INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 11, November 2026 | ISSN: 2456-4184 | IJNRD.ORG

Efficient CPU Scheduling Through an Adaptive Round Robin Mechanism

Dr. Raj pal Singh
AMICT, Pathankot

Mr. Bhuvnesh Pathania
AMIMT, Pathankot

ABSTRACT

Every process which runs in an operating system is processed through multiprogramming, by sharing the processor. Multiprogramming is a capability of the operating system to execute one or a lot of programs in a single processing machine. Multiprogramming's main objective is to optimize the use of CPUs, throughput and minimize average waiting time (AWT), the average turnaround time (ATT), average response time (ART) and the number of context switches (NOC). Multi-programming can be performed by several CPU scheduling algorithms. Most preferable scheduling algorithm among all is Round robin (RR). RR algorithm is mainly depended on time quantum which is stationary. Hence, the proposed work is new approach to the round-robin scheduling algorithm (RR algorithm) with an enhanced time quantum-based algorithm. This enhancement of RR algorithm by using dynamic time quantum leads to minimize AWT, ATT, ART and NOC. This approach inherits the properties of Round robin, shortest job first (SJF) algorithm and first come first serve algorithm (FCFS). Therefore, the proposed algorithm is a hybrid round robin scheduling mechanism for process management (HYRR Mechanism). HYRR Mechanism is an innovative scheduling algorithm which reduces ATT, ART, AWT and NOC to the desired levels.

Keywords

Scheduling Algorithm, number of context switches, Process Management, Dynamic time Quantum.

1. INTRODUCTION

Operating system (OS) organizes the processes by allotting different tasks properly according to their arrival time [1]. To increase the CPU utilization, Operating system's primary role is to allow many processes concurrently [2, 3]. There are different types of CPU scheduling algorithm which perform differently according to the situation, mainly CPU-scheduling algorithms concentrate on increasing CPU- performance by reducing turnaround time, the number of the context switch and waiting time [4, 5]. In this work, the proposed algorithm improves the capability of the round-robin-algorithm by introducing a new enhanced quantum time in place of the time quantum in round-robin.

The proposed work inherits the properties of RR Algorithm, SJF algorithm and other algorithms. In SJF Algorithm, length of the process is associated with CPU. Lengths are to schedule the process with the shortest time [6]. Once the Processor is free, it is given to the process which has the least next CPU. It is also called as shortest CPU [7]. FCFS scheduling is used for disrupting the knot when next two processes have same-length [8]. SJF stays ideal, so it gives least AWT for process [9]. In RR algorithm, a fixed time slot is cyclically allocated to each process [10]. Most widely used CPU scheduling strategies as a foundation. Because of time slots, processes don't suffer starvation [11]. More number of context switches takes place which is a big disadvantage of RR Algorithm [12]. In this paper, the proposed mechanism tries to reduce NOC. Some of the basic terms used in this paper are, waiting time: the time that a process stays or waits in the ready queue [13].

Turnaround time: The time it takes for a CPU to execute a process. It is measured as the time between the request and completion of a phase [14].

CPU utilization: CPU is utilized to its extreme extents by keeping CPU busy.

Response time: Amount of time taken by process to get response from its initial request [15].

Throughput: Throughput is the rate of completion of processes per unit of time [14].

Context switch: switching of one process in CPU with another process which already is present in ready queue. While switching, the state of a currently executing process is preserved and CPU is loaded with a new process from its saved state is called context switch [16].

Therefore, to be a suitable scheduling algorithm, it must possess characteristics such as minimum waiting time, minimum turnaround time, maximum CPU utilization, minimum response time, maximum throughput and a minimum number of the context switch [17].

2. PROPOSED HYRR MECHANISM

This algorithm mainly focuses on improvement of Round robin scheduling algorithm. It maximizes CPU utilization, throughput and minimizes AWT, ATT, ART and NOC. This proposed algorithm works more effectively than the RR Algorithm. In HYRR Mechanism, time quantum is calculated dynamically by using the mean of burst time and the minimum of burst time. In phase-1, Enhanced quantum time (ETQ)is calculated using mean and minimum of burst time. After calculation, Process which has shortest burst time and which is not executed in CPU is given high priority and allocated in CPU for 1 Enhanced quantum time. Phase-1 is repeated until every process gets allocated in CPU for once. In phase-2, ready queue processes are ordered in ascending based on their remaining burst time. After arrangement, process which is first in ready queue is allocated in CPU for 1 enhanced quantum time. After allocation, if currently executing process remained burst time in CPU is less than or equivalent to 1ETQ then current process is reallocated in CPU for its execution. phase-2 is repeated until the ready queue becomes empty.

2.1 Flow Chart

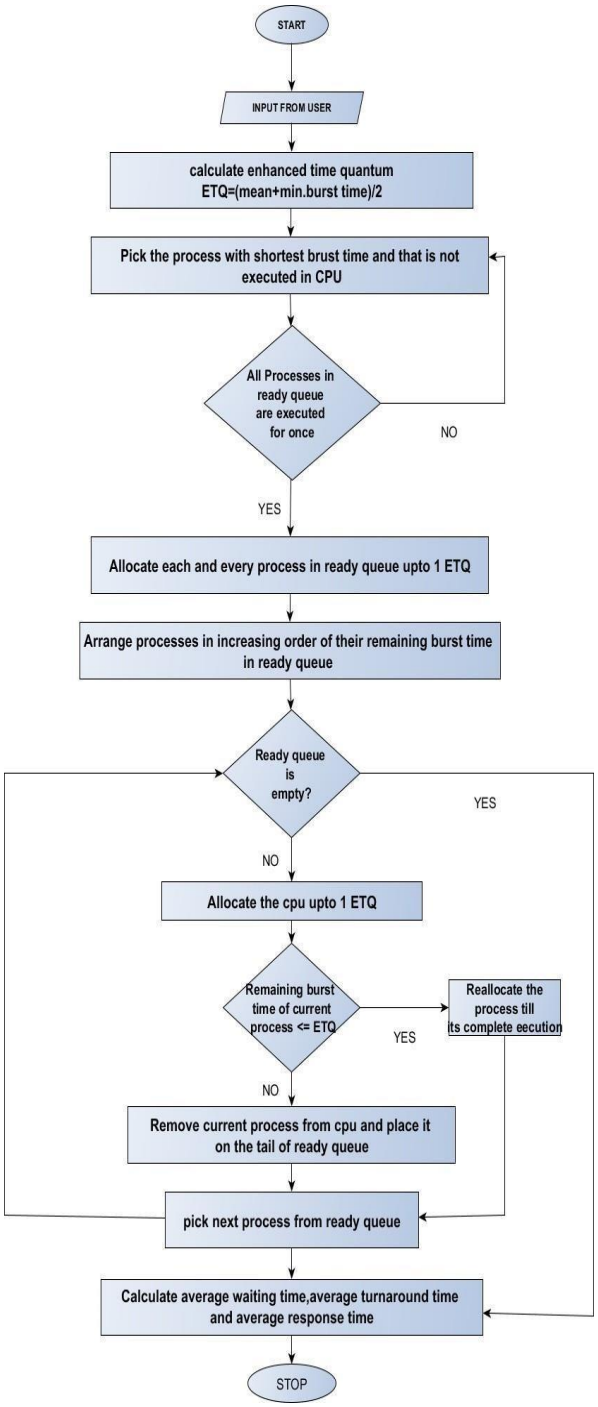

Fig-1: Proposed HYRR mechanism

2.2 HYRR Algorithm

Step-1. Calculate average of CPU time of all process. Average= (P1+P2+P3+P4 PN)/N

Step-2. Calculate enhanced quantum time by given formula. Enhanced quantum time= ceiling (average + minimum time)/2

Step-3. Select the process with minimum time. which is not executed even for 1time in CPU.

Step-4. Allocate the selected process to CPU from ready queue up to 1 Enhanced quantum time and repeat STEP-3 until all processes in get executed for one time in CPU.

Step-5. After completion of step-4 order all the processes in ascending order of their remaining burst time with in the Ready queue state.

Step-6. A Process which is first in ready queue state is allocated to CPU for Enhanced Quantum time.

Step-7. The current process is reallocated in CPU if remaining process burst time is less than or equivalent to 1ETQ. If Process gets entirely executed in CPU, then it is discarded from ready queue state Else, Place the current process at the end of the ready queue for executing it in future.

Step-8. Allocate following process from the queue to CPU for the period of enhanced quantum time and recheck the step-7.

Step-9. Execute all processes until the queue is empty. Step-10. Calculate NOC, ATT, AWT and ART of processes.

3. EXPERIMENT AND RESULTS INVESTIGATION

Different processes are outlined with CPU Burst time and arrival time. These processes are scheduled in both HYRR Mechanism and RR Algorithm. After scheduling, ATT, AWT, ART, NOC and results are compared.

3.1 Experiment-1

In this experiment five process are defined only with Burst time as Pr0, Pr1, Pr2, Pr3, and Pr4. Time quantum of RR Algorithm=3ms.

Table1. Process Id with their burst time

PROCESS-ID	TIME
Pr0	15
Pr1	6
Pr2	1
Pr3	4
Pr4	11

3.1.1 RR-ALGORITHM:

Let time quantum be 5ms.

3.1.1.1 Gantt chart:

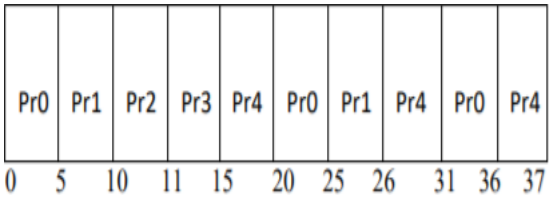

Fig-2:Gnatt chart

Table 2. RR-Algorithm Results

PROCESS ID	WAITING TIME (ms)	TURN AROUND TIME (ms)	RESPONSE TIME (ms)
Pr0	21	36	0
Pr1	20	26	5
Pr2	10	11	10
Pr3	11	15	11
Pr4	26	37	15

Therefore,

NOC=9

AWT= 17.6ms

ATT=25ms

ART=8.2ms

3.1.2 Proposed Hybrid Round Robin Scheduling Mechanism(HYRR):

Mean=7.4ms

Minimum burst time=1ms
ETQ=ceiling(mean+minimum)/2=ceil ((7.4+1)/2) =5ms
ETQ=5ms

3.1.2.1 Gantt chart:

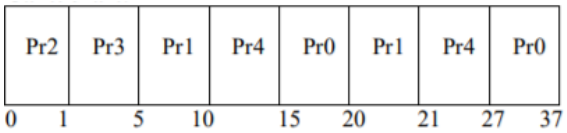

Fig 3: GNATT chart

Table3. HYRR Performance Analysis

PROCESS ID	WAITING TIME (ms)	TURN AROUND TIME (ms)	RESPONSE TIME (ms)
Pr0	22	37	15
Pr1	15	21	5
Pr2	0	1	0
Pr3	1	5	1
Pr4	16	27	10

From Table 4 and Fig 4, the NOC, ART, ATT and AWT of proposed HYRR Mechanism has drastically decreased when compared with RR algorithm.

Table 4. Experiment-1 Result analysis based on processes arriving at same time.

ALGORITHM	RR ALGORITHM	HYRR MECHANISM (Proposed Algorithm)
AWT	17.6	10.8ms
ATT	25ms	18.2ms
ART	8.2ms	6.2ms
NOC	9	7

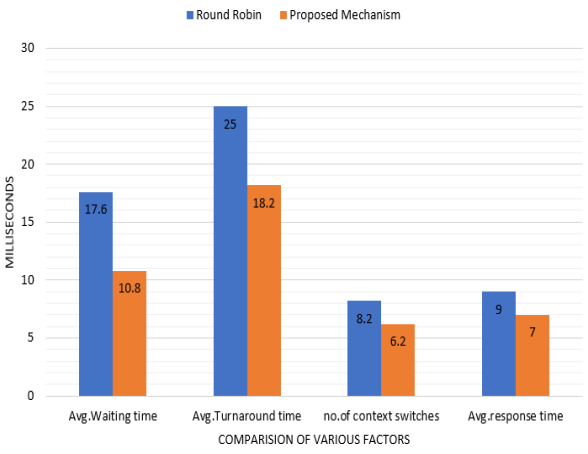

Fig 4: comparison graph of Round-Robin and proposed Hybrid Round Robin Mechanism

3.2 Experiment-2:

In this experiment six processes are defined with time and arrival time as Pr1, Pr2, Pr3, Pr4, Pr5 and Pr6.

Time quantum of RR algorithm=3ms

Table5. Process Id with their burst time

PROCESS-ID	ARRIVAL TIME(ms)	BURST TIME (ms)
Pr1	3	5
Pr2	4	4
Pr3	2	7
Pr4	0	8
Pr5	1	1
Pr6	5	2

3.2.1 RR-ALGORITHM:

Time quantum=3ms

3.2.1.1 Gantt Chart:

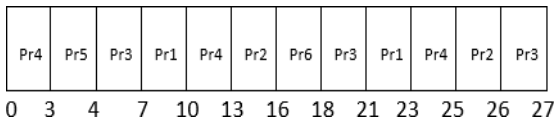

Fig-5: Gnatt chart

Table 6.RR-Algorithm Results

PROCESS ID	WAITING TIME (ms)	TURN AROUND TIME (ms)	RESPONSE TIME (ms)
Pr1	15	20	7
Pr2	18	22	13
Pr3	18	25	4
Pr4	20	25	0
Pr5	3	3	3
Pr6	16	13	16

Therefore, NOC=11

AWT= 15ms

ATT=18ms

ART=4.66ms

3.2.2 Proposed Algorithm:

Mean=4.5ms

Minimum time=1ms

ETQ=3ms

3.2.2.1 Gnatt chart:

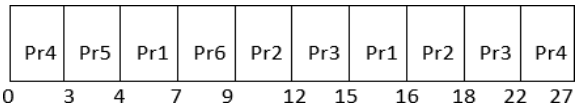

Fig-6: GNATT chart

Table 7. Proposed Mechanism Results

PROCESS ID	WAITING TIME (ms)	TURN AROUND TIME (ms)	RESPONSE TIME (ms)
Pr1	8	12	4
Pr2	10	15	9
Pr3	13	20	12
Pr4	19	27	0
Pr5	2	3	3
P6	2	4	7

Therefore,

NOC=9

AWT= 9ms

ATT=13.5ms ART=3.33ms

Table 8. Experiment-1 Result analysis based on processes arrival time

ALGORITHM	RR Algorithm	HYRR Mechanism (Proposed Algorithm)
AWT	15ms	9ms
ATT	18ms	13.5ms
ART	4.66ms	3.33ms
NOC	11md	9 ms

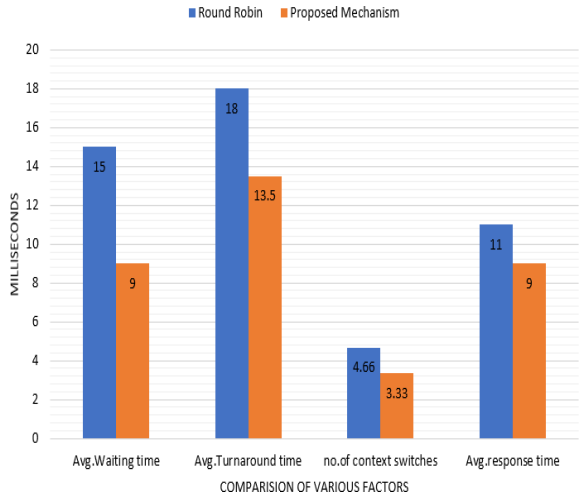

Fig 5: comparison graph of Round-Robin and proposed Hybrid Round Robin Mechanism

From above Table 7 and Fig 5, the ATT, AWT, ART and the NOC of proposed hybrid round-robin scheduling Mechanism has drastically decreased when compared with RR algorithm

5. CONCLUSION

In this paper the performance of traditional RR algorithm and proposed HYRR algorithm has been investigated and compared the results in terms of AWT, ATT, ART and NOC. The proposed mechanism is more effective in terms of maximum CPU utilization, throughput and minimized AWT, ATT, ART and NOC. The main reason is dynamic time quantum instead of static time quantum. The proposed mechanism can improve the performance of operating system in terms of better process management and optimization.

6. REFERENCES

[1] Singh H., Sarin S.K., Patel A., Sen S. (2018) Performance Analysis of Hybrid CPU Scheduling Algorithm in Multi-tasking Environment. In: Bhattacharyya P., Sastry H., Marriboyina V., Sharma R. (eds) Smart and Innovative Trends in Next Generation Computing Technologies. NGCT 2017. Communications in Computer and Information Science, vol 828. Springer, Singapore

[2] T. Singh, D. K. Srivastava and A. Aggarwal, "A novel approach for CPU utilization on a multicore paradigm using parallel quicksort," *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, Ghaziabad, 2017, pp. 1-6. K.

[3] U. Chindalia, S. C.S., S. Chandrashekar and O. Sharma, "Real Time Application and CPU Utilisation Monitoring Tool," *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2019, pp. 1136-1140.

[4] 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," *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, Jalgaon, 2016, pp. 184-187K.

[5] Yasin, A., Faraz, A., & Rehman, S. (2016). Prioritized Fair Round Robin Algorithm with Variable Time Quantum. In *Proceedings - 2015 13th International Conference on Frontiers of Information Technology, FIT 2015* (pp. 314–319). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/FIT.2015.62.

[6] M. A. Alworafi, A. Dhari, A. A. Al-Hashmi, A. B. Darem and Suresha, "An improved SJF scheduling algorithm in cloud computing environment," *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, Mysuru, 2016, pp. 208-212

[7] P. Pathak, P. Kumar, K. Dubey, P. Rajpoot and S. Kumar, "Mean Threshold Shortest Job Round Robin CPU Scheduling Algorithm," *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam, Tamilnadu, India, 2019, pp. 474-478

[8] L. Jian, Y. Xing, L. Ke-zhong and Y. Zhi-tao, "Study on the fluency of one-way waterway transportation based on First Come First Served (FCFS) model," *2015 International Conference on Transportation Information and Safety (ICTIS)*, Wuhan, 2015, pp. 669-674.

[9] J. Teraiya and A. Shah, "Comparative Study of LST and SJF Scheduling Algorithm in Soft Real-Time System with its Implementation and Analysis," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, 2018, pp. 706-711.

[10] R. Srujana, Y. M. Roopa and M. D. S. K. Mohan, "Sorted Round Robin Algorithm," *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019, pp. 968-971.

[11] A. Karapici, E. Feka, I. Tafa and A. Allkoçi, "The Simulation of Round Robin and Priority Scheduling Algorithm," *2015 12th International Conference on Information Technology - New Generations*, Las Vegas, NV, 2015, pp. 758-758.doi: 10.1109/ITNG.2015.131

[12] M. Aijaz, R. Tariq, M. Ghori, S. W. Rizvi and E. F. Qazi, "Efficient Round Robin Algorithm (ERRA) using the Average Burst Time," *2019 International Conference on Information Science and Communication Technology (ICISCT)*, Karachi, Pakistan, 2019, pp. 1-5.

[13] Zhou, P., Zheng, Y., & Li, M. (2014). How long to wait? Predicting bus arrival time with mobile phone based participatory sensing. *IEEE Transactions on Mobile Computing*, *13*(6), 1228–1241. https://doi.org/10.1109/TMC.2013.136

[14] Chandel, G. S., & Singh, P. K. (2014). International Journal of Advanced Research in Computer Science and Software Engineering. *International Journal*, *4*(2), 62–70. Retrieved from http://www.ijarcsse.com/docs/papers/July2012/Volume_2_issue_7/V2I700161.pdf

[15] J. C. Palencia, M. G. Harbour, J. J. Gutiérrez and J. M. Rivas, "Response-Time Analysis in Hierarchically-Scheduled Time-Partitioned Distributed Systems," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 2017-2030, 1 July 2017.

[16] I. Ungurean and N. C. Gaitan, "Performance analysis of tasks synchronization for real time operating systems," *2018 International Conference on Development and Application Systems (DAS)*, Suceava, 2018, pp. 63-66doi: 10.1109/DAAS.2018.8396072

[17] M. Sacha, "Measuring the real-time operating system performance," *Proceedings Seventh Euromicro Workshop on Real-Time Systems*, Odense, Denmark, 1995, pp. 34-40