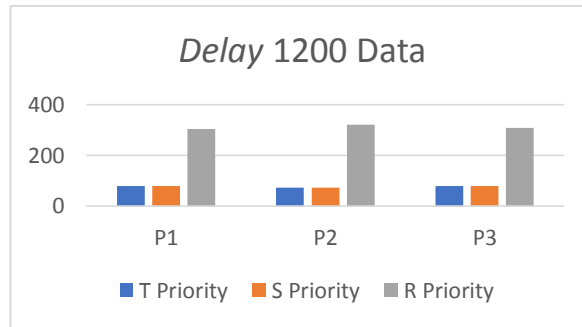


Figure 7 Data Testing with 1200 Data

Based on Figure 7, the test results obtained 1200 data delay times, each of 400 data each priority. In experiment 1, the maximum delay time achieved in T priority was 78.33 s, S priority was 78.48 s and low priority was 303.33 s. In experiment 2, the maximum delay time achieved in T priority is 77.5 s, S priority is 77.52 s and low priority is 321.21 s. In experiment 3, the maximum delay time achieved at T priority is 79.88 s, S priority is 79.54 s and low priority is 308.98 s.

### 3. 2.5 Testing scenario in 1500 Data

The fifth scenario is sending data in 1500 data, each of which sends 500 data each priority. Figure 9 shows the results of testing the packet loss ratio when the data sent is 1500 data.

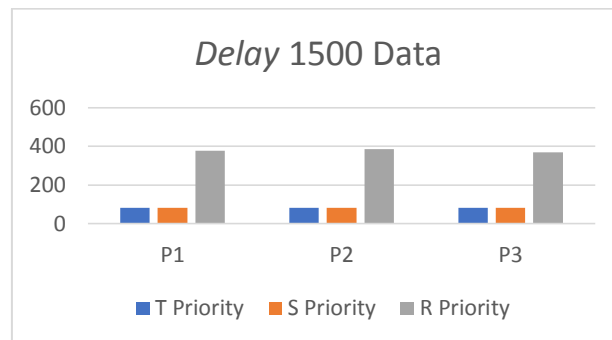


Figure 8 Data Testing with 1500 Data

Based on Figure 8, the test results obtained 1500 data delay time for each 500 data for each priority. In experiment 1, the maximum delay time achieved on T priority was 81.55 s, S priority was 81.34 s and low priority was 377.54 s. In experiment 2, the maximum delay time achieved in priority T was 82.73 s, priority S was 82.55 s and low priority was 386.77 s. In experiment 3, the maximum delay time achieved in T priority was 81.45 s, S priority was 81.52 s and low priority 370.11 s.

### 3. 2 Discussion

The packet loss ratio testing when the number of buffer ratios and the amount of data entered 1: 1 and 1: 2 both on QS-WFQ and QS-WRR algorithm did not experience loss of data packets. It caused the data in each priority that goes into the gateway can still be accommodated in the buffer so that the data can still be sent all to the cloud so there is no data loss. When the ratio of the number of buffers and the amount of data entered 1: 3 with variations of buffers from high priority as much as 75, medium priority as much as 75 and low priority 150 there is no data loss. When data on high priority and medium priority is more sent than low priority, so making low priority is more waiting to be sent, but the place to hold data in the low priority



buffer is still able to accommodate the data so there is no data loss. When the ratio of the number of buffers and the amount of data entered 1: 4 with variations of buffers from high priority as much as 75, medium priority as much as 75 and low priority 150, both high priority data, medium priority and low priority data loss occurs. The packet loss ratio testing uses the QS-WFQ method when the variation of buffers from high priority is 75, medium priority is 75 and low priority 150, both high priority data, medium priority and low priority data loss occurs. This is because the incoming data is more than before and the buffer cannot accommodate the incoming data, making the data packet lost.

Testing the delay time is done in 5 scenarios. In the scenario when the ratio of the number of buffers and data entered 1: 1 and 1: 2, the QS-WFQ and QS-WRR algorithms produce the same delay time for each priority. That is because the queue sequence in each priority alternates in sending data to the cloud and the data entered in the buffer has not exceeded the threshold so there has been no change in its weight. Unlike the scenario in the number of buffers ratio and the amount of data entered 1: 3, 1: 4, 1: 5, in the QS-WFQ algorithm, at the time of initial delivery, the delay time of each priority is almost the same. When traffic starts to solid or the average queue length in the buffer has exceeded a predetermined threshold, the weight of each priority has begun to show changes. But the weight changes in each priority which gradually change does not affect the queue order of each priority. When high priority weights and priority weights are at maximum weight, making low priority data has no weight and data in low priority buffers is not sent. Data delay time in high and medium priority buffers has almost the same delay time. That is because the sequence of data queues in the high priority buffer and is alternating in sending data to the cloud.

#### 4. CONCLUSION

Based on the results of testing the packet loss ratio on the QS-WFQ scheduling algorithm, when the ratio of the number of buffers and the number of data is 1: 1, 1: 2, and 1: 3, data loss still occurs because data entering the buffer can still be accommodated in the buffer except when the ratio of the number of buffers and the amount of data is 1:3 there is a loss of data at low priority when the variation of buffer used is 100: 100: 100. Based on the results of testing the delay time with the QS-WFQ method, when the ratio of the number of buffers and the number of data is 1: 1 and 1: 2, the delay generated from each priority is almost the same. That is because the order of sending data from high priority and alternating.

Based on the results of testing the delay time with the QS-WFQ method, when the ratio of the number of buffers and the number of data is 1: 3, 1: 4, and 1: 5, the delay generated in the high priority and medium priority will decrease and be relatively the same. That is because the order of sending data from high priority and alternating. Whereas in the low priority, the delay generated will increase. That is because low priority data will wait for high priority and is being sent first.

#### REFERENCES

- [1] M. I. Mahali, "Smart Door Locks Based on Internet of Things Concept with mobile Backend as a Service," vol. 1, no. 3, pp. 171–181, 2016.
- [2] I. M. O. Widyantara and N. P. Sastra, "Internet of Things for Intelligent Traffic Monitoring System: A Case Study in Denpasar," *Int. J. Comput. Trends Technol.*, vol. 30, no. 3, pp. 169–173, 2016.
- [3] Z. Shouran, A. Ashari, and T. Kuntoro, "Internet of Things (IoT) of Smart Home: Privacy and Security," *Int. J. Comput. Appl.*, vol. 182, no. 39, pp. 3–8, 2019.
- [4] F. Hussain, "The Application of the Internet of Things in Healthcare," *Int. J. Comput. Appl.*, vol. 180, no. 18, pp. 19–23, 2018.

- [5] S. Taherizadeh and V. Stankovski, "Quality of Service Assurance for Internet of Things Time-Critical Cloud Applications: Experience with the Switch and Entice Projects," in *Proceedings - 2017 6th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2017*, 2017, pp. 289–294.
- [6] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.
- [7] A. Siddique and G. R. Rao, "On-Demand Cloud Based Scheduling for Location Based Services," vol. 3, no. 6, pp. 1–4, 2017.
- [8] O. B. Pratama, A. Bhawiyuga, and K. Amron, "Pengembangan Perangkat Lunak IoT Cloud Platform Berbasis Protokol Komunikasi HTTP," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, 2018.
- [9] X. Li , L. Junhai, L. Wenjing, Y. Huadong, L. Zhenwei, "A queue scheduling approach to QoS support in terminal communication access network," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016*, 2016, pp. 1974–1979.
- [10] M. Tim J, "IoT Gateways: What They Are and How to Use Them," 2016.