ISSN 2278-3091

Volume 9, No.5, September - October 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse106952020.pdf

https://doi.org/10.30534/ijatcse/2020/106952020

Multilevel Priority Queue Scheduling (MPQS) - An efficient hybrid protocol requests scheduling algorithm

Poonam Gupta[1], K V V Satyanarayana[2], Dilip Shah[3]

[1]Research Scholar, Department Of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India - 522502,
poonam77gupta@gmail.com

[2]Professor, Department Of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India - 522502,

[3]Vice Chancellor, G H Raisoni University, Amaravati, Maharashtra, India- 444701

ABSTRACT

As Internet developing fast towards the future “Internet of Things” (IoT) having potential to connect billions / trillions of devices would require to produce requests at a very high speed[1]. The traditional hardware and technology available may not be able to tackle certain challenges in processing all requests as required by the end user. With the multiple protocols this overhead increases. To meet the users expectations the scheduling algorithms need an improvement.

In this paper we explore related algorithms their limitations and present our work in Multilevel Priority Queue Scheduling Algorithm (MPQS) and Optimized Multilevel Priority Queue Scheduling Algorithm (OMPQS).

Key words: IoT, Multilevel Queue, Protocol, Scheduling algorithm, cloud

1. INTRODUCTION

In this the era of IoT, as heard IoT is making front line topic[1]. Soon, everything would be connected over the internet. With this communication network of IoT all devices and users/ clients are connected to receive and transmit required information. Performance measures are important for making the communication useful and reliable. The IoT is an ever growing connected network with all devices collecting and sending data to IoT platforms. IoT based application areas are wide and huge, across all the sectors shopping, health, transport etc.

Mainly, the Internet of Things is developed via connected devices that send and receive requests to be managed for performing operations and taking decisions. The requests process across the application to the information queue to regulate the enormous amount of requests/ tasks generated by the IoT. Here the basic aim of queue is to schedule tasks for the IoT activities functioning efficiently.

Since IoT is a queuing system controlled, it involves requests / information sending and receiving out from the system with certain arrival patterns how they have been executed. The execution of the tasks is very much dependent on the burst of the inputs into the system. The requests are very much dependent on how the IoT devices functioning well. This whole end to end working highly depends on the scheduling algorithm.

In this paper, we are proposing scheduling algorithms to depict the efficient processing of Internet of Things requests from hybrid protocol. To explain the same, rest of the paper has different sections, in section 2 related work are discussed section 3 deals with the overview of proposed system, section 4 shows the results and last 5 section is conclusion.

So far in queuing algorithms [8] the execution time for tasks is divided within multiple queues where higher level queues always gets priority while scheduled first for execution, so the tasks with low priority may starve for execution. Tasks in these queues use various scheduling methods. High level Priority Processes are generally scheduled using round robin scheduling technique [5, 9], while low level priority processes belong to in lower level queues is usually scheduled using the traditional FCFS [2] scheduling technique.

We have proposed a Multi-Level Optimised Feedback Queue, where the tasks are categorized as per the protocol and then further assigned to respective Priority Queues. The tasks in the queues are then executed sequentially while following the execution ratio assigned. The execution ratio is then changed dynamically per the feedback received from the status of the tasks execution for the respective queues.

2. RELATED WORK

With the recent development in this era of technology and revolutions in Internet of Things (IoT) world, day to day life expects better performances [1–2]. With every application handling various activities and tasks, these tasks are scheduled dynamically as per the need of the user so is required to be executed via different clients, and solely based on the arrival time of these tasks and non-pre-emptive in nature. The biggest challenge in task scheduling is how well the tasks are processed in the heterogeneous computing environment, so that these tasks can be executed within a stipulated time period or the schedule length can be minimized.

With overlarge number of tasks queued, the capacities of the system with the new tasks entering may be reduced. With task scheduling (TS) algorithms [3] any task in a distributed system, helps to make better use of the resource capabilities by proper task schedule among different clients.
The performance of the system can be measured by many parameters like overall completion time, response time of a distributed system.
With heterogeneous environments, their capabilities need to be considered during designing scheduling algorithms [4-5].
The priority of applications vary for all the sectors, a banking application would require higher priority than any shopping application [6-7].
In this paper, the "P" (Priority) level term has been used to indicate the differences among H, M & L priorities. The expectations for Priority levels are bond to the SLA and terms agreed. Assigning priority levels helps scheduling the requests for tasks and so, TS (Task Scheduling) algorithms must use resources efficiently while satisfying expectations of various IoT requests. Here we aim with this work a TS algorithm by considering priority assigned to IoT requests. In this paper, we focus on priority-based queuing and scheduling algorithms because the priority-based scheduling algorithm is more realistic for IoT due to the existence of different classes of non-standardization applications. There are various scheduling techniques architected for scheduling processes,
The various scheduling techniques used so far are:
FirstComeFirstServed (FCFS), in this scheduling technique [8] the tasks in execution queue are scheduled in the same order by which they are inserted in the queue, so all the tasks have the same execution priority.
Round Robin Scheduling (SJF), here the tasks in the queue are share the time equally so the tasks gets equal time. While tasks don't face any starvation problem but the tasks with high priority may not execute as required.
Shortest Job first (SJF), in this scheduling the small jobs are given the highest priority.
Dedicated Server Scheduling (DDS) Architecture [6] as shown in Figure No. 1.

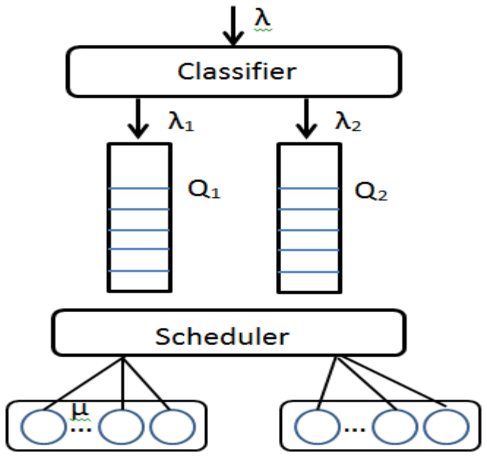

Figure No. 1: Dedicated Server Scheduling (DDS) Architecture [6]

In DSS architecture has fixed servers, the requests are classified in two classes example: (C1) request and (C2) request traffic flows: such as requests of banking and heath applications. Some servers process C1 while the others process C2. C1 and C2 arrive at Arrival Rate λ1 and λ2. These requests are buffered in Q1 and Q2 buffers. Consider the services are processed at the rate of μ, while all servers can serve all types of requests.

Dynamic Dedicated Server Scheduling (DDSS) Architecture [6] as shown in Figure No. 2. In DDSS architecture servers are updated dynamically, the classified requests are assigned to the respective servers only so the traffic is not shared amongst other servers.

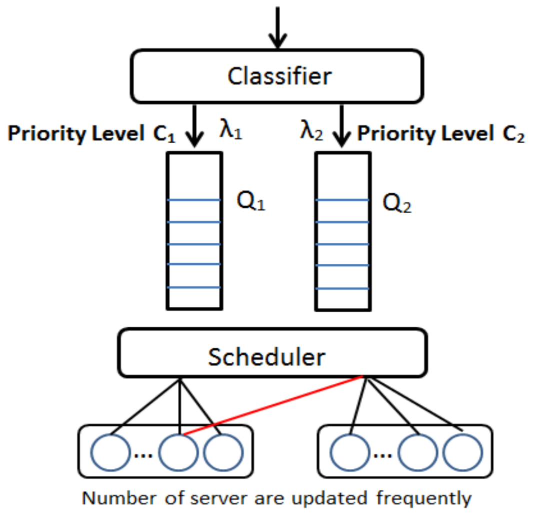

Figure No. 2: Dynamic Dedicated Server Scheduling (DDSS) Architecture [6]

So with increase in arrival rate of one type of requests (example C1) may hamper the throughput although other servers may be available to process the request.

Heterogeneous Dynamic Dedicated Servers Scheduling (h-DDSS) Architecture [7] as shown in Figure No. 3.

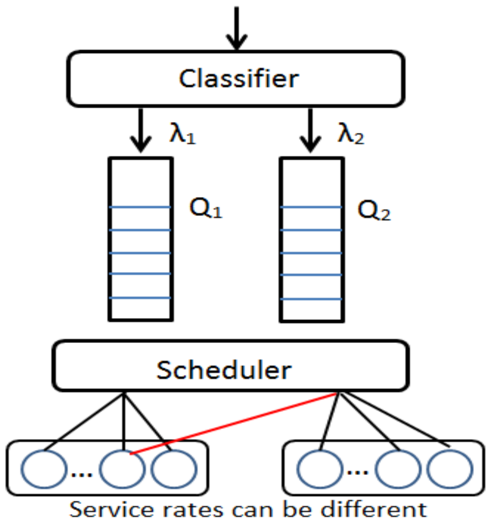

Figure No. 3: Heterogeneous Dynamic Dedicated Servers Scheduling (h-DDSS) Architecture [7]

Multilevel queue technique is most referred In MLQ the task is further categorised in multiple queues depending on the classification parameter set. With traditional approach all queues are provided a priority level - High and Low. The tasks in the H-level queues is assigned a higher priority than the Tasks in L-level queue. The disadvantage here is tasks in L queues may starve of resources while the H Level tasks are getting processed.

In cloud computing Multi-Queue Job Scheduling is used thisapproach improves the scheduler combining the various burst time centered jobs into queue. All the jobs submitted by the clients are sorted on the basis of burst time in ascending order by giving importance to all the jobs. As clients requirements are changing based on the current need, this scheduling helps to raise the client approval and contentment. The main aim of this system is to decrease starvation by using dynamic allocation for jobs. From the pool of jobs the best suitable job is selected also maintain the performance of the system. The various resources are provided by queue manager for the underlying network. The queue manager is very vital part of could computing system and responsible to manage the utilization of all available resources. In this system total three queues are made small, medium and long depending on burst time. The small queue consists of first 40% jobs, medium queue has next 40% jobs and long queue has remaining 20 % jobs. This method gives importance to all the jobs because many number of client in cloud computing and each client wants more needs and expectation. It grants equal importance for all in dynamic selection. The architecture of MQS using dynamic selection of jobs based on the sorted burst time. The client submitted jobs are enter in to the service provider it consists of queue manager that sort the jobs in ascending based on burst time. The jobs are executed based on dynamic selection and enter in to cloud environment. The best allocation reduces the time and availability of space in an effective manner without compensating the quality of the system and customer needs [8-17].

3. PROPOSED SYTEM ARCHITECTURE

The proposed system gets request from various IoT applications users. All these applications are based on different protocols such as Message Queuing Telemetry Protocol (MQTT), constrained application Protocol (CoAP) and WebSocket. Similarly every application has assigned priority High/ Medium/ Low depending on the services it is providing. So whenever any request enters a system first its protocol is identified and directed to appropriate server/ broker where it is executed according to its priority. Here it is to be noted that the priority of request is same as that of the application to which it is intended. The figure no. 4 shows the processing flow of the request received in the system [18-30]. As depicted in figure No. 4 all the incoming requests in input queue are handled by hybrid protocol handler, it further forwards the requests to respective protocol queue as per the request's priority.

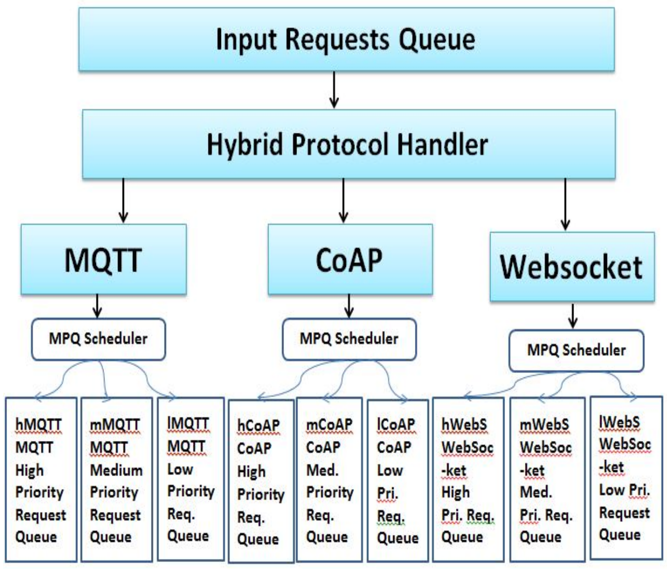
Input Requests Queue
Hybrid Protocol Handler
MQTT
CoAP
Websocket
MPQ Scheduler
MPQ Scheduler
MPQ Scheduler
hMQTT MQTT High Priority Request Queue
mMQTT MQTT Medium Priority Request Queue
lMQTT MQTT Low Priority Req. Queue
hCoAP CoAP High Priority Req. Queue
mCoAP CoAP Med. Priority Req. Queue
lCoAP CoAP Low Pri. Req. Queue
hWebS WebSoc -ket High Pri. Req. Queue
mWebS WebSoc -ket Med. Pri. Req. Queue
lWebS WebSoc -ket Low Pri. Request Queue

Figure No. 4: Multilevel Priority Queue Scheduling (MPQS) Architecture

All these requests scheduling is carried out by Multilevel Priority Queue (MPQ) scheduler. It uses Multilevel Priority Queue Scheduling (MPQS) algorithm to execute requests in multilevel priority queues.

The Multilevel Priority Queue Scheduling (MPQS) algorithm is given below:

Input: request from client/application

Output: processing of request as the application priority

Step 1) Initialize priority for every application in the Beginning

Step 2) Receive application Request, Push the request in appropriate Queue

```
If ReqType == MQTT
            If ReqPri == High
                  hMQTT [i]= Request; i=i+1;
            else If ReqPri == Medium
                  mMQTT [i]= Request; i=i+1;
            else If ReqPri == Low
                  lMQTT [i]= Request; i=i+1;
      Else if ReqType == CoAP
            If ReqPri == High
                  hCoAP [i]= Request; i=i+1;
            else If ReqPri == Medium
                  mCoAP [i]= Request; i=i+1;
            else If ReqPri == Low
                  lCoAP [i]= Request; i=i+1;
      Else if ReqType == WebSocket
            If ReqPri == High
                  hWebSocket [i]= Request; i=i+1;
            else If ReqPri == Medium
                  mWebSocket [i]= Request; i=i+1;
            else If ReqPri == Low
```

lWebSocket [i]= Request; i=i+1;

Step 3) Process all high priority request of all protocol type

Step 4)ifhigh priority queue is empty process next request in medium priority queue.

Go to step 4.

Step 5)if high priority && Medium priority queues are empty process next request in low priority queue.

Go to step 4.

Althoughthe algorithms mentioned above may help in scheduling the request and perform better compare to FCFS algorithm as shown in Figure No.5-7 for all types of requests high, medium low priority. But it has one drawback that low level requests suffer from starvation problem. Soto handle real time requests and solve the starvation problem, we propose OptimisedMultilevel Priority Queue Scheduling (OMPQS) algorithm. In OMPQS algorithm instead of executing all high level priority request then medium priority request and then at last low priority requests ratio is decided that is after executing every 16 high priority requests, 4 medium priority requests will executed followed by 1 low priority request will be executed and again if there are still pending requests in high priority queue again same ratio will be followed. This mechanism helps to solve starvation problem at certain extend.

The process with greater priority will get major share in execution time to meet the deadline condition. In this way, this algorithm will meet the deadline condition which is the property of real time systems. Hence the real time requests will also not suffer from increased waiting time.

The system consists of three fundamentals operations, first to maintain a queue of all the requests. If a new request enters, it is appended at the end of the queue. Second, as per the request protocol and priority it is shifted to appropriate queue. And depending on the priority of the request it is executed with the ratio 16:4:1. The priority queues (H:M: L) as assigned execution proportion with the count (16: 4: 1).the ratio above helps avoid major starvation caused for the tasks in the queue with Low priority during the peak periods. The ratio is decided per the experience and the feedback loop running in the current execution.

1. Create Multi-Level Priority Queue (MPQ).
2. Classify the requests to particular protocol queue (MQTT, CoAP, WebSocket)
3. Process the requests in the queue with High Priority, switch to the medium Queue if the request in High priority queue get over or the count has reached the ratio of 16.
4. Process the tasks in the queue with Medium Priority, switch to the Low Priority Queue if the tasks in Medium priority get over or count has reached the ratio of 4.
5. Process the tasks in the queue with Low Priority, switch back to the High Priority Queue if the tasks in Low priority get over or count has reached the ratio of 1.
6. Go to step #2

4. RESULTS

The performance of multilevel priority queue scheduling for high, medium and low priority is shown in following Figure No. 4, Figure No. 5, Figure No. 6 respectively on the basis of total waiting time and average waiting time.

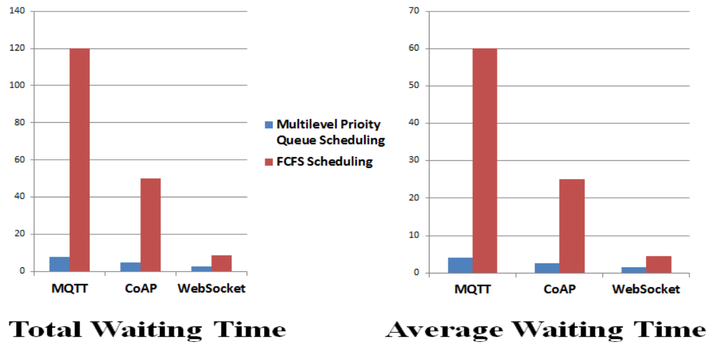

Figure No. 5: Total Waiting Time and Average Waiting Time for High Priority Requests

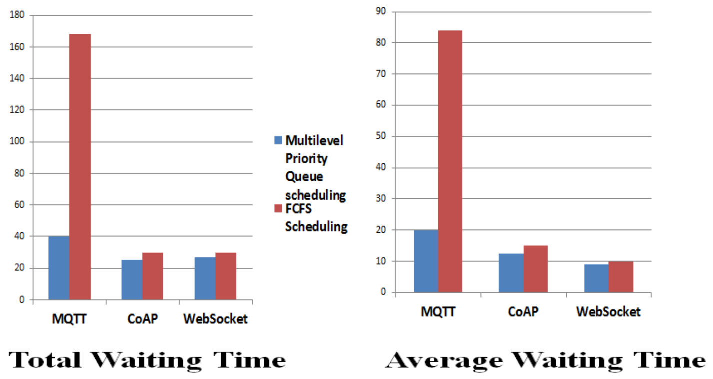

Figure No. 6: Total Waiting Time and Average Waiting Time for Medium Priority Requests

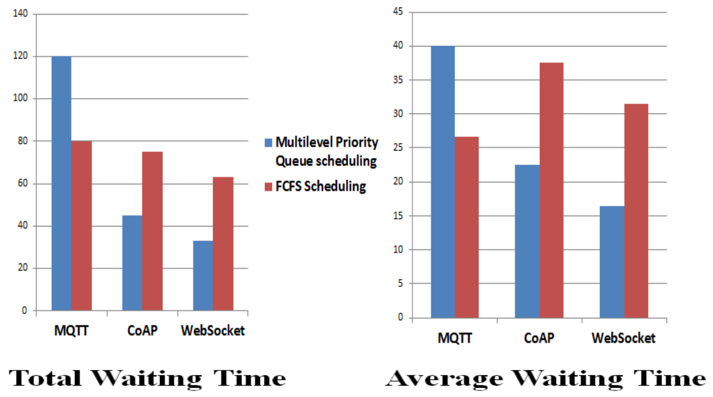

Figure No. 7: Total Waiting Time and Average Waiting Time for Low Priority Requests

General observation is low priority requests are starving to address this issue Optimized Multilevel Priority Queue Scheduling is implemented and the following figure No. 8-10depict its performance.

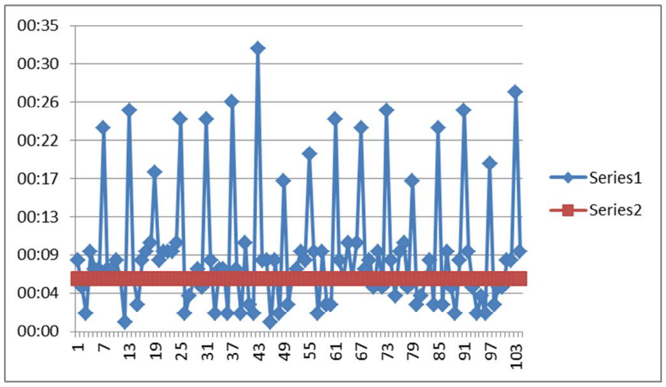

Figure No. 8: OMLPQS algorithm with ratio 16:4:1

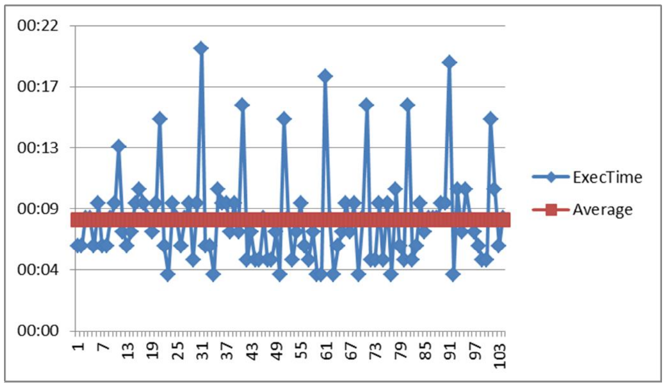

Figure No. 9: OMPQS algorithm with ratio 10:3:1

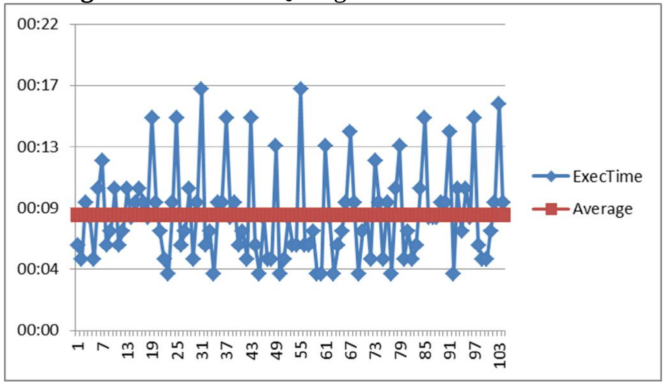

Figure No. 10: OMPQS algorithm with ratio 6:2:1

5. CONCLUSION

An efficient hybrid protocol request scheduling algorithm is Multilevel Priority Queue Scheduling (MPQS). It reduces the waiting time of requests at server/broker end and handles real time hybrid protocols requests efficiently. It has been observed that low priority requests have to wait a lot to . To address this issue Optimized Multilevel Priority Queue Scheduling (OMPQS) is implemented where the various priority requests are executed as per the ratio assigned. Hence MPQS and OMPQS are useful in scheduling real time hybrid protocol requests.

ACKNOWLEDGEMENT

We would like to thank K L University,Vijayawada and G. H. Raisoni college of Engineering and Management, Pune for providing constant encouragement and support to complete this research work.

REFERENCES

1. J. Pan and J. McElhannon, **Future Edge Cloud and Edge Computing for Internet of Things Applications,** in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439-449, Feb. 2018, doi: 10.1109/JIOT.2017.2767608.
2. T. Qiu, N. Chen, K. Li, M. Atiquzzaman and W. Zhao, **How Can Heterogeneous Internet of Things Build Our Future: A Survey,** in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2011-2027, thirdquarter 2018, doi: 10.1109/COMST.2018.2803740.
3. Sravani, M., Kumar, K. R., & Rahul Babu, B.,**Efficient usage of natural resources to automation of agriculture using iot,** International Journal of Innovative Technology and Exploring Engineering, 8(5), 250-254, 2019.
4. Pavan Kumar, T., Lala, S. K., Sravani, B., & Sandeep, A., **Internet of things survey on crop field smart irrigation automation using IOT,** International Journal of Engineering and Technology(UAE), 7(2.8 Special Issue 8), 503-506, 2018
5. Y. Jiang and J. Jiang, **Contextual Resource Negotiation-Based Task Allocation and Load Balancing in Complex Software Systems,** in *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 5, pp. 641-653, May 2009, doi: 10.1109/TPDS.2008.133.
6. B. Hong and V. Prasanna, **Adaptive Allocation of Independent Tasks to Maximize Throughput**, in *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 10, pp. 1420-1435, Oct. 2007, doi: 10.1109/TPDS.2007.1042.
7. Q. Zhang, M. F. Zhani, R. Boutaba and J. L. Hellerstein, **Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud**, in *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 14-28, Jan.-March 2014, doi: 10.1109/TCC.2014.2306427.
8. H. S. Narman, M. S. Hossain and M. Atiquzzaman, **DDSS: Dynamic dedicated servers scheduling for multi priority level classes in cloud computing**,*2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW, 2014, pp. 3082-3087, doi: 10.1109/ICC.2014.6883794.
9. H. S. Narman, M. S. Hossain and M. Atiquzzaman, **h-DDSS: Heterogeneous Dynamic Dedicated servers scheduling in cloud computing**,*2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW, 2014, pp. 3475-3480, doi: 10.1109/ICC.2014.6883859
10. A. V. Karthick, E. Ramaraj and R. G. Subramanian,**An Efficient Multi Queue Job Scheduling for Cloud Computing**,*2014 World Congress on Computing and Communication Technologies*, Trichirappalli, 2014, pp. 164-166, doi: 10.1109/WCCCT.2014.8.
11. X. Li and P. Lun, **Research and Improvement of Real-Time Queue Scheduling Algorithm**,*2010 International Forum on Information Technology and*

Applications, Kunming, 2010, pp. 102-104, doi: 10.1109/IFITA.2010.312.

12. J. Tan and S. G. M. Koo, **A Survey of Technologies in Internet of Things**,*2014 IEEE International Conference on Distributed Computing in Sensor Systems*, Marina Del Rey, CA, 2014, pp. 269-274, doi: 10.1109/DCOSS.2014.45.
13. S.Vanithamani and N. Mahendran, **Performance analysis of queue based scheduling schemes in wireless sensor networks**,*2014 International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, 2014, pp. 1-6, doi: 10.1109/ECS.2014.6892593.
14. M. M. Hamdi, S. A. Rashid, M. Ismail, M. A. Altahrawi, M. F. Mansor and M. K. AbuFoul, **Performance Evaluation of Active Queue Management Algorithms in Large Network**,*2018 IEEE 4th International Symposium on Telecommunication Technologies (ISTT)*, Selangor, Malaysia, 2018, pp. 1-6, doi: 10.1109/ISTT.2018.8701716.
15. D. S. Nguyen, **Application of queuing theory in service design**, *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, 2017, pp. 837-840, doi: 10.1109/IEEM.2017.8290009.
16. H. B. Parekh and S. Chaudhari, **Improved Round Robin CPU scheduling algorithm: Round Robin, Shortest Job First and priority algorithm coupled to increase throughput and decrease waiting time and turnaround time**, *International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pp. 184-187, doi: 10.1109/ICGTSPICC.2016.7955294, 2016.
17. Ashish, M.M.S. Rauthan,Varun Barthwal,Rohan Varma, **The Latest Review: Scheduling Algorithms On The Cloud Computing Based Environment**, International Journal Of Scientific & Technology Research Volume 9, Issue 01, ISSN 2277-8616, January 2020.
18. M. Li and Y. Fang, **Research and Application of the Queuing Theory of the Virtualization Technology Resource Sharing Mode**, *2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, Changsha, 2016, pp. 245-248, doi: 10.1109/ICITBS.2016.123.
19. T. Biswas, P. Kuila and A. K. Ray, **Multi-level queue for task scheduling in heterogeneous distributed computing system**,*2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, pp. 1-6, doi: 10.1109/ICACCS.2017.8014673, 2017.
20. Siva Nageswara Rao, G., Jayaraman, R., &Srinivasu, S. V. N. **Efficient PIMRR algorithm based on scheduling measures for improving real time systems,** International Journal of Engineering and Technology(UAE), pp. 275-278 ,2018.
21. Balaji K., Sai Kiran P., **Efficient resource allocation algorithm with optimal throughput in cloud computing,** Journal of Advanced Research in Dynamical and Control Systems, 9, pp 1902-1910, 2017.
22. V M Mohan, K V VSatyanarayana, **Efficient task scheduling strategy towards QoS aware optimal resource utilization in cloud computing**, journal of Theoretical and Applied Information Technology 80 (1), pp 152 – 159, 2015.
23. V M Mohan, K V VSatyanarayana, **The Contemporary Affirmation of Taxonomy and Recent Literature on Workflow Scheduling and Management in Cloud Computing**, Global Journal of Computer Science and Technology 16 (1), pp 1-16, 2016.
24. P Gupta, KVV Satyanarayan, DD Shah, **Development and Testing of Message Scheduling Middleware Algorithm with SOA for Message Traffic Control in IoT Environment,** International Journal of IntelligentEngineering and Systems 11 (5), pp 301-313, 2018
25. Yuga Vamshi, B., Nikhil Sai, M., Ali Hussain, M**. IoT based smart appointment alert system,** International Journal of Innovative Technology and Exploring Engineering, 8(5), 956-959, 2019.
26. Gavvala, SK; Jatoth, C; Gangadharan, GR; Buyya, R **QoS-aware cloud service composition using eagle strategy**, Future Generation Computer Systems-The International Journal of science, pp 19-31, Jan 2019
27. Rao, P. Ravinder; Sucharita, V A,**Framework to Automate Cloud based Service Attacks Detection and Prevention**, International Journal Of Advanced Computer Science And Applications, Vol. 10 Issue. 2 pp 241-250, Feb 2019.
28. Priyanka Sharma,Sanjay Shilakari,UdayChourasia, Priyanka Dixit, Alpana Pandey,**A Survey On Various Types Of Task Scheduling Algorithm In Cloud Computing Environment**International Journal Of Scientific & Technology Research Volume 9, Issue 01, ISSN 2277-8616, January 20
29. Ahmad AlauddinAriffin, Aws Fadhil Ibrahim, Sazlinah Hasan, RohayaLatip, **An Efficient Virtual Machine Scheduling Algorithm ToMinimize Makespan And MaximizeProfit Using Hyper Heuristic Approach,** International Journal of Advanced Trends in Computer Science and Engineering, ISSN 2278-3091, Volume 8, No.1.4, 2019
30. G.Kiruthiga, Dr. S. Mary Vennila, **An Enriched Chaotic Quantum Whale Optimization Algorithm Based Job scheduling in Cloud Computing Environment**, International Journal of Advanced Trends in Computer Science and Engineering, 6(4) pp. 1753-1760, August 2019, DOI: 10.30534/ijatcse/2019/105842019