Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFICIENT QOS MANAGEMENT AND MONITORING FRAMEWORK FOR CLOUD COMPUTING

[1] S.BEGHIN BOSE, [2] DR.S.S.SUJATHA

[1]Research scholar in computer science, ST Hindu College, Nagercoil, (Manonmaniam Sundaranar University) Abishekapatti, Trinelveli, 627012, Tamilnadu, India.

[2]Associate Professor, Computer Science and Applications, ST Hindu College, Nagercoil, 629 002.

E-mail: [1] s.beghinbose@gmail.com, [2] sujaajai@gmail.com.

ABSTRACT

Cloud services provide various types of pay-per-use computing services to cloud consumers. However, Cloud Service Providers (CSP) are not always able to offer the service agreed in the Service Level Agreement (SLA). The performance of the cloud services depends not only on the CSP but also on the system performance and network bandwidth of the cloud consumers. As a result, sometimes even a good cloud service may not be able to provide a better service to the cloud consumers. Consequently, client-level system monitoring and QoS management is a significant research problem in cloud computing. In this research, propose a QoS management and QoS monitoring framework for cloud computing. In this innovative architecture, the management framework recommends the SLA guaranteed cloud service to customers, and the monitoring framework monitors cloud consumers' system performance and network bandwidth. Through this proposed system, SLA violation and cloud consumer migration can be effectively prevented. Besides, the performance of cloud computing can be improved multiple times. The experimental study shows that this strategy gives a highly adaptable model of customer satisfaction. Additionally, the results demonstrate significant increases in client-side QoS parameters such as latency, response time, and throughput.

Keywords: *QoS, Cloud QoS Management, Fuzzy Logic, Cloud Computing, Service Level Agreement.*

1. INTRODUCTION

Cloud computing is the provision of networking, storage, servers, software, database, and other computing services at the lowest possible cost with the help of the Internet [7][8]. The cloud computing industry has developed tremendously over the last two decades. Cloud computing architecture is divided into three components: the front end, the network, and the back end. Cloud customers access the cloud service from the front end, which includes desktop computers, mobile devices, tabs, web browsers, and thin clients. Cloud service providers manage the back end, which includes interconnected computing servers, storage devices, virtual machines, and security mechanisms. This front end and back end are connected by a network [9][10]. The general workflow architecture of cloud computing is depicted in Figure 1.

Cloud service providers normally offer a wide variety of cloud services to cloud service users on the basis of SLA [11]. This service level agreement includes the most significant QoS metrics of cloud services such as availability, security, privacy, throughput, response time, price, and others. CSP will face several consequences if they fail to provide the agreed-upon service to CSU [12]. The first is to erode cloud users' trust, leading them to seek out alternative services. Second, if the cloud service is not provided properly, the CSP will be liable to CSU for a substantial penalty. As a result, proper management of QoS is a very important mission for CSP. In cloud computing, poor cloud service is usually caused by two factors: CSP using a poor QoS ranking prediction method and CSU using cloud service with very limited computing resources (CPU, Memory, and Bandwidth). Through this research a QoS monitoring and management system has been developed to resolve the cloud service interruptions caused by CSU.

Numerous methods have been developed by cloud researchers and academic researchers to resolve service interruptions caused by CSU. They are only observing and monitoring the configurations (CPU processing speed, memory size, and others) of cloud client systems [2][3]. However, the performance of the cloud client system varies depending on the resource utilization (memory usage and CPU usage) despite the optimal configuration. In this proposed research, three algorithms have been proposed to monitor cloud client's resource utilization: cloud client CPU utilization monitoring, cloud client memory utilization monitoring, and network bandwidth monitoring. Additionally, fuzzy logic has been used in this proposed research to manage cloud QoS effectively. Finally, a user friendly QoS management dashboard has been created to facilitate communication between CSP and CSU. The proposed framework addresses the problems listed above.

Finally, three types of experiments were carried out in real-time cloud architecture to demonstrate the efficacy of the proposed QoS management and monitoring system. First, the experiment determines how efficiently the proposed method handles QoS when network bandwidth varies. Second, the experiment determines how efficiently the suggested solution handles QoS whenever the cloud client's CPU resource utilization changes. Finally, the experiment evaluates the suggested solution's efficiency in managing QoS when the cloud client's memory consumption varies. Further comparisons with newly published cloud QoS monitoring and management approaches are conducted to demonstrate the proposed method's QoS enhancement. Experiments have demonstrated that regulating QoS based on cloud client network bandwidth, memory consumption, and CPU utilization reduces latency, response time, and boosts throughput dramatically.

The paper's primary contributions of this research are summarized below.

1. Automatically collects the status of the cloud client's network usage, memory utilization, and CPU utilization.
2. Without manual intervention, determining the exact cause of Cloud QoS degradation.
3. Automatically managing the QoS according to the status of the proposed three algorithms.

The remainder of this article is structured as follows: Section II discusses recently suggested cloud client-side QoS monitoring and management methods. Section III details the proposed framework and its implementation. Section IV presents the findings of the experiments and a comparative review. Section V concludes this paper.

2. LITERATURE REVIEW

In this section, the recently developed QoS management and monitoring methods are reviewed in detail and its pros and downsides are discussed.

Khalid Alhamazani et al [1]. created a system for tracking and benchmarking the quality of service in cloud computing. The prime advantage of the approach is to track and benchmark individual application components such as databases and web servers that are distributed through multiple cloud layers. At the same time, this method does not suggest any effective communication system for CSP and customers.

Asif Ali Laghari et al [2] have developed a Quality of Experience system for video streaming services. This is used in a cloud computing environment to keep track of video streaming services. This agent-based system collects objective QoE from the cloud and automatically sends it to the client device. The user having the option of sending subjective QoE to cloud management.

Khalid Al Hamadani et al [3] developed the QoS monitoring system for big data analytics. It has two important modules: monitoring manager and monitoring agent. The monitoring manager collects the QoS values of the cloud virtual machine. The monitoring agent is located inside the cloud virtual machine which sends the QoS values of the virtual machine to the monitoring agent. The virtual machine is often monitored in this technique, but the QoS loss caused by the cloud client is not addressed.

Tarik Taleb et al [4] created a framework enabling mobile cloud users to migrate their data centers. This method's primary objective is to make it simple for mobile users to transfer their mobile services to the nearest data center while on the move. As a result, the latency is reduced. Although its objective is admirable, executing it presents numerous challenges, most notably data security and privacy. In addition, implementing this would be more expensive for cloud service providers.

CloudCop is a conceptual network monitoring system described in [5] that is implemented using SNMP. CloudCop is based on the Service-Oriented Enterprise (SOE) model. CloudCop is made up of three components: an agent with web service clients, a backend network monitoring application, and a web service-oriented enterprise. If CloudCop is concerned with network quality of service monitoring, CLAMBS is concerned with the quality of service monitoring.

D. Preetha Evangeline et al [20] created a quality-of-experience paradigm for cloud-based video streaming. For this, an algorithm called Guess Fit has been proposed. The Guess Fit Algorithm is a combination of Naive Bayes and association rule mining. It was created only for cloud media streaming, so adaption problems emerge when new cloud services are introduced.

Chin-Feng Lai et al [21] proposed QoS aware resource allocation strategy for mobile computing. It is based on the graphics processing capabilities of the client machine and the graphics processing units of the cloud virtual machine. It primarily decides the frame rate based on the client machine's data transfer rate. It does not attempt to monitor the client machine's processor and memory use capabilities. The quality of video services can only be increased to a certain extent using this strategy.

XIAOFEI WANG et al [22]. presented two modules to enhance the performance of video streaming for mobile cloud users. These modules comprise social-aware video prefetching and cloud-assisted adaptive video streaming. According to this research, a private agent module has been introduced to enable the mobile cloud center's adaptability to be adjusted. Additionally, the private agent module tracks mobile users' online social media activity. This article discusses how video coding can be used to increase video quality.

1.1 Limitations of the existing works

- Several current approaches track only the cloud client's network. However, the cloud's QoS varies according to the cloud's CPU and memory resource variance.

- Since the majority of current approaches focus exclusively on cloud client or server monitoring. Thus cloud customer migration cannot be effectively prevented. Therefore, an intelligent framework that prevents user migration must be developed [6].

- With current approaches, end user communication is ineffective. Cloud customer can have more trust in the cloud service providers only if communication between the cloud customer and the cloud user is effective.

3. PROPOSED METHODLOGY

The proposed QoS management and monitoring framework is created to automatically manage the quality of the cloud service. It is divided into two sections: QoS management and QoS monitoring. If the issue is in the client system or client network, the cloud user is notified immediately. If the problem is identified inside the cloud environment, the proposed system will look for the exact problem and submit the exact reason to the cloud service provider. For example, when the cloud customer system output is poor due to network traffic or not using a device with the minimum configuration decided upon in the SLA, this monitoring system sends an immediate warning to the cloud customer. This monitoring approach is particularly useful for cloud customers who are unfamiliar with cloud client-side resources. As a result, cloud customers will strengthen their device efficiency and network bandwidth defined in the SLA to achieve the most satisfying QoS in cloud computing.

3.1 Monitoring layer

The monitoring layer monitors the cloud client's network bandwidth, CUP usage, and memory status. The proposed approach uses three algorithms to obtain tracking data for the above-mentioned metrics.

Algorithm 1 monitors and collects the CPU usage of the cloud client system.

Algorithm 2 monitor and collect the bandwidth status of a cloud client system.

Algorithm 3 monitor and collect the memory usage of a cloud client system.

Algorithm 1 is used to measure and monitor the cloud client's CPU usage. In this algorithm, P_Id is the identifier of the current CPU operations, where Cpu_U is the CPU usage of that cloud client system, and Client_ID is the cloud client system's ID, U_status is the CPU Usage status and SharpPcap is the open-source API. It has built-in classes for retrieving information about the cloud client network and system details.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 1 monitors and collects the CPU usage of the cloud client system

Begin
Initialize system variables:
P_Id,Cpu_U,Client_ID,U_status.
***Get** P_Id SharpPcap(The process identifier for a particular Cloud Client System)*
Get Cpu_U SharpPcap (This refers to the CPU Utilization of the Cloud Client System.)
***Get** Client_ID SharpPcap (Cloud client system Id)*
do
***If** (Cpu_U <= 30 %)*
{
U_status=' Normal'
}
Else
***If** (Cpu_U >= 30 % AND Cpu_U <= 60 %)*
{
U_status=' Average'
}
Else
***If** (Cpu_U >= 60 %)*
{
U_status=' Below_Average'
}
***Return** U_status*
End

CPU usage is calculated by the following formula.

$$Cpu_U = \frac{((CT_u+CT_k)-(PT_u+PT_k))}{100T} \times 100 \quad (1)$$

- $CT_u -$ This variable specifies the CPU's current time in user mode.
- $CT_k -$ This variable specifies the CPU's current time in kernel mode.
- $PT_u -$ In user mode, this variable defines the previous CPU time.
- $PT_k -$ In kernel mode, this variable defines the previous CPU time.
- Multiplying by 100 yields the total percentage of CPU usage.

Algorithm 1 generates three types of outputs: normal, average, and below average. If the CPU Utilization is less than 30%, Algorithm 1 returns the status as normal, if it is between 30% and 50%, Algorithm 1 returns the status as average, and if it is greater than 50%, Algorithm 1 returns the status as below average.

Algorithm 2 is used to collect the bandwidth status of a cloud client system. According to Algorithm 2, P_Id represents the cloud client system's process ID, B_width represents the cloud client machine's network usage, MAC_Adr represents the cloud client machine's hardware MAC address, Client_ID represents the cloud client system's system ID, and B status represents the cloud client system's current bandwidth status.

Algorithm 2 monitor and collect the bandwidth status of a cloud client system

Begin
Declare variables: P_Id,B_Usage ,MAC_Adr,Client_ID,U_status.
Get P_Id SharpPcap(The process identifier for a particular Cloud Client System.)
Get B_width SharpPcap (Bandwidth of a specific Cloud Client System)
Get Client_ID SharpPcap (Cloud client system Id)
do
***If** (B_Usage > 10 MB)*
{
B_status=' Normal'
}
Else
***If** (B_width <= 10 MB AND B_Usage >= 6) MB*
{
B_status=' Average'
}
Else
***If** (B_Usage <= 6 MB)*
{
B_status=' Below_Average'
}
***Return** B_status:*

Cloud client network bandwidth is calculated by the formula 3.

$$B_{Usage} = \frac{(Din_c - Din_p) \times 8 \times 100}{C_t - P_t} \quad (2)$$

- Din_c - The variable describes the current bit value of data received rate (incoming data) to the cloud client.
- Din_p - Variable describe the previous value of the data received rate (incoming data) to the cloud client which is in bits.
- To convert byte, the value 8 is used.
- C_t – The variable denotes the current time of

data arrival (incoming data) to the cloud client system.

- P_t – Variable describe the previous time of data arrival (incoming data) to the cloud client system.

The outputs of Algorithm 2 are divided into three categories: normal, average, and below average. Algorithm 2 returns the status as normal if the bandwidth is greater than 10 MB. Algorithm 2 returns the status as average if the bandwidth is less than 10 MB and greater than 6 MB, and below average if the bandwidth is less than 6 MB.

Memory usage of a cloud client device was collected using Algorithm 3. P_Id is the cloud client system's process ID. CMU is the cloud client machine's primary memory usage. The System ID of the cloud client system is represented by Client_ID. M_status is the cloud client system's actual Memory usage status.

Algorithm 3 monitor and collect the memory usage of a cloud client system
Start *Declare variables:* *P_Id,CMU,Client_ID,M_status.* *Get C_Ram SharpPcap(Random Access Memory of a specific Cloud Client System)* *Get P_Id (Process ID of a specific Cloud Client System)* *Get Client_ID SharpPcap (Cloud client system Id)* ***do*** ***If*** *(CMU≤ 20%)* *{* *(M_status=' Normal')* *}* ***Else*** ***If*** *(CMU ≥ 20 % AND CMU≤ 40 %)* *{* *M_status=' Average'* *}* ***Else*** ***If*** *(CMU ≥ 40%)* *{* *M_status=' Below_Average* *'}* *Return M_status:* ***End***

The memory usage of a cloud client can be calculated by the formula 3.

$$CMU = \frac{(CM_u - CM_b - CM_c)}{CM_u + CM_f} \times 100 \quad (3)$$

- CMU-Total client memory usage.
- $\boldsymbol{CM_u}$- This variable describes the memory assigned to a process.
- $\boldsymbol{CM_b}$ - This variable indicates how much client memory takes process buffers.
- $\boldsymbol{CM_c}$- This variable is used to define a usage of cloud cached memory.
- $\boldsymbol{CM_f}$- This variable indicates the amount of available memory.

In terms of client memory status, Algorithm 3 produces three types of results: normal, average, and below average. If the memory consumption is less than 20%, algorithm 3 returns the status as normal. Algorithm 3 returns an average status if the memory usage is greater than 20% and less than 40%, and a below average status if the memory usage is greater than 40%.

3.2 QoS management

Qos management in the proposed method by Fuzzy logic algorithm. It is automatically calculated the penalty details caused by cloud client and cloud service. The fuzzy logic used in the proposed method consists of three main parts: fuzzification, inference mechanism and defuzzification. Mamdani style fuzzy logic is used in this study. The following sections address each of the three blocks:

(1) The fuzzification method generates fuzzy sets based on the following values: CUP utilisation, memory usage, and bandwidth. [13].
(2) Inference mechanism: The method of constructing a mapping for an output using fuzzy logic from a given input. [14].
(3) Defuzzification: It specifies what action the CSP or CSU should take in order to improve the QoS, which is based on the fuzzy set [15].

The proposed QoS management module has three layers. Figure 2 illustrates the QoS Management module's graphical representation. The overall resource utilization of the cloud client system is calculated using the given formula (4).

$$\mu_{RC} = [Cpu_U, \mathrm{B_{Usage}}, \boldsymbol{CMU}] \quad (4)$$

Cpu_U : Indicates the current process utilization of the cloud client CPU calculated by algorithm 1.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

B_{Usage} : Indicates the cloud client's bandwidth status as determined by algorithm 2.
CMU : Indicates the cloud client's memory status as determined by algorithm 3.

The output variable evaluates the customer satisfaction and penalty details according to the behavior of the input variables. The customer satisfaction values of cloud service is explained in table1. The range of the output is from 0 to 10. This variable consists of four fuzzy sets.

1. Customer satisfaction is excellent if the output value is between 0 and 3. It means that both the parties are strictly following the agreement stated in the SLA.
2. Customer satisfaction is best if the output value is between 3 and 6. It means that both the parties are following the agreement stated in the SLA.
3. Customer satisfaction is moderate if the output value is between 7 and 9. This means that cloud customers are not adequately following the agreement stated in the SLA.
4. Customer satisfaction is very poor if the output value 10. This means that cloud service provider are not effectively following with the agreement stated in the SLA. In this situation, Cloud service providers must pay the penalty to the cloud customer.

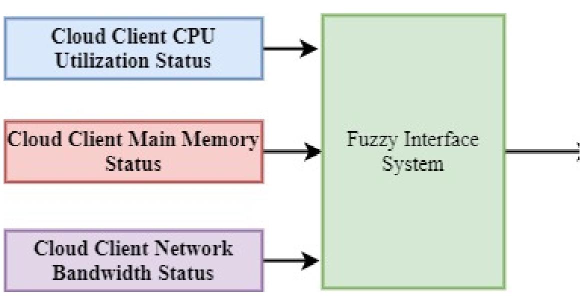

Figure 2. Proposed fuzzy logic QoS management.

3.3Customer Dashboard

This proposed approach is created to enhance the quality of various cloud services, including video conferencing, audio conferencing, and voice to text typing. When device output and network speed fall below the levels specified in the SLA, the cloud customers are automatically notified. When system performance and network speed are lower than stated in the SLA, it automatically alerts the cloud customers. This allows the cloud customer to select the optimal system and optimal network bandwidth specified in the SLA. This immensely helps to make better media services available to cloud customers. Moreover, this method dramatically benefits cloud users and service providers in so many ways. The implementation details of the proposed method is described below.

Dashboard of QoS management and monitoring system

This opens as the first screen when a user logs in. It has six main pages: Network information and monitoring, System information and monitoring, SLA details, reports, feedback, and notifications. Figure 3 depicts the main dashboard of the proposed methodology.

System Monitoring and Information

This screen monitors the cloud client's current processor speed, CPU utilization, total memory, memory usage, GPU speed, and total storage space. It also informs customers how much service redundancy occurs if the system configuration is less than stated in the SLA. The system monitoring and information screen are shown in figure 4. This screen also informs how much throughput, latency, and response time are available according to the current system configuration.

Network monitoring and information

The network monitoring and information screen display the incoming and outgoing data rates of the client system used by the cloud customer. Second, the screen will show the SLA stated incoming and outgoing data rate (network speed required for cloud customers to get the correct QoS). The network information screen of the proposed method is displayed in figure 5. It also informs customers how much service latency occurs if download and upload speed are lower than stated in the SLA.

SLA information

This screen contains the SLA and cloud service level information. The SLA information screen shown in figure 6. This screen contains important information that the cloud customer must follow to get the best QoS. It mainly covers the penalty details that CSP has to pay to the cloud customer in case of QoS loss caused by CSP, the minimum system configuration that cloud customer has to maintain, cloud service availability, cloud security, and privacy.

Customer feedback

Through this page, cloud customer can report customer service feedback to cloud service providers. It creates a direct communication between the cloud customer and the service providers. Customer feedback form is shown in Figure 7.

4. EXPERIMENTAL ANALYSIS

4.1 Software and hardware details

Five desktop computers with varying configurations were used to evaluate the proposed method's effectiveness. The QoS monitoring and management dashboard was developed using ASP.Net 2016 and MS SQL server. Additionally, this model is implemented in Windows 10.

4.2 Evaluation metrics

The proposed method's performance is evaluated using three impartent client-side QoS metrics: response time, latency and throughput.

Response time: The amount of time it takes for the cloud service provider to react to a cloud user [17]. It is expressed as a millisecond.

Throughput: Throughput is a metric that indicates how many units of data can be processed in a given amount of time [19]. It is expressed in megabits per second.

Latency: The time between sending a packet and receiving it at its destination [18]. It's measured in milliseconds.

The proposed method splits the experiment into three sections. The first QoS monitoring procedure was carried out over a network with varying bandwidth. Second, a computer with varying CPU consumption is used to monitor QoS values.

Finally, a computer with varying memory consumption is used to monitor the QoS values.

To ascertain how network bandwidth affects cloud performance, the experiment is conducted at three distinct time intervals: morning, peak hour, and night. Second, the experiment is performed in three ways to find out how CPU utilization affects cloud QoS. For this cloud service is used from three computers. For this cloud service is used from three computers. The first computer is running at a high rate of processor utilization, the second computer is running at a moderate rate of CPU utilization, and the third computer is running at a very low rate of CPU utilization. Finally, three machines are used to conduct an experiment to determine how cloud client memory usage affects the cloud service which includes high memory utilization, low memory utilization, and moderate memory utilization.

Table 3 The Qos Values Measured At Three Time Intervals

Time intervals	Throughput	Response Time	Latency
Peak hours	345	52	66
Morning	320	56	66
Night	400	40	45

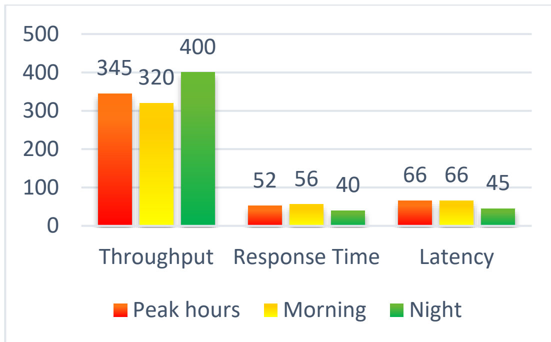

Figure 8. Performance Comparison During Different Time Intervals.

Table 4: Qos Monitoring During Different CUP Utilization

CPU utilization	Throughput	Response Time	Latency
High utilization	320	89	104
Moderate utilization	346	51	48
Low utilization	400	28	26

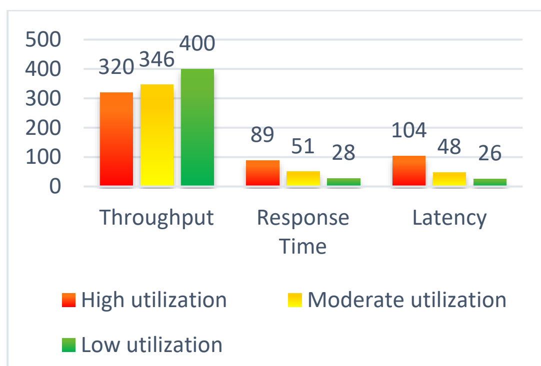

Figure 9. Qos Performance Comparison During Different CUP Utilization

Table 5: Qos Monitoring During Different Memory Utilization

Memory utilization	Throughput	Response Time	Latency
High memory utilization	356	72	90

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Moderate memory utilization	311	59	75
Low memory utilization	310	34	55

According to varying bandwidth, cloud efficiency is poor during periods of heavy network traffic, but the cloud performance is extremely high at night. As a result, cloud computing performance varies according to the network's bandwidth status. The results of these experiments indicate that the efficiency of cloud computing is dependent on the network's bandwidth status. Table 3 summarizes the QoS values obtained at three-time intervals. Figure 8 depicts the QoS variation caused by network bandwidth. Table 4 summarizes the QoS values measured at various CPU utilization levels. When the CPU consumption is high, cloud services consume a lot of response time and latency. The QoS variation caused by high CPU utilization is illustrated in figure 9. Experimental results show that high memory utilization also has a major impact on latency and response time. Table 5 summarizes the QoS values obtained at different stages of memory usage. Figure 10 illustrates the degradation in QoS caused by heavy memory consumption. Cloud service providers will definitely strengthen QoS by implementing this approach.

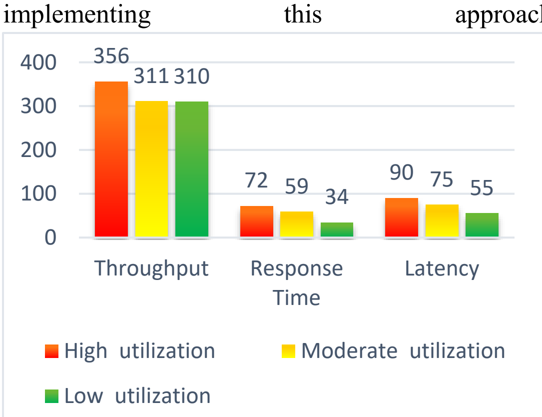

Figure 10. Qos Performance Comparison During Different Memory Utilization.

Comparatively analysis

The proposed method's efficiency is evaluated using the following recently developed QoS management and monitoring methods.

- Mobile Cloud Gaming CMG [17].
- Follow Me Cloud (FMC) [4].
- Cross-Layer Multi-Cloud Application Monitoring as a Service Framework (CLAMS) [3].
- Quality of Experience (QoE) [2].

Table 6: Qos Monitoring During Peak Hours

Number of tasks	Throughput	Response Time	Latency
Proposed Method	397	52	66
QoE	280	61	70
CLAMS	295	66	78
FMC	299	59	79
CMG	301	69	81

To demonstrate the proposed method's efficacy, a comparative study with current methods was conducted. The experimental results indicate a significant increase in throughput, latency, and response time. Table 6 summarizes the outcomes of the experiments. The proposed method's efficiency is compared in Figure 11. In terms of throughput, the proposed approach has a throughput of 397 MBPS. This is 96 MBPS faster than the following approach CMG. The proposed method has a response time of 52 ms, which is lower than the existing methods. Since the response time of the proposed method is very short, QoS will be high. The proposed method has a lowest latency of 66 milliseconds. According to the comparative analysis, it is obvious that by properly managing QoS based on cloud client resource consumption, QoS valuation can be significantly decreased.

This study investigates methods for preventing QoS degradation due to a cloud client's high resource utilisation. Simultaneously, QoS degradation also happens due to excessive resource utilisation of cloud virtual machines (VMs). The cloud customer can achieve 100 percent QoS by properly correcting the QoS degradation caused by the cloud VMs.

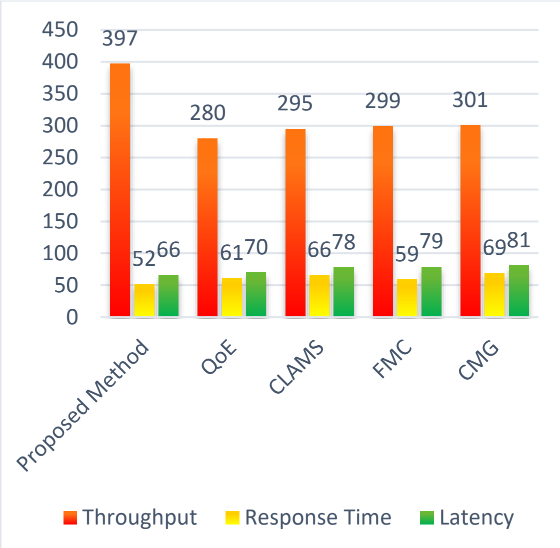

Figure 11. Performance Comparison With State-Of-The-Art Methods.

5 CONCLUSION

Quality-of-Service (QoS) is a vital concept for optimal cloud service selection and customer satisfaction in cloud computing. QoS rankings provide essential information for making the best choice of cloud services from a range of functionally equivalent cloud services. It is clear from this paper that the performance of cloud computing depends not only on a ranking prediction but also on the customer-side network and system performance. In this proposed method, a revolutionary QoS management and monitoring framework is proposed for cloud QoS enhancement. Instead of monitoring the cloud client's system configuration, this proposed approach tracks the CPU resource usage, memory utilization, and network bandwidth utilization of the cloud client system. Three algorithms have been proposed to accomplish this. Second, a QoS management framework based on fuzzy logic has been developed for efficient QoS management, which significantly reduces cloud customer migration. Finally, a user-friendly dashboard has been created to facilitate communication between cloud users and cloud service providers. The experimental results demonstrate that the proposed method can significantly improve the throughput, latency, and response time of the client-side QoS matrices.

In feature study, we're looking to explore strategies for preventing QoS degradation caused by a cloud virtual machine's high resource consumption. This will significantly increase cloud customer satisfaction.

REFERENCES

[1] K. Alhamazani et al., "Cross-Layer Multi-Cloud Real-Time Application QoS Monitoring and Benchmarking As-a-Service Framework," in IEEE Transactions on Cloud Computing, vol. 7, no. 1, pp. 48-61, 1 Jan.-March 2019, doi: 10.1109/TCC.2015.2441715.
[2] A. A. Laghari, H. He, A. Khan, N. Kumar and R. Kharel, "Quality of Experience Framework for Cloud Computing (QoC)," in IEEE Access, vol. 6, pp. 64876-64890, 2018, doi: 10.1109/ACCESS.2018.2865967.
[3] K. Alhamazani et al., "Real-Time QoS Monitoring for Cloud-Based Big Data Analytics Applications in Mobile Environments," 2014 IEEE 15th International Conference on Mobile Data Management, Brisbane, QLD, Australia, 2014, pp. 337-340, doi: 10.1109/MDM.2014.74.
[4] T. Taleb and A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks," in IEEE Network, vol. 27, no. 5, pp. 12-19, September-October 2013, doi: 10.1109/MNET.2013.6616110.
[5] M. K. Nair and V. Gopalakrishna, "'CloudCop': Putting network-admin on cloud nine towards Cloud Computing for Network Monitoring," 2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA), Bangalore, India, 2009, pp. 1-6, doi: 10.1109/IMSAA.2009.5439474.
[6] P. Jamshidi, A. Ahmad and C. Pahl, "Cloud Migration Research: A Systematic Review," in IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 142-157, July-December 2013, doi: 10.1109/TCC.2013.10.
[7] M. Asim, Y. Wang, K. Wang and P. -Q. Huang, "A Review on Computational Intelligence Techniques in Cloud and Edge Computing," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 6, pp. 742-763, Dec. 2020, doi: 10.1109/TETCI.2020.3007905.
[8] Y. Hung, "Investigating How the Cloud Computing Transforms the Development of Industries," in IEEE Access, vol. 7, pp. 181505-181517, 2019, doi: 10.1109/ACCESS.2019.2958973.

[9] C. Wu and S. Marotta, "Framework for Assessing Cloud Trustworthiness," 2013 IEEE Sixth International Conference on Cloud Computing, Santa Clara, CA, USA, 2013, pp. 956-957, doi: 10.1109/CLOUD.2013.76.

[10] C. Yoon, "Cloud client-computing architecture and methods for high-quality media oriented services," 2014 International Conference on Information and Communication Technology Convergence (ICTC), Busan, Korea (South), 2014, pp. 389-390, doi: 10.1109/ICTC.2014.6983163.

[11] Hussain W., Hussain F.K., Hussain O.K. (2014) Maintaining Trust in Cloud Computing through SLA Monitoring. In: Loo C.K., Yap K.S., Wong K.W., Beng Jin A.T., Huang K. (eds) Neural Information Processing. ICONIP 2014. Lecture Notes in Computer Science, vol 8836. Springer, Cham. https://doi.org/10.1007/978-3-319-12643-2_83.

[12] Chowdhury A.G., Das A. (2018) Importance of SLA in Cloud Computing. In: Aggarwal V., Bhatnagar V., Mishra D. (eds) Big Data Analytics. Advances in Intelligent Systems and Computing, vol 654. Springer, Singapore. https://doi.org/10.1007/978-981-10-6620-7_15.

[13] Hayat, B., Kim, K.H. & Kim, KI. A study on fuzzy logic based cloud computing. Cluster Comput 21, 589–603 (2018). https://doi.org/10.1007/s10586-017-0953-x.

[14] Tariq, Muhammad Imran et al. 'An Analysis of the Application of Fuzzy Logic in Cloud Computing'. 1 Jan. 2020 : 5933 – 5947.

[15] AbhishekKesarwani et al, "Development of trust based access control models using fuzzy logic in cloud computing", https://doi.org/10.1016/j.jksuci.2019.11.001.

[16] Shaoxuan Wang and Sujit Dey. 2012. Cloud mobile gaming: modeling and measuring user experience in mobile wireless networks. SIGMOBILE Mob. Comput. Commun. Rev. 16, 1 (January 2012), 10–21. DOI:https://doi.org/10.1145/2331675.2331679.

a. S. H. Vidya and R. M. Prakash, "Response time analysis of dynamic load balancing algorithms in Cloud Computing," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2020, pp. 371-375, doi: 10.1109/WorldS450073.2020.9210305.

[17] M. S. Bali and S. Khurana, "Effect of latency on network and end user domains in Cloud Computing," 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), 2013, pp. 777-782, doi: 10.1109/ICGCE.2013.6823539.

[18] A. P. Shameer and A. C. Subhajini, "Throughput Maximization on Efficient Load Balancing in Cloud Task Scheduling Using Enhanced Bee Colony Algorithm," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, pp. 294-298, doi: 10.1109/CTCEEC.2017.8455119.

[19] Preetha Evangeline D., Palanisamy A. (2018) A Multimedia Cloud Framework to Guarantee Quality of Experience (QoE) in Live Streaming. In: Rajsingh E., Veerasamy J., Alavi A., Peter J. (eds) Advances in Big Data and Cloud Computing. Advances in Intelligent Systems and Computing, vol 645. Springer, Singapore. https://doi.org/10.1007/978-981-10-7200-0_15.

[20] C. Lai, R. Hwang and H. Chao, "A QoS Aware Resource Allocation Strategy for Mobile Graphics Rendering With Cloud Support," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 1, pp. 110-124, Jan. 2017, doi: 10.1109/TCSVT.2016.2589740.

[21] X.Wang, T. Kwon, Y. Choi, H.Wang, and J. Liu, ``Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users,'' IEEE Wireless Commun., vol. 20, no. 3, pp. 72_79, Jun. 2013.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

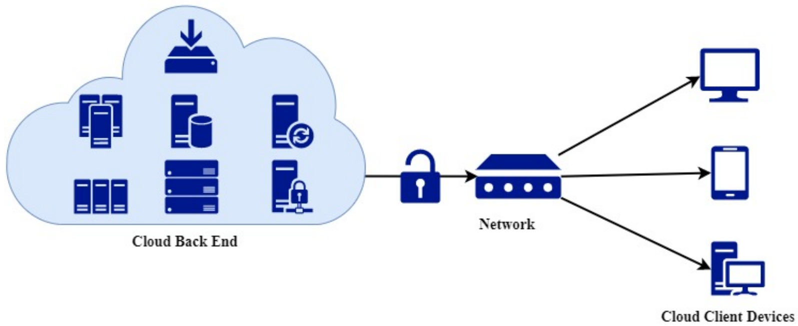

Figure 1. Traditional Client Server Model For Cloud Computing.

Table 2 Fuzzy Membership Functions For Effective Qos Management

Parameters		**Membership functions**			
Input	Cloud Client Network bandwidth	Above SLA	According to SLA	Below SLA	According to SLA
	Cloud CPU Utilization	Above SLA	According to SLA	Below SLA	According to SLA
	Cloud Client Main Memory Status	Above SLA	According to SLA	Below SLA	According to SLA
	Cloud Service	Optimal Service	Optimal Service	Optimal Service	Optimal Service
Output	Penalty and customer satisfaction level	[0-3]	[4-6]	[7-9]	10

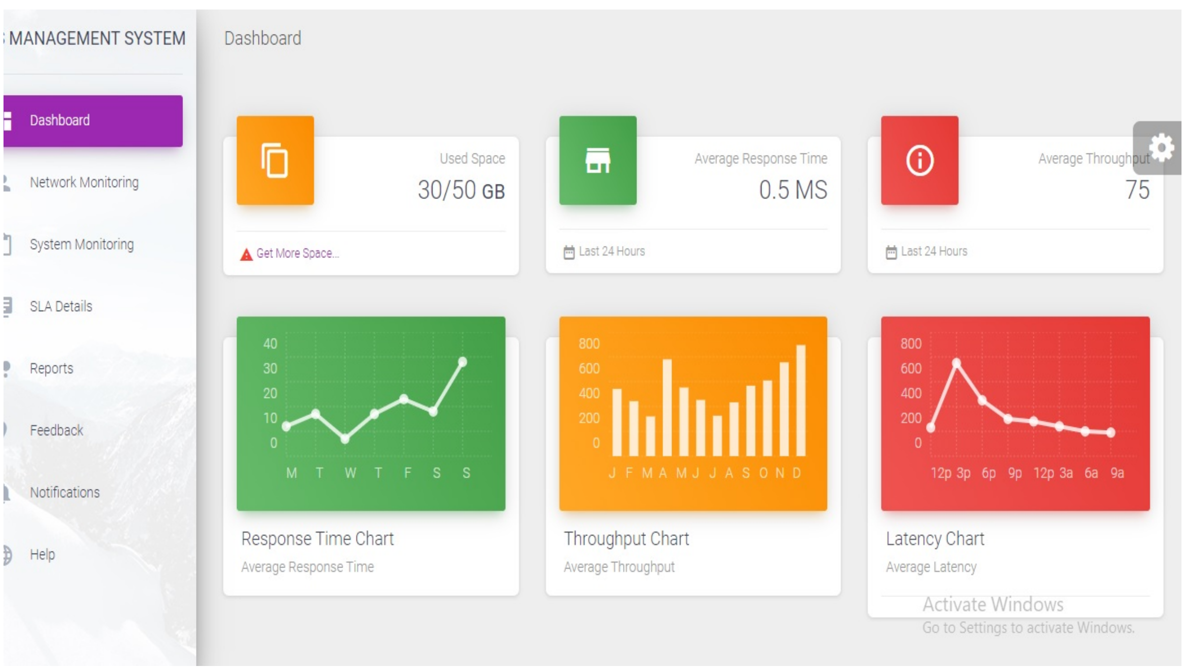

Figure 3: Dashboard Of The Proposed Qos Management And Monitoring System.

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

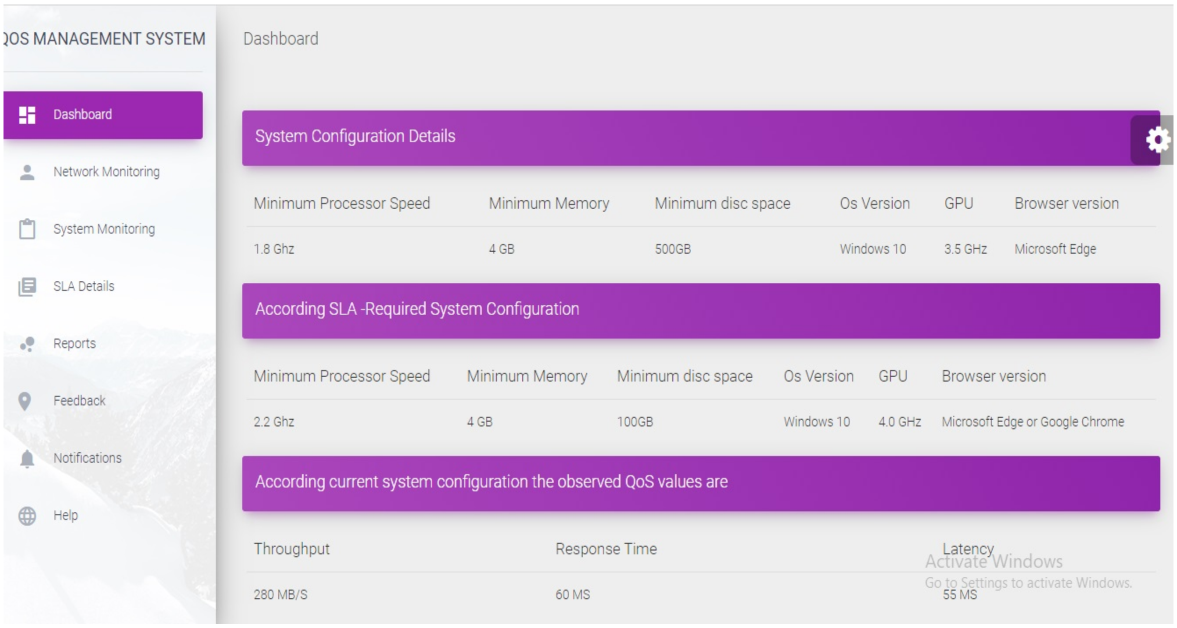

Figure 4: Screen For Tracking The Cloud Client System Configuration.

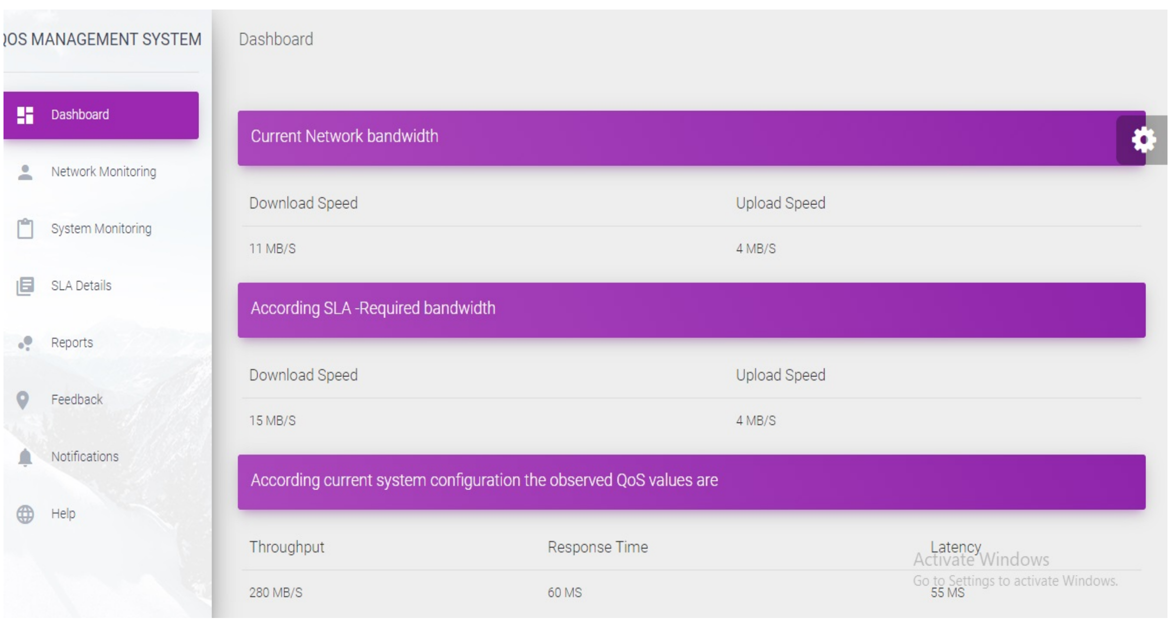

Figure 5: Information About The Cloud Client System's Network Bandwidth.

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

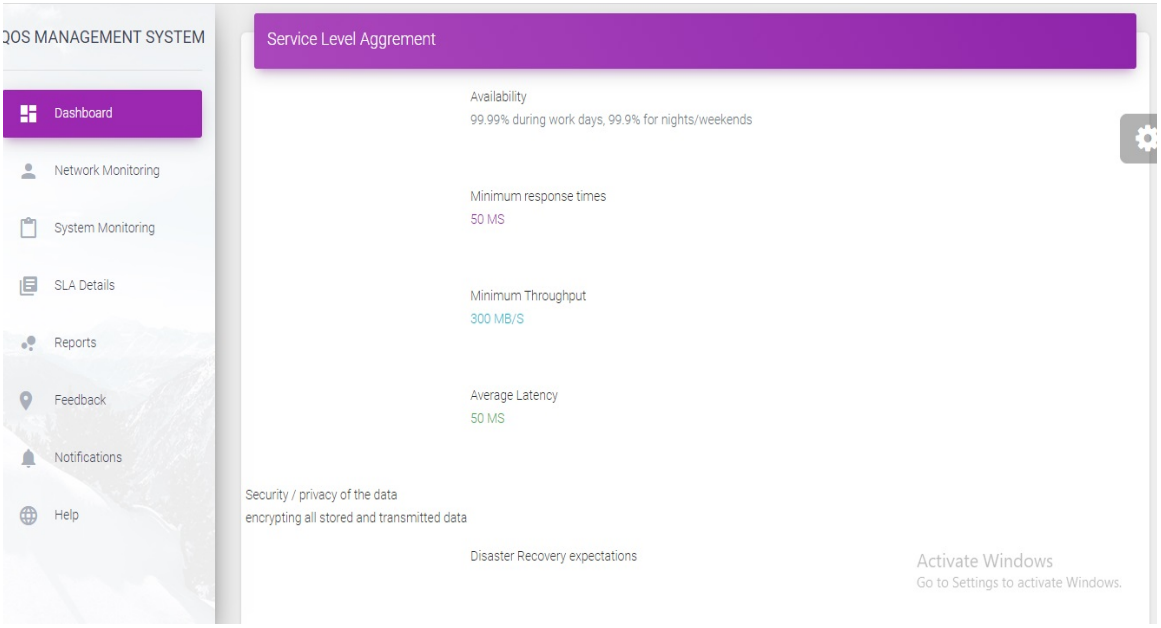

Figure 6: Specific Terms And Conditions Of The Service Level Agreement.

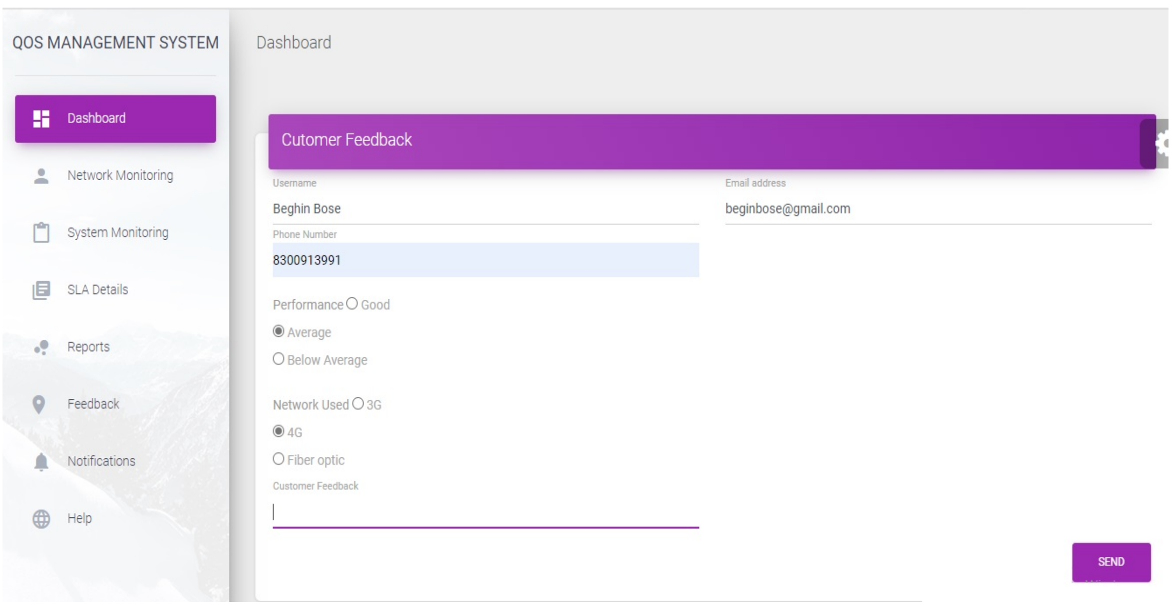

Figure 7: Customer Feedback Form For Cloud Customer Communication.