Total Bandwidth Server*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Muhammad Iqbal Bin Mohd Fauzi
Hochschule Hamm- Lippstadt
*S*oest, Germany
Muhammad-iqbal.bin-mohd-fauzi@stud.hshl.de

***Abstract*—The purpose of the paper is to present the scheduling in a real-time embedded system. Multiprocessor system executed many types of tasks. There are two main tasks in this system: periodic and aperiodic tasks. These two tasks are commonly related to the algorithm in scheduling their execution time. Total bandwidth server algorithm provides one of the best methods for scheduling aperiodic tasks, which consists of a certain equation to find the shortest time to execute the task. Total bandwidth server contains a server that is hugely effective in improving responsiveness by allocating the earliest deadline for aperiodic task. Procedure on finding the earliest deadline for the aperiodic task while maintaining the schedulability is also being introduced. This paper will also introduce the adaptive total bandwidth server algorithm, which can be described as a total bandwidth server improvement. The worst-case execution time and predictive execution time will be compared in implementing the total bandwidth server and adaptive total bandwidth server.**

***Index Terms*—Total bandwidth server, real-time system, Schedulability technique, adaptive total bandwidth server, worst-case execution time, predictive execution time**

I. Introduction

In this modern technology century, different types of tasks exist in embedded system. There are two main tasks that are behaving in embedded system which are so called as periodic and aperiodic task. It is very common for these two tasks is mixed within a system. For example, in autonomous car system which can be interrupted by the human activities such as pedestrian walking on the street and want to crossroad [4]. To avoid such an accident, the scheduling algorithm for scheduling periodic and aperiodic tasks has been introduced and is widely used in the modern era. Furthermore, several scheduling algorithms with both periodic and aperiodic tasks have been presented. These algorithms can be categorized as either fixed-priority servers or dynamic priority servers. [1]. The fundamental concept of a fixed-priority server is rate monotonic scheduling. (RM) which the tasks are scheduled by their priorities. Priorities of the task is based on the period of the task. For example, task with shorter periods will have higher priorities compares with the task that have longer periods [3]. Deferrable server, Slack stealing, Sporadic server, Priority exchange are the examples fixed-priority servers [1]. In comparison, the scheduling of dynamic-priority servers is based on the earliest deadline first (EDF) principle. This server also will be scheduling the tasks based on priorities, but the priorities is based on the deadline of the task. The earliest deadline will have higher priorities for dynamic priority server [3]. Total bandwidth server (TBS), which are discussed in this paper is one of the examples of dynamic priority server. Generally, aperiodic tasks occur during an emergency and must be completed immediately. However, it should be noted that some systems do not require aperiodic tasks to be prioritized higher than periodic tasks [4]. The most widely used technique is to place the aperiodic task in an empty slot between the periodic tasks which is the main concept in total bandwidth server.

Identify applicable funding agency here. If none, delete this.

II. Dynamic Priority Server

One example of a dynamic priority server is total bandwidth server. The dynamic priority server's basic concept is that it will use the earliest deadline first scheduling method (EDF). The total bandwidth server algorithm is being implemented in such a way that it will aid in the reduction of time for aperiodic task requests. There are many more dynamic priority server examples, such as dynamic priority exchange, dynamic sporadic server, earliest deadline late server, and constant bandwidth server [3]. The dynamic scheduling algorithm is very efficient in assisting the processor in utilizing the aperiodic server, in this manner the aperiodic server's size is increased.

III. Worst-case execution time

WCET is used to assign the aperiodic deadline in total bandwidth servers. Not only for total bandwidth server but also in general, task execution in real-time scheduling will use the worst-case execution time. WCET works by estimating the execution time for a given task. For example, the execution time for an aperiodic task is unknown, but it can be estimated using WCET. In practice, the deadline in a total bandwidth server is unspecified, where WCET comes in. However, sometimes the actual execution time is longer than the worst execution time.

IV. Real-Time Scheduling

The multiprocessor handles the execution of aperiodic tasks in real time scheduling. One of the algorithms that will aid in the scheduling of aperiodic tasks on multiprocessors is the total bandwidth server algorithm. Real time scheduling algorithms are classified into two types: global scheduling algorithms and partitioning scheduling algorithms [4]. The distinction between these two algorithms is shown below on Table I.

TABLE I
COMPARISON BETWEEN GLOBAL SCHEDULING ALGORITHM AND PARTITIONING SCHEDULING ALGORITHM

No	Real-Time Scheduling	Explanation
1.	Global Scheduling Algorithm	There is only a queue with various tasks. All tasks in the queue will be handled by a predetermined number of multiprocessors.
2.	Partitioning Scheduling Algorithm	For this algorithm, there are numerous queues of various tasks. All of the task's queues will be executed in the specified multiprocessors.

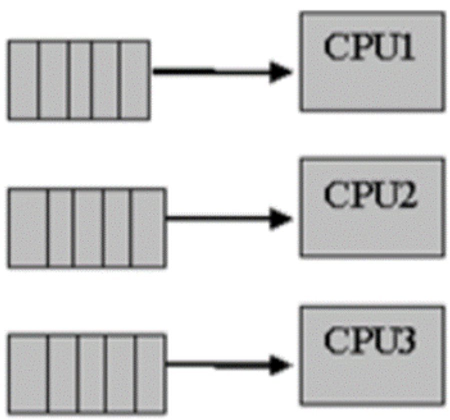

Fig. 1. Partitioning scheduling algorithm [4]

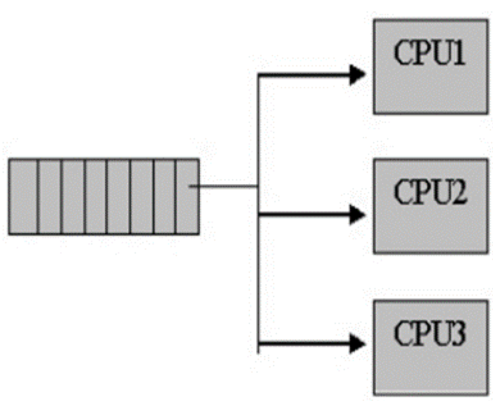

Fig. 2. Global scheduling algorithm [4]

V. EARLIEST DEADLINE FIRST

Earliest deadline first is a scheduler in real-time system. Total bandwidth server oversees assigning deadlines for aperiodic tasks in a server, but EDF oversees scheduling aperiodic tasks based on their priority. The deadline is the most important factor in determining priority, the shorter the deadline, the higher the priority. For example, the first aperiodic task has a deadline at 10 ticks and the second aperiodic task has a deadline of at 12 ticks. Due to its earlier deadline, the first aperiodic task will be given higher priority than the second aperiodic task. As a result, the first aperiodic task will be executed first.

VI. TOTAL BANDWIDTH SERVER (TBS)

The concept of total bandwidth server is to assign an aperiodic task in a server the earliest possible deadline. TBS will find the deadline for the tasks whenever an aperiodic task is assigned. In short, TBS intends to schedule aperiodic tasks using the EDF scheduler [3]. The formula to find the deadline is shown in figure 3

$$d_k = \max(r_k, d_{k-1}) + \frac{C_k}{U_s},$$

Fig. 3. Deadline for aperiodic task [3]

Where C_k represent the execution time of the aperiodic task, r_k represent the arriving time of aperiodic task and U_s is server utilization factor [3]. When an aperiodic task attains its deadline, it is added to the system's ready queue. Following that, EDF will schedule the task. The processor utilization will be used by aperiodic tasks. As a result, the deadline is critical to ensuring that the processor utilization used by the aperiodic task does not exceed the U_s (server utilization). The CPU utilization of the aperiodic task can also be calculated using the formula shown in figure 4. U_p represents the CPU utilization of the periodic task. For example, if there are two periodic tasks, U_1 and U_2, with CPU utilization values of 0.2 and 0.1, respectively, the total U_p will be 0.3. The CPU utilization for an aperiodic task is U_s = 1-0.3 = 0.7. As a result, the CPU utilization of an aperiodic task cannot be greater than 0.7.

$$U_s = 1 - U_p$$

Fig. 4. CPU utilization of aperiodic task [1]

A. *Example of Total Bandwidth Server*

Figure 5 depicts a total bandwidth server example. In this example, EDF has scheduled two periodic tasks. The first periodic task has a period of T_1=6 and an execution time of C_1=3, with a CPU utilization of 3/6 = 0.5. In contrast, the second periodic task has a period of T_2 = 8 and an execution time of C_2 = 2, with a CPU utilization of 2/8 = 0.25. Following

TABLE II
DESCRIPTION FOR FIRST , SECOND AND THIRD APERIODIC TASK

First Periodic Task	Second Periodic Task
C_1 = 3	C_2=2
T_1= 6	T_2 = 8

TABLE III
DESCRIPTION FOR FIRST AND SECOND PERIODIC TASK

First Aperiodic Task	Second Aperiodic Task	Third Aperiodic Task
C_1=1	C_2=2	C_3=1
d_1=7	d_2=17	d_3=21
r_1=3	r_2=9	r_3=14

that, the total CPU utilization for the periodic task, U_p, is 0.75. As a result, TBS utilization for aperiodic tasks is given by U_s = 1 – 0.75 = 0.25. Table II is the description for the two periodic tasks [3].

In this example, there are three aperiodic tasks that must be completed: the first, second, and third aperiodic tasks. These aperiodic tasks arrive at t=3, t= 9, and t=14 respectively with execution time C_1=1, C_2= 2, and C_3=1 respectively. We can now use the formula d_k = r_k + C_k /U_s to calculate the deadline for each aperiodic task. Initially, the first aperiodic task deadline can be calculated as $d_1 = r_1 + C_1/U_s$ = 3+1/0.25 = 7. Because the first aperiodic task has the earlier deadline than second periodic task, it will be executed first. The same thing happened when it came to determining the deadline for the second aperiodic task. Using the above formula, the deadline for the second aperiodic task will be d_2 = 17. However, this task will not be executed immediately because, at tick 9, a second periodic task with a shorter deadline, equal to 16, is running. At last, the third aperiodic task arrives at t=14, with a deadline calculated using the formula d_3= max (r_3, d_2) +C_3/U_s = 17 + 1/0.25 = 21. The same thing happened with the third aperiodic request, which could not be executed directly because a periodic task ran at t=14. Table III is the description for the three aperiodic tasks [3].

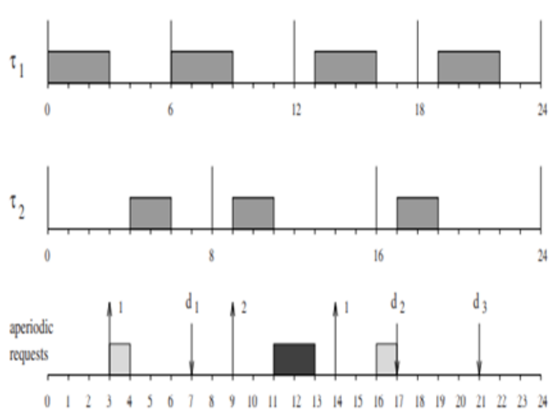

Fig. 5. CPU utilization of aperiodic task [1]

TABLE IV
DESCRIPTION OF FIRST AND SECOND PERIODIC TASK

First Periodic Task	Second Periodic Task
C_1 = 1	C_2=3
T_1= 4	T_1= 6

TABLE V
DESCRIPTION OF APERIODIC TASK OF ORIGINAL AND ADAPTIVE TOTAL BANDWIDTH SERVER.

Aperiodic Task for Original TBS	Aperiodic Task for Adaptive TBS
C_{WCET} = 3	C_{PET}=2
r_1= 3	r_1 = 3
d_{WCET} = 15	d_{PET} =11

VII. ADAPTIVE TOTAL BANDWIDTH SERVER

Predictive execution time is used instead of the worst execution time (WCET) in adaptive total bandwidth servers. This is due to the fact that the actual execution time, which is shorter, sometimes has a larger gap than the worst execution time [2]. Following that, a shorter execution time can be estimated without disrupting schedulability. Furthermore, when calculating the deadline, the shorter execution time, which is shorter than WCET, will result in a shorter deadline. This method allows for a faster response time. Instead of using WCET, several methods can be used to implement predictive execution time (PET). For example, the execution time will be chosen at random, and the execution time history will be used to predict the execution time [1].

A. Example of Adaptive Total Bandwidth

In this example, a comparison will be made between the original total bandwidth server and adaptive total bandwidth server. Figure 6 depicts the schedulability of the original TBS, whereas Figure 7 depicts the schedulability of the adaptive total bandwidth server. In this example, there are two periodic tasks. The period for the first periodic task is T_1 = 4, and the execution time is C_1 = 1. The period for the second periodic task will be T_2=6 and the execution time will be C_2=3. The CPU utilization for these two periodic tasks will be U_p=0.25 + 0.5 = 0.75. Following that, U_s= 1-0,75=0.25 is used to calculate the value for CPU utilization for aperiodic tasks. Table IV shows the description of the two periodic tasks [1].

In the case of aperiodic tasks, for the original TBS, the aperiodic task will arrive at r_1= 3, and the deadline for this task is given by the calculation 3+3/0.25 =15 based on WCET. The adaptive total bandwidth server will also arrive at r_3=3, but its execution time is less than that of the original TBS, which is =2, demonstrating that the actual execution time is less than the worst execution time. Following that, the ATBS deadline is given by 3+2/0.25 = 11. Table V show the description of the aperiodic task of total bandwidth server and adaptive total bandwidth server[1].

For the original TBS, the aperiodic task will be executed at tick 5, but the execution is not yet complete. Because of its earlier deadline, the second periodic task, will have higher priorities to be executed first. The execution of the aperiodic task will continue at tick = 10 and end at tick = 11. This results in a response time of 11-3=8. In the case of adaptive TBS, the execution of the aperiodic task will also begin at tick 5, but this time the execution of the aperiodic task will be completed. When compared to the original TBS, adaptive TBS will have an earlier deadline than the second periodic task. The aperiodic task will complete at tick 7 and give the response time 7-3=4. Below shows the example of original and adaptive TBS [1].

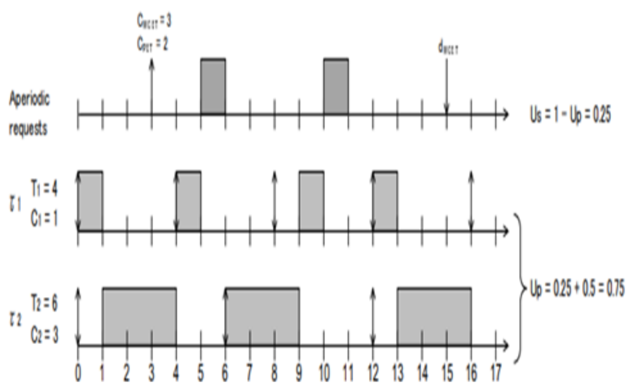

Fig. 6. Original total bandwidth server [1]

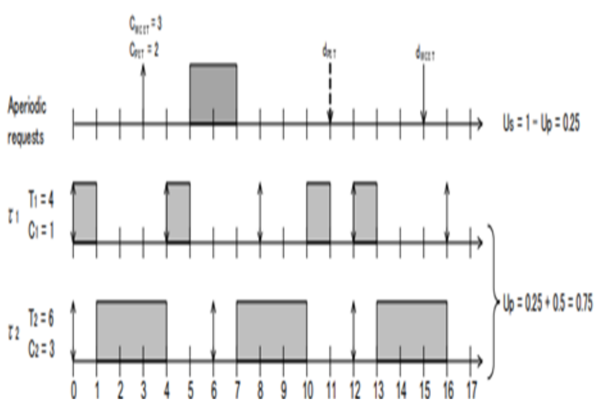

Fig. 7. Adaptive total bandwidtth server [1]

VIII. Total Bandwidth Server with UPPAAL

Uppaal is used in this example to demonstrate total bandwidth server modeling. Furthermore, this example will represent three task modeling, which are periodic task 1, periodic task 2, and aperiodic task, as well as a scheduler that functions to schedule the task.In this example, the deadline of the periodic task 1 is earlier than the deadline of periodic task 2. As a result, periodic task 1 will be executed first and periodic task 2 need to wait until periodic task 1 has already finish the execution. Because there is no aperiodic task arriving, the periodic task 2 will be executed after the periodic task 1 is completed. Below shows the modelling for execution of periodic task 1, periodic task 2 wait ,and execution of periodic task 2.

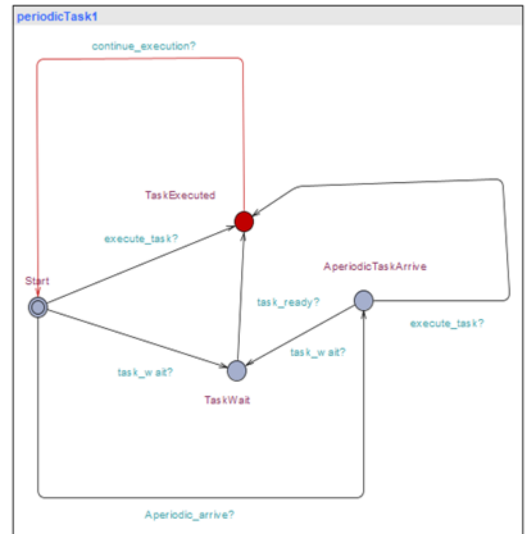

Fig. 8. Execution of periodic task 1

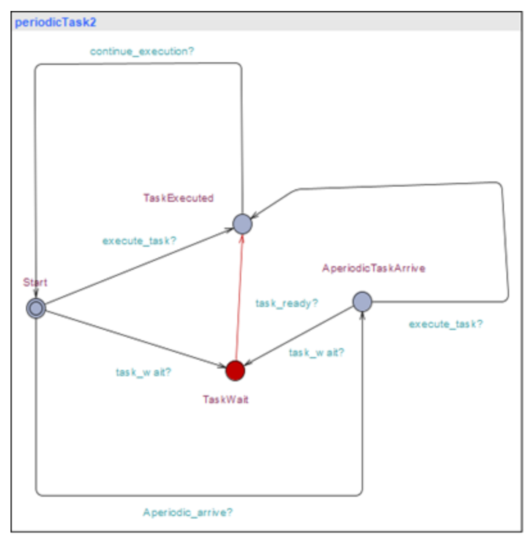

Fig. 9. Periodic task 2 wait until the execution of periodic task 1 is finished

In this modeling example, aperiodic task will have a longer deadline than both periodic tasks 1 and 2. Following that, in this model, the aperiodic task must be delayed. After the two periodic tasks have been completed, the aperiodic task will be executed, and then it will be the turn of the aperiodic task to be executed. The task's schedulability will be determined with the assistance of a scheduler. The execution of tasks will be determined by the scheduler based on their deadline. According to EDF, tasks with earlier deadlines will be prioritized and completed first. Figure 13 shows the scheduler modeling.

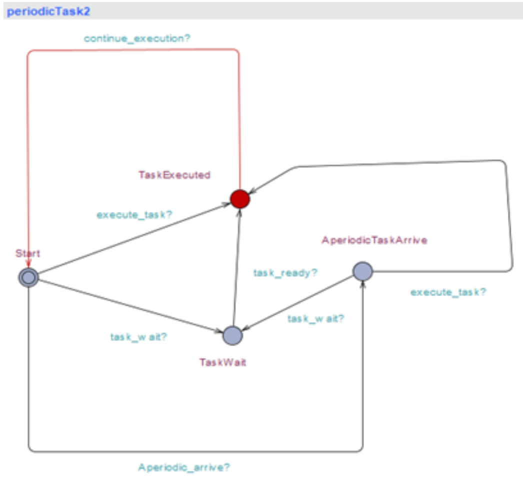

Fig. 10. Execution of periodic task 2

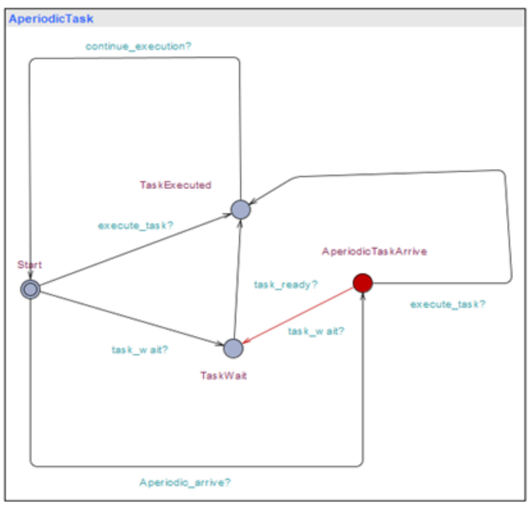

Fig. 11. Aperiodic task arrive

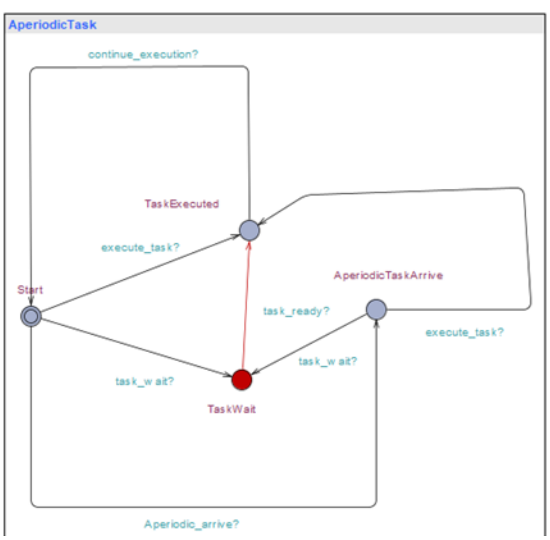

Fig. 12. Aperiodic tasks must wait due to its longer deadline

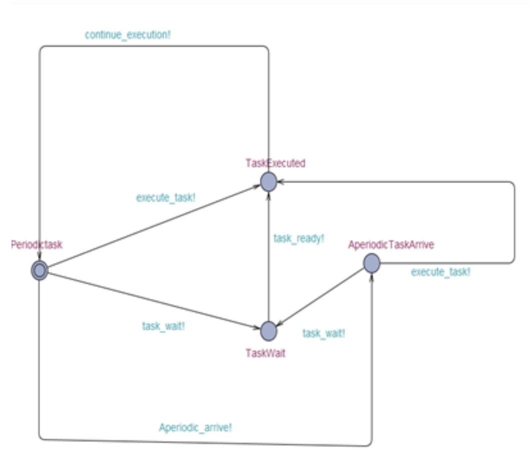

Fig. 13. Scheduler for periodic and aperiodic task.

IX. Conclusion

To summarize, the total bandwidth server is one of the most effective scheduling techniques for aperiodic tasks. The scheduling is accomplished by inserting the aperiodic task between the periodic tasks and calculating its deadline using the worst-case execution time. EDF is essential in scheduling tasks based on their priorities, which are periodic tasks and aperiodic tasks. Priorities are always determined by their deadline, with the earlier the deadline, the higher the priority. Instead of using WCET in TBS, the predictive execution time can be used as adaptive TBS, where the actual execution time is less than the worst execution time, resulting in a shorter response time.

References

[1] K. Tanaka, "Adaptive Total Bandwidth Server: Using Predictive Execution Time", IFIP Advances in Information and Communication Technology, pp. 250-261, 2013. Available: 10.1007/978-3-642-38853-8_23 [Accessed 24 May 2022]..

[2] D. Duy and K. Tanaka, "An effective approach for improving responsiveness of Total Bandwidth Server", 2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), 2017. Available: 10.1109/ictemsys.2017.7958777 [Accessed 24 May 2022]..

[3] G. Buttazzo, Hard real-time computing systems. Johanneshov: MTM, 2013.

[4] A. Khan et al., "A Migration Aware Scheduling Technique for Real-Time Aperiodic Tasks Over Multiprocessor Systems", IEEE Access, vol. 7, pp. 27859-27873, 2019. Available: 10.1109/access.2019.2901411 [Accessed 26 May 2022].

[5] S. Kato and N. Yamasaki, "Scheduling Aperiodic Tasks Using Total Bandwidth Server on Multiprocessors", 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008. Available: 10.1109/euc.2008.28